Weighted Hybrid Features To Resolve Mixed Entities

Ingyu Lee
Troy University
Troy, AL. USA

Byung-Won On
Advanced Institute of Convergence Technology
Suwon, Korea

***Abstract*—With the popularity of Internet, tremendous amount of unstructured document information is available to access. Extracting related information from huge unstructured documents is a very difficult task. Especially, confusion can occur by synonym and polysemy, miss spelling, abbreviation, etc. To resolve those confusion is known as an *Entity Resolution* problem. Clustering algorithms have been popularly used to resolve mixed entities. However, most researches focus on one feature of an entity such as co-author lists or paper titles. In this paper, we are proposing a weighted hybrid feature scheme to distinguish mixed entities among unstructured documents. Experimental results show that weighted hybrid approach improves the accuracy and efficiency.**

Keywords: mixed entity resolution, data mining, web document clustering, feature selections

I. Introduction

With the advent of Internet, tremendous amount of unstructured information such as web pages becomes available to public access. However, using partial identifier makes it difficult to distinguish two different entities which is called an entity resolution problem. In addition, spelling errors, synonym and polysemy, abbreviation makes the problem much more difficult. For example, there are *18* different *Wei Wang*s in DBLP author database as shown in Figure 1. Each of *Wei Wang* has the same name spelling but each name is a unique personnel. Our goal is to distinguish different entities using attributes such as co-authors and paper titles. Formally, a mixed 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.

Clustering algorithms have been popularly used to resolve mixed entities including K-means and hierarchical clustering. K-means is the most popular supervised clustering algorithm [15]. The algorithm repeatedly computes the distance from centroid and assigns the node to the nearest cluster [15]. Assume we have N entities, $e_1, \ldots, e_N$ and k clusters, $C_1, \ldots, C_k$ in the corpus. Then, K-means algorithm repeatedly computes distance from centroid m_j and assigns an entity e_i to the nearest cluster C_j. Then, K-means algorithm recomputes centroid $m_1, \ldots, m_k$ with new cluster members until the algorithm converges (no membership changes occur) or it reaches a maximum iteration. K-means is the most popular algorithm by its simplicity. On the other hand, hierarchical clustering generates a series of nested clusters by merging simple clusters into larger ones. Assume we have $p_1, p_2, \ldots, p_N$ partitions at the first level, then we compute the pairwise distance for each partitions. After that, we merge two closest partition p_i and p_j into one partition p_{ij}. The algorithm repeatedly merges the closest pair until it reaches to one partition or to the predefined cut off distance. Hierarchical clustering is an unsupervised algorithm which does not require the number of clusters in advance. However, since hierarchical clustering methods are not able to reallocate entities, it it plausible to be poorly classified in the early state of text analysis [26].

At the same time, Term Frequency (TF) and Inverse Document Frequency (IDF) have been popularly used as a feature to represent each entity in document vector space model. For example, DBLP name data set has co-authors and paper titles for each entity. Then, TF/IDF for each co-authors and paper titles can be used as features for each entity. Assume we have n textual documents and we want to represent each document d with m terminologies, then the corpse A can be represented as a vector as

$$A_{m \times n} = [d_1 | d_2 | \cdots | d_n] \quad (1)$$

where each document d_i is a vector which consists of $\{tfidf_1, tfidf_2, tfidf_3, \ldots, tfidf_m\}$. The component $tfidf_i$ is the multiplication of tf_i with idf_i for document d_i. Co-author names and paper titles are used to generate $tfidf$ vector components. Co-author list is known as a better feature and shows better performance in terms of accuracy. However, a single author document cannot be resolved by using a co-author feature. A paper title feature could resolve in the single author problem. However, the latter will falsely distinguish an author who is working on multiple venues. In our paper, we propose a hybrid method to use both author names and paper title features with different weights. We also explored micro (N-gram) and macro (Top-K) level features to resolve mixed entities.

The remainder of this paper is organized as follows. Section II describes a framework for mixed entity resolution and details of the proposed method. In Section III, we describe experimental validation with DBLP name data set. Related works are described in Section IV. Concluding remarks and future plans are followed in Section V.

978-1-4577-1539-6/11/$26.00 ©2011 IEEE

Fig. 1. Search results of *Wei Wang* in DBLP.

II. Methodology

Clustering algorithms applied on TF/IDF (Term Frequency and Inverse Document Frequency) matrices have been popularly used to resolve mixed entities. Especially, constructing TF/IDF matrices using co-author lists, paper titles, and venues are a popular approach. Using co-author lists show a better performance than using paper titles but clustering algorithms are not stable with co-author lists. Especially, for a single authored paper, the algorithm could not find a proper cluster. The paper titles TF/IDF matrix shows stable performance but overall quality is poor compare with co-author lists TF/IDF. Especially, if authors are working on several different areas (e.g. database, architecture, and network), then it is not easy to distinguish authors. In this paper, to get benefits from multiple attributes, we used a hybrid approach with co-author lists and paper titles.

In addition, we used two different levels of attributes selection: Micro-level and Macro-level. Micro-level N-gram method is based on the assumption that part of spelling error or abbreviation can be overcome by using N-gram algorithm rather than using the full term frequency. For example, 'John Kim' and 'J. Kim' are treated as the same entity using N-gram. It generally shows the better accuracy than using regular TF/IDF with additional cost to compute 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. With a traditional TF/IDF and Micro-Level N-gram could not distinguish the semantic difference. However, Macro-level N-gram could use semantic information with the given example.

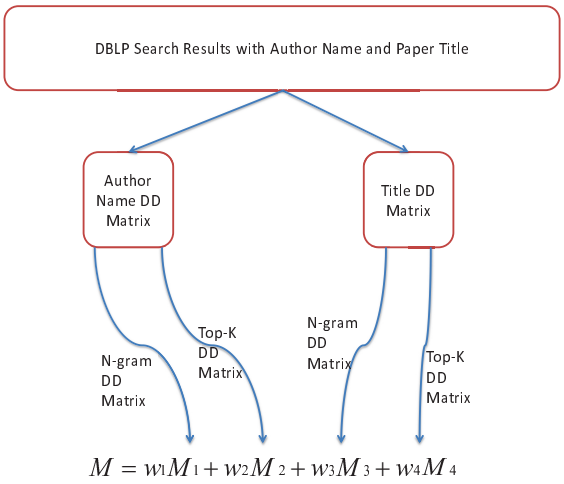

Fig. 2. Weighted Hybrid Framework.

To apply our weighted hybrid algorithm, we generated a term-document matrix A by

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

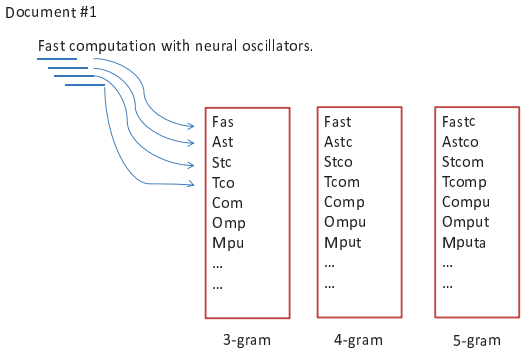

Fig. 3. N-gram Matrix Construction.

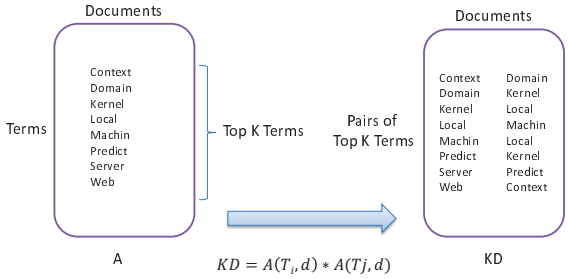

Fig. 4. Top-K Matrix Construction.

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 term t_i in document d_j, respectively. Then, we created document-document matrix by multiplying A^T (document-term matrix) with A (term-document matrix). Therefore, if two terms are appeared in documents d_i and d_j at the same time, the multiplication of two values contributes on $A(i, j)$ value elements. Otherwise (if two documents do not share a term), $A(i, j)$ element is set to zero as in

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

Intuitively, document d_i is strongly related with document d_j when two documents are sharing many terminologies.

Figure 2 shows the framework of weighted hybrid approach. First, 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 as shown in Figure 3. We also constructed Top-K co-occurrence matrices based on the assumption that two different terminologies are used in two different documents, then two documents are strongly related and the co-occurrence terminologies can be used a feature. Figure 4 shows the construction details for Top-K matrices. First, we sorted the terminology by decreasing order, then carefully selected terms which cover the whole documents by adding terminology one by one. Then, we computed the pair wise TF/IDF values by multiplying two TF/IDF values for Top-K terminology.

In our scheme, the similarity matrix L is defined as

$$L = \sum_{i=1,2} w_i \times (ND_i + TD_i + GD_i) \quad (4)$$

where w_i is a weight for each corpse document, ND_i is a regular TF/IDF matrix on title and authors, TD_i is Top-K macro-level TF/IDF with title and authors, and GD_i is N-gram TF/IDF with title and authors.

III. Experimental Validation

To measure the performance of different features, we used *DBLP* author name data set as shown in Table I. It has *18* different "Wei Wang"s who have exactly the same name spelling which makes the problem much more difficult. The name data set is extremely skewed as shown in Table I. For example, "Wei Wang" in UNC has *91* entries which is almost *1/3* of whole corpse while "Wei Wang" in Fudan has only *1* data entry. Furthermore, there are papers written by only one author which makes it much more difficult to properly cluster using only author name or paper title.

To evaluate the proposed method, we used precision, recall, and F-measure. The precision is defined as the number of entities correctly clustered divided by the number of entities in the cluster. The recall is defined as the number of correctly clustered entities divided by the number of entities in the solution set. The precision is high when it has large cluster and the recall is high when it has small size of cluster. To balance the latter, we also used F-measure which is an arithmetic mean of precision and recall.

Table II shows the performance results with the name data set using only **Title** field as a feature vector. Among those methods, using N-gram features shows slightly better precision than using other features. However, the recall for N-gram feature is worse than that of 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 but keep the recall as similar, we used different weights for each feature. As expected, the results shows better precision with similar recall. The recall is still very low compared to that of 5-gram. We conjecture that the corpse has *19* clusters, which is much higher than usual classification problem, worsen the recall.

Table III shows the performance results with **Authors** property feature. As expected, using co-author lists shows much better performances than those of using paper titles. Since the same group of authors prefers to work together, co-author lists can be used as a good entity resolution feature than using

ID	Description	Documents
Wei Wang 1	School of Life Science, Fudan University, China	1
Wei Wang 2	Department of Mechanical Engineering, MIT	2
Wei Wang 3	University of Maryland Baltimore County	5
Wei Wang 4	University of Naval Engineering	2
Wei Wang 5	Institute of Acoustics, Chinese Academy of Sciences	1
Wei Wang 6	Rutgers University, New Brunswick, NJ. USA	2
Wei Wang 7	Purdue University, Indianapolis	11
Wei Wang 8	INRIA Sophia Antipolis, Sophia Antipolis, France	16
Wei Wang 9	Institute of Computational Linguistics, Peking University	4
Wei Wang 10	National University of Singapore	3
Wei Wang 11	Nanyang Technological University, Singapore	3
Wei Wang 12	CSE, University of Nebraska Lincoln, NE. USA	20
Wei Wang 13	The University of New South Wales, Australia	36
Wei Wang 14	Language Weaver, Inc.	4
Wei Wang 15	The Chinese University of Hong Kong, Mechanical Engineering	3
Wei Wang 16	Center for ESC, Zhejiang University, China	2
Wei Wang 17	Fudan University, Shanghai, China	66
Wei Wang 18	University of North Carolina at Chapel Hill	91
Total		272

TABLE I
AUTHOR NAME DATA SET: *18* DIFFERENT WEI WANG.

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	0.61	0.25	0.35
Normal (0.2) + Micro (5-gram) (0.6) + Macro (top-30) (0.2) TF/IDF	0.64	0.26	0.37
Normal (0.1) + Micro (5-gram) (0.8) + Macro (top-30) (0.1) TF/IDF	0.65	0.24	0.35

TABLE II
EXPERIMENTAL RESULTS FOR AUTHOR NAME DATA SET USING **Title** PROPERTY.

a paper title. Among six different methods, N-gram shows the highest precision without losing 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 some weight values shows the similar performance for our data set. We conjecture that co-author lists itself shows high performance and could not be better with adding other features. However, we still have a difficulty to distinguish a single authored paper. The latter can be corrected by adding paper title field.

Table IV shows the experimental results of using weighted hybrid of **Titles** and **Authors** property. Since using co-author lists shows better performance than those of using paper titles, we used *0.8* for co-author lists and *0.2* for paper titles. The performance results is similar or slightly better than that of using co-author lists alone. In addition, 5-gram method shows the best precision and a decent recall. Since *DBLP* name data set is already cleaned and fixed errors such as typos and misspell, the benefits of using hybrid features is marginal at best. However, for a single authored paper, the hybrid method shows better performance to distinguish than other methods. We conjecture that corpse with a little bit errors, which is common in reality, our hybrid method will show better performance that others.

Figure 5 shows the similarity matrix structure for three difference cases: Titles, Authors and Hybrid (Titles+Authors). The structure shows the co-author lists matrix shows the best cluster structure than those of others. However, hybrid matrix preserves the details such as single author papers. It also shows that using paper title alone is difficult to distinguish different authors.

The performance of hybrid method depends on the selection and combination of weighting factors for each feature. The optimal weights can be converted to the solution of

$$L = ||A - XWX^T||_F^2 + \lambda||W - I||_F^2 \quad (5)$$

where A is a term document matrix and W is a diagonal matrix whose values are weight w_i for each different properties including the regular TF/IDF, N-gram and Top-K for co-author lists and paper titles. Matrix X is a corresponding data matrix of regular TF/IDF, N-gram and Top-K in column normalized format. The matrix W is a diagonal matrix with a weighting vector $w = (u, v, w, x, y, z)$ for three different methods of two

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	0.55
Macro (top-20) TF/IDF	0.89	0.38	0.54
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) TF/IDF + Macro (top-30) TF/IDF	0.92	0.39	0.55
Normal (0.2) + Micro (5-gram) (0.6) + Macro (top-30) (0.2) TF/IDF	0.91	0.42	0.57
Normal (0.1) + Micro (5-gram) (0.8) + Macro (top-30) (0.1) TF/IDF	0.91	0.39	0.55

TABLE III
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) TF/IDF + Macro (top-30) TF/IDF	0.87	0.40	0.55
Normal (0.2) + Micro (5-gram) (0.6) + Macro (top-30) (0.2) TF/IDF	0.89	0.39	0.55
Normal (0.1) + Micro (5-gram) (0.8) + Macro (top-30) (0.1) TF/IDF	0.90	0.39	0.54

TABLE IV
EXPERIMENTAL RESULTS FOR AUTHOR NAME DATA SET USING **Author** AND **Title** PROPERTIES.

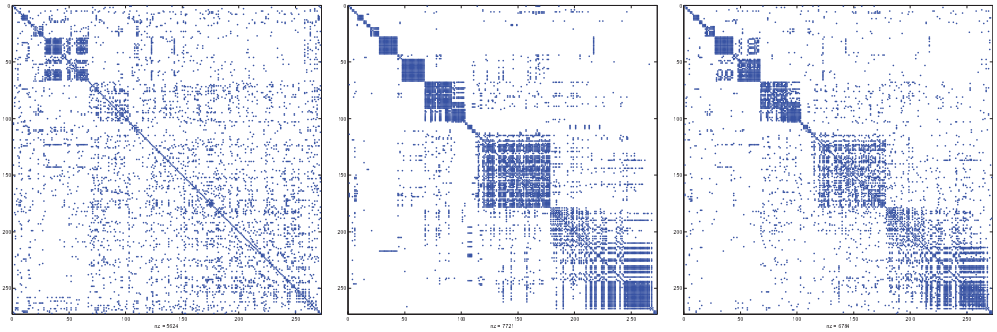

Fig. 5. Similarity (Document-Document) Matrix Structure for **Titles**(Left), **Authors**(Middle), and **Titles+Authors** (Right).

different matrices. To get the optimal weight value, we take partial derivatives for (u, v, w, x, y, z) and set to zero. The solution of the linear equations $\frac{\partial L}{\partial u} = \frac{\partial L}{\partial v} = \frac{\partial L}{\partial w} = \frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} = \frac{\partial L}{\partial z} = 0$ is the optimal weighting factors. The next step is using learning algorithm to change the weighting factors during the process such as support vector machines.

IV. RELATED WORKS

Many researches have been done to resolve mixed entities. Han et al. in [22] proposed two supervised learning-based approaches: one is based on Naive Bayes Model and the other is using Support Vector Machines. The authors also proposed a *K*-way spectral clustering method to resolve mixed entities. Since the spectral clustering considers the global connectivity, the proposed method shows better performance for overlapped venue or authors. Malin [25] utilized hierarchical clustering methods on the exact name similarity. 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.

Recently, Bekkerman et al. in [4] proposed methods for disambiguating namesakes that appear in the web using Agglomerative using link structure of web pages and Conglomerative Double Clustering which uses a multi-way distributional clustering method. Monkov et. al. used a lazy graph walk algorithm for disambiguating 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.

As authors aware, our paper is the first approach to build a weighted hybrid scheme of features to resolve mixed entities. Using one attribute feature to resolve mixed entity may be limited by typos, miss spellings, polysemy and synonym, single author paper, etc. However, combining several attributes with different weights can avoid the aforementioned problems.

V. Conclusion

In this paper, we proposed a weighted hybrid scheme to select features to resolve a mixed entity problem. In our experiment, using co-author list TF/IDF shows better performance than using paper title TF/IDF. To improve the reliability, we proposed a hybrid approach which combines the co-author list with paper title. We also 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 short length and the corpse got spelling error, and abbreviations. The macro level Top-K scheme can detect the semantical difference by using co-occurrent terminologies. The experimental results show that the proposed hybrid method keeps the accuracy with handling a single author document.

The current version of the algorithm is based on a supervised algorithm. In reality, unsupervised method is more natural approach than supervised one. In the near future, we will develop a semi-supervised algorithm based on the feedback from user experience. 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.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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