AEFS: Authoritative Expert Finding System Based on a Language Model and Social Network Analysis

Chia-Ching Chou*, Kai-Hsiang Yang†, Hahn-Ming Lee*†

***Department of Computer Science and Information Engineering**
National Taiwan University of Science and Technology
Taipei 106, Taiwan
{M9415048, hmlee}@mail.ntust.edu.tw

†Institute of Information Science,
Academia Sinica,
Taipei 115, Taiwan
{khyang,hmlee}@iis.sinica.edu.tw

Abstract

Searching for experts on a given topic is a critical problem in many real-world situations, such as collaborative finding. Even so, previous work has only focused on searching for experts based on the appearance of topic query in an organization's documents, which means that the experts selected might not be suitable for the task at hand. To resolve this problem, we propose an Authoritative Expert Finding System, called AEFS, which ranks the publications of experts to indicate their level of expertise. AEFS uses non-textual information, e.g. impact factor, to represent the quality of publications, and provides a citation matching function that removes duplicated citations based on the concept of centrality in social network analysis (SNA). In our experiments, we compare a number of related approaches to show that: (1) the proposed approach achieves a good performance in terms of the average F-measure; (2) citation matching can reduce the number of training examples required; and (3) non-textual features are very effective for searching for experts.

1 Introduction

Finding experts who have the proper skills and knowledge in a particular field has become increasingly important in recent years [35]. The task, called expert finding [2, 32, 3] is often critical to the success of projects. For example, an enterprise may want to find employees who have appropriate skills to solve a special problem, or a conference may need to find a reviewer who has the necessary expertise to review a technical paper. Traditional expert finding approaches store data about each expert in a database [17, 27], which is maintained manually. However, such approaches are very expensive because of the manpower required to manage a database.

The Text REtrieval Conference (TREC) has provided a platform with an enterprise track for the expert finding task since 2005 [35]. Given a query on a particular topic, the goal of the track is to rank a list of candidate experts based on a set of documents related to the query. However, existing approaches can only find relevant experts, but can not ensure that the experts are authoritative.

An expert finding system has two goals. One is to discover "who knows what", and the other is to identify "who are the experts on a given topic q". The first is called the expertise location or expertise finding problem [2], and the second is called the expertise identification problem. In this paper, we focus on the latter issue.

This paper is motivated by an organization that often needs to find experts to review technical proposals and scientific research papers. Experts are usually selected by a committee of the organization's senior researchers. However, this could be unfair because the committee members may have personal interests that influence their choice of experts. This is called the Conflict of Interest (COI) problem [1]. To accelerate the selection process and generate more accurate results to prevent the COI problem, we propose an automatic expert finding system that

- identifies appropriate experts efficiently;
- does not maintain a database of expertise;

- detects conflicts of interest between candidate experts and the authors of proposals and papers under review.

The proposed Authoritative Expert Finding System (AEFS) involves four phases: (1) publication crawling, (2) citation matching, (3) citation ranking, and (4) expert ranking. We collect the publications of experts from the Web as our dataset and remove duplicated records via a citation matching mechanism. AEFS then uses the remaining citations as experts' profiles to rank the experts. The system is based on the probabilistic language model, which has been applied successfully in Information Retrieval (IR) systems.

The remainder of this paper is organized as follows. In Section 2, we describe the architecture of AEFS. Section 3 contains a performance evaluation of the proposed approach. Then, in Section 4, we present our conclusions and discuss avenues for future research.

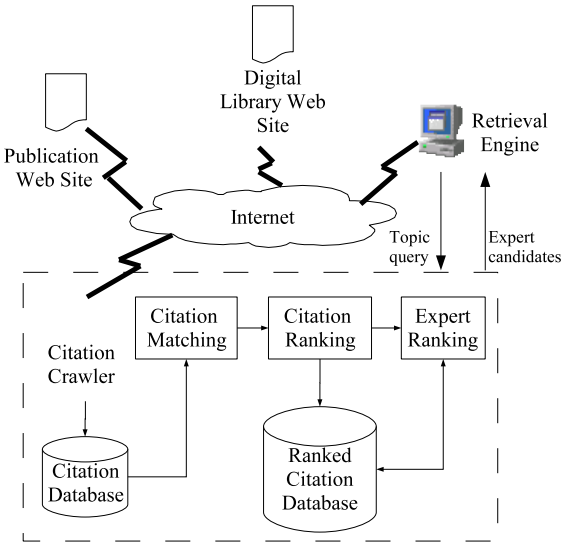

Fig. 1: The Architecture of AEFS.

2 System Architecture

This section describes the architecture of the Authoritative Expert Finding System (AEFS). Basically, AEFS uses the impact factor of a venue as evidence to show the quality of citations in order to search for more authoritative exports for a given topic. As shown in Figure 1, the system architecture consists of the following modules: Citation Crawler, Citation Matching, Citation Ranking, Expert Ranking, and an Expert Retrieval Engine. The Citation Crawler searches web sites for publication lists and extracts the bibliographic sections to collect citation information, which is then stored in the Citation Database. In addition, the Citation Crawler uses an author's name to query the web sites of digital libraries to gather more citation information. The Citation Matching mechanism removes duplicate citations, and Citation Ranking orders the citations for a given topic. The Expert Ranking mechanism then collates the ranked citation list and the experts' profiles to generate the final ranked list of expert candidates. The Retrieval Engine provides a user interface to support the search service that AEFS provides to users. We discuss the Citation Matching, Citation Ranking, and Expert Ranking functions in the following subsections.

2.1 Citation Matching

The goal of Citation Matching is to identify different citations that refer to the same paper because citations that appear in papers or web pages may follow different citation formats. Traditional approaches for matching citations use some standard similarity metrics to determine whether two citations refer to the same author or paper [7]. However, it is not easy to determine a threshold for the similarity metrics. To address this problem, Felligi and Sunter [20] proposed learning distance functions for entity pairs in order to determine the threshold. They used some training samples and computed the samples' similarity distances to generate pair-wise vectors for training a pairing function. Several approaches [28, 6, 31, 15] are based on Felligi and Sunter's work.

However, learning distance functions is still a challenging issue. The algorithm is inefficient because it needs to generate all pairs of citations from the training data to train the pairing function [15]. To address this problem, we propose a citation matching mechanism called CMSNC, which is based on Social Network Centrality. The social network centrality measure is an important structural property in social network analysis. The concept of centrality indicates the importance of the nodes in the network [38]. The CMSNC reduces the required number of training samples so that they are balanced.

Basically, CMSNC finds the most representative pairs of citations in a cluster, so it substantially reduces the number of training examples. The social network centrality technique is used to find the most popular nodes in a graph [38]. In this work, we use the most popular nodes in a graph to generate the representative pairs of citations. Because there are many similar pairs of citations, only a few representative samples are needed.

The CMSNC model is comprised of the following components: a Social Network Centrality Pair Generator, an Attribute Dependency Similarity Calculator, a Binary Classifier, and a Social Network Centrality Cluster Builder. Figure 2 shows the system architecture of the CMSNC Model. We discuss the various components in the following subsections.

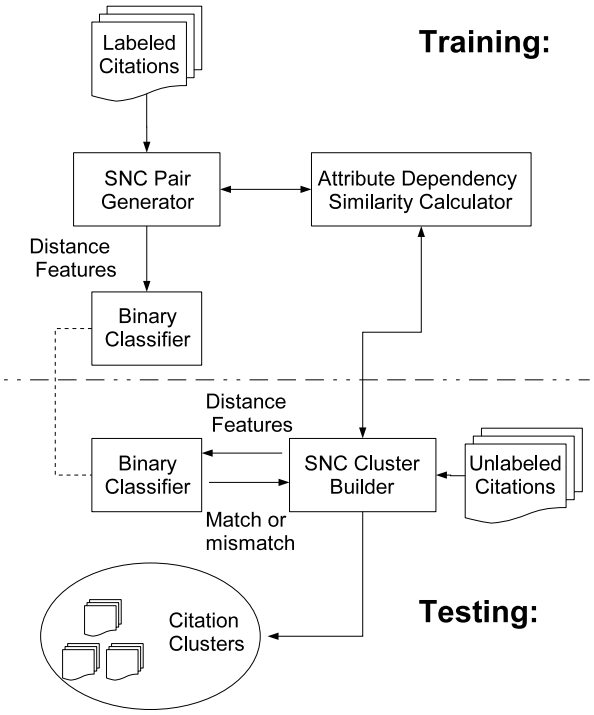

Fig. 2: The System Architecture of the CMSNC Model.

2.1.1 Social Network Centrality Pair Generator

The Social Network Centrality Pair Generator is used to produce representative pairs of citations in each cluster. An element of a training citation set $C = c_1, ..., c_n$ is regarded as a vertex in a graph. In the graph, an edge exists between c_i and c_j if and only if c_i and c_j in the same cluster and the weight of the edge, $(c_i, c_j) > T$, where T is a threshold that filter the edges with relative low weight. The weights of edges can be computed by a cheap metric [30] and the threshold can be set by the average weight of edges.

We determine the centrality degree, which is a social network centrality measure, to find the center node of each connected component/cluster in the weighted graph. The center node is the node that has maximum degree in each cluster. We use the centrality degree technique to accelerate the process speed because of its low computation cost than other centrality measures. Then, the center nodes are combined to generate pairs and the weight of edge of those are recomputed by Attribute Dependency Similarity Calculator. The labels of the pairs are positive if their elements are in the same cluster, and negative if their elements are in different clusters. The labeled pairs are then added into a binary classifier.

2.1.2 Attribute Dependency Similarity Calculator

In a pair-wise matching approach, we must provide an effective and efficient similarity metric for comparing the features of citations. In this work, we adopt soft-TFIDF as our basic similarity metric because it can work well not only for typographical errors but for two equivalent strings expressed by multiple words that are added or transported [13]. Soft-TFIDF is a hybrid similarity metric. It needs a character-based similarity metric as its secondary similarity metric that performs well on short strings. In implementation, we use Jaro-Winkler similarity metric as the secondary similarity metric.

In addition, the attributes of citations can be obtained by applying the domain knowledge, which is specified by experts or learned from training data. In this paper, we use four attributes, namely author, title, venue, and date. We calculate the default similarities of each attribute individually with soft-TFIDF. Many researchers assume that a single article cannot be published in two different venues [18]. This assumption means that some attributes would be affected by the other attributes. For instance, if the title and author of two citations are very similar, we can assume that the venue of the two citations is the same. We could increase the similarities in the following ways:

1. We multiply the similarity of venues by an increasing factor if the other similarities are above a predefined threshold.

2. We multiply the similarity of authors by an increasing factor if the other similarities are above a predefined threshold.

3. We multiply the similarity of dates by an increasing factor if the other similarities are above a predefined threshold.

2.1.3 Binary Classifier and Social Network Centrality Cluster Builder

Another problem we consider is citation clustering. In this case, we have to partition a set of citations into k sub-sets (where k is the number of clusters). The proposed CMSNC can easily achieve this task, which will be detailed below.

To cluster citations that refer to the same paper, all pairs of citations are classified as either positive or negative by a trained binary classifier. Pairs that are classified as positive are put in the same cluster, and an edge is added between the pairs. After all pairs have been classified, we have an un-weighted graph with several connected components (clusters). Because we only use a few training samples we have to choose a classifier that is suitable for a learning task with a small number of examples. The Support Vector Machine (SVM) model is a classifier that meets our requirements [36].

In our implementation, we adopt LIBSVM [12] as our binary classifier. We use the string similarity of two citations as our features. If the citations have been segmented, we only calculate the similarities of four fields which are author, title, venue and date.

Some naive algorithms surveyed in [15] need to generate all pairs in the clusters created. Therefore, we propose a Social Network Centrality Cluster Builder to address this problem. We do not generate all pairs at once; instead, we generate them sequentially. When a new node is added to the graph, it is tested to determine if it has an edge with the other added nodes according to the classified result. We then grow the graph into several clusters and apply the social network centrality measure to find the center node in each cluster. The nodes that are not added to the graph can only be compared with the center node. This is very similar to the work of the Social Network Centrality Generator, except that the graph un-weighted in this case.

2.2 Citation Ranking

Citation Ranking is an IR system that ranks citations with respect to a given topic query. We use a language model to rank each citation according to the given query because the model has been applied successfully in many IR systems [33]. In the language model, the probability of a query q is generated by a probabilistic model based on a document d. Given a query q with s words $q = q_1q_2...q_s$ and a document d with m words, $d = d_1d_2...d_m$, the probability, denoted by $p(q|d)$, can be calculated by the following Bayes' [4] formula:

$$p(d|q) \propto p(q|d)p(d) \tag{1}$$

The multinomial model that assigns the probability $p(q|d)$ is:

$$p(q|d) = \prod_{i=1}^{s} p(q_i|d) \tag{2}$$

where $p(q_i|d)$ is estimated by maximum likelihood:

$$p_{ml}(q_i|d) = \frac{tf_{q_i,d}}{|d|} \tag{3}$$

where $tf_{q_i,d}$ is the term frequency of q_i in d and $|d|$ is the total number of terms in the documents. However, the probability $p_{ml}(q_i|d)$ may be zero due to data sparseness. This is called zero frequency problem [39]. Hence, we adopt Jelinek-Mercer smoothing method [22] to smooth for our language model:

$$p(q_i|d) = (1 - \lambda)p_{ml}(q_i|d) + \lambda p(q_i|C) \tag{4}$$

where $\lambda \in [0, 1]$ and C is a corpus.

In general, the probability of document $p(d)$ is assumed to be uniform, so it does not affect the document ranking [5, 33]. However, in the expert finding task, in order to find appropriate experts, the quality of a paper should be considered by our system because the language model can only find relevant documents by the statistics of words. Hence, we apply impact factor of journal to set the probability of document, $p(d)$.

2.3 Expert Ranking

An expert might have several citations that can be taken as his/her profile. In the expert finding task, we need to rank expert candidates for a given topic q by ranking their citations. In this work, we employ the voting process in C. Macdonald et al. [26] because it can be easily and flexibly combined with any IR system. The mechanism aggregates the votes for each candidate to produce the final ranking list of expert candidates. An expert receives a higher score if his citations have higher ranking values.

Expert Ranking also provides a function that can identify experts who have close relationships with some given experts' names. AEFS includes a COI Detector that can filter out experts who have a conflict of interest with given experts' names. We discuss COI Detector next.

We integrated the COI detector function into our system. The goal of this function is to detect the conflict of interest problem among potential reviewers and authors. To implement this function, AEFS collects theses and publications from the Electronic Theses and Dissertation System [19] and parses the bibliography sections to construct a social network. The social network is a graph in which the nodes represent some objects e.g., people and organizations, which are connected by one or more relations, such as friendships, communications, and kinship. Then, we analyze the social network to get five relationships, student, teacher, classmate, student of student and teacher of teacher. Here, the COI problem arise if people have those relationships. Thus, the system automatically checks and removes all experts who have those relationships.

3 EXPERIMENTS

This section presents the experiment results. We conducted two experiments: one to analyze the performance of CMSNC and the other to evaluate the effectiveness of the impact factor feature in solving the expert finding problem. To evaluate the effectiveness of the CMSNC, we tested it on two datasets: a segmented dataset and a un-segmented dataset. The segmented data comprised citations that were segmented into several fields, such as author, title, venue, and date. The un-segmented data comprised citations in the dataset that formed continuous strings.

We also compared the CMSNC with the CRF [28] and Learned Edit Distance [6] on un-segmented dataset, and with the INDEPDEC and the DEPGRAPH [18] on

the segmented dataset. The datasets are described in following subsections.

3.1 Datasets

As mentioned above, we used two datasets. They were the Citeseer dataset and the Cora dataset. The Citeseer dataset [25] contains un-segmented citations and is divided into four subsets: the Reinforcement, Reasoning, Face, and Constraint subsets [16]. The Reasoning subset contains 514 citation records that represent 196 unique papers; the Face subset contains 349 citations for 242 papers; the Reinforcement subset contains 406 citations for 148 papers; and the Constraint subset contains 295 citations for 199 papers. These datasets have been used in other works [25, 6, 28]. As in Bilenko and Mooney [6], we use a 50/50 train/test split of the data, and repeat the process with the folds interchanged. The Cora dataset is segmented and contains 1,295 distinct citations to 122 Computer Science research papers from the Cora Computer Science research paper search engine. It was previously used in [6, 18].

In the second experiment, we constructed a dataset to evaluate the AEFS. We collected the personal data of 882 experts who have submitted scientific proposals to the Division of Computer Science of the National Science Council (NSC) of Taiwan. The personal data includes each expert's name, affiliation, expertise and publications. The experts are separated into nine groups that cover different topics by the NSC according to their expertise; an expert can belong to more than one group. The number of categories of expertise used to classify experts' personal data is 53. The classification is within Chinese. We translated it into English and used the categories of expertise as the query terms. Table 1 shows the distribution of the number of query terms in each topic.

Table 1: Distribution of query terms; N denotes the number of query terms in the corresponding topics

Topic	N
Image and Pattern Recognition	6
Natural Language and Speech Processing	4
Artificial Intelligence	6
Computer Graphics	8
Information System Management	7
Database	5
Bioinformatics	7
Web Technologies	7
Quantum Computing	3

3.2 Experimental Methodology

As in many other citation matching research works, we measure the overall performance in terms of the recall, precision, and F-measure, which are defined as follows:

$$F_{measure} = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5)$$

where precision and recall are defined by following equations:

$$Precision = \frac{CIDP}{IDP} \quad (6)$$

$$Recall = \frac{CIDP}{TDP} \quad (7)$$

where CIDP means correctly identified duplicated pairs, IDP means identified duplicated pairs and TDP means true duplicated pairs. For the evaluation of expert finding task, we used the following measures to evaluate the performance of AEFS: Mean Average Precision (MAP), R-precision, precision@10, and precision@20 [35].

3.3 Experiment 1

In Experiment 1, we evaluate CMSNC by comparing it with other citation matching approaches. Table 2 shows the results on the un-segmented data. It is clear that the CMSNC yields a higher average F-measure than other two methods. We also observe that CMSNC is insensitive to the training data. Table 3 shows the results on the segmented data. In this case, CMSNC is not better than the other methods. We find that the classifier can not correctly predict negative samples with high similarity because there are only a few pairs of citations in the training samples. Even so, CMSNC provides an efficient way to match the citations because it does not require tuning parameters.

Next, we show that the CMSNC can reduce the required number of training examples from C_2^n to $C_2^p + (n-p)$, where n is the total number of citations and p is the total number of citation clusters. Suppose that a citation set $C = c_1, c_2, ...c_n$ is partitioned into p clusters, each of which has a center node x, and all the center nodes form a center node set, $X = x_1, x_2, ..., x_p$. Then, the elements in X are paired with each other to generate C_2^p negative examples. Meanwhile, x_i is paired with c_j if x_i and c_j are in the same cluster and $x_i \neq c_j$ to generate positive examples. Finally, the total number of required training samples is $C_2^p + (n-p)$.

3.4 Experiment 2

The objective of Experiment 2 is to determine whether the non-textual features work well and whether the parameters affect the performance of the AEFS.

Table 2: Average F-measure for citation matching on four Citeseer datasets. The top two rows are the results reported by McCallum *et al*. [28] and Bilenko [6]; the bottom row shows the performance of the proposed CMSNC method

	Reinforcement	Constrain	Face	Reasoning	Average
CRF [28]	0.917	0.976	0.918	0.964	0.944
Learned Edit Distance [6]	0.907	0.966	0.938	0.948	0.940
CMSNC	0.916	0.958	0.956	0.957	0.947

Table 3: Average precision, recall and F-measure on Cora dataset, the top two rows are results from X. Dong *et al*. [18] ; the bottom row is our result

	Precision/Recall	F-measure
INDEPDEC	0.997/0.977	0.987
DEPGRAPH	0.999/0.976	0.987
CMSNC	0.979/0.977	0.978

We adopt the Institute for Scientific Information Impact Factor (ISI IF) as our non-textual feature, because we believe it can represent the quality of a paper, and thereby improve the search quality of the AEFS. From the results listed in Table 4, we observe that AEFS works well with non-textual features, and all measures have been improved. It is easy to see that applying the impact factor can improve the quality of citation ranking; that is, it enables us to find more authoritative experts.

For smoothing, the dataset was randomly split into 2 folds for cross-validation for each experimental run. Figure 3 shows the degree of variation for different values of the λ parameter. The optimal performance is achieved when $\lambda = 0.2$. When λ approaches 0, it means that there is no smoothing and the documents that match more query terms will be ranked higher than those that match fewer terms. This shows that AEFS is insensitive to different λ values.

For the performance of COI detection, we show the number of COI occurrences. We use all the combination of expert names and topics to test our system. A total of $2,711$ COI occurrences are detected from $22,800$ queries. This means that if we don't consider the COI problem, inappropriate experts will be recommended.

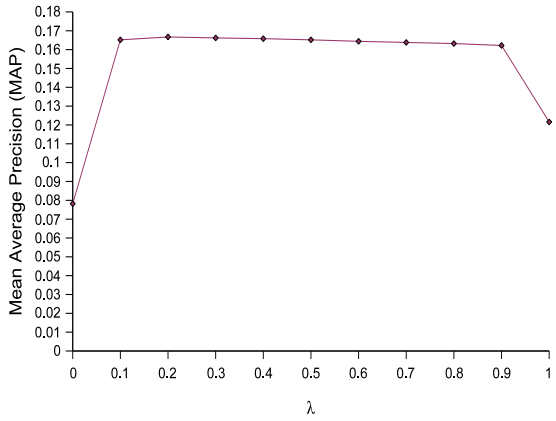

Fig. 3: Mean average precision with different λ values.

Table 4: Results of the proposed expert finding system with impact factor and without impact factor

	MAP	R-prec	P10	P20
with IF	0.167	0.266	0.520	0.488
without IF	0.156	0.257	0.501	0.454

4 Conclusion and Further Work

Our expert finding system contains many useful tools, including a Citation Matching module that speeds up the training process, and a COI Detector that detects experts who might have conflict of interest relationship with a given name. The most importance is that impact factor is effective in improving the search quality. We apply the ISI Impact Factor when ranking citations in order to improve the search quality. The experiment results show that we can achieve a significant improvement in the search performance. When the impact factor is applied, the Mean Average Precision (MAP) improves from 0.156 to 0167.

In addition, we have proposed a Citation Matching mechanism based on the CMSNC technique used in Social Network Analysis. The CMSNC is inspired by the social network centrality measurement. The objective is to find the center node in a social network, and thereby reduce the number of training samples. In order to avoid repeatedly computing citations refer to a paper many times, it is also important to identify the different formats of citations that refer to the same paper in an expert finding system. Our experiment results show that CMSNC yields a higher average F-measure than other

approaches on un-segmented data. Moreover, CMSNC does not need to generate pairs when clustering citations carefully tune the system's parameters. The model also reduces the required number of training examples from C_2^n to $C_2^p + (n - p)$.

The most important task for AEFS is to rank citations authoritatively, but the ISI Impact Factor is only a measure of popularity, not of prestige [8]. In our future work, we will apply link structure analysis, such as the PageRank [10] and HITS [23] algorithms, to improve the citation ranking results. In [8], the authors demonstrate how a weighted version of the popular PageRank algorithm can be used to obtain a metric that reflects prestige.

Acknowledgement

This work was supported in part by the National Digital Archive Program(NDAP, Taiwan), the National Science Council of Taiwan under grants NSC 95-2422-H-001-024, and also by the Taiwan Information Security Center (TWISC), the National Science Council of Taiwan under grants NSC 95-2218-E-001-001, and NSC 95-2218-E-011-015, moreover by the International Collaboration for Advancing Security Technology Program(iCAST), the National Science Council of Taiwan under grant NSC 95-3114-P-001-002-Y02 and NSC95-3114-P-001-001-Y02.

References

[1] B. Aleman-Meza, M. Nagarajan, C. Ramakrishnan, L. Ding, P. Kolari, A. P. Sheth, I. B. Arpinar, A. Joshi, and T. Finin, "Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection," in *Proceedings of the 15th International Conference on World Wide Web*, 2006, pp. 407–416.

[2] K. Balog, L. Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise corpora," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 43–50.

[3] K. Balog and M. de Rijke, "Finding experts and their details in e-mail corpora," in *Proceedings of the 15th International Conference on World Wide Web*, 2006, pp. 1035–1036.

[4] R. T. Bayes, "An essay towards solving a problem in the doctrine of chances," *Philosophical Transactions of the Royal Society London*, vol. 53, pp. 370–418, 1763.

[5] A. Berger and J. Lafferty, "Information retrieval as statistical translation," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 222–229.

[6] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 39–48.

[7] M. Bilenko, R. J. Mooney, W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "Adaptive name matching in information integration," *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 16–23, 2003.

[8] J. Bollen, M. A. Rodriguez, and H. Van de Sompel, "Journal status," *Scientometrics*, vol. 69, no. 3, pp. 669–687, 2006.

[9] B. E. Boser, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992.

[10] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of 7th International World-Wide Web Conference*, 1998, pp. 107–117.

[11] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom, "Expertise identification using email communications," in *Proceedings of the 12th ACM Conference on Information and Knowledge Management*, 2003, pp. 528–531.

[12] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[13] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03)*, 2003.

[14] W. W. Cohen and J. Richman, "Learning to match and cluster entity names," in *ACM SIGIR'01 workshop on Mathematical/Formal Methods in IR*, 2001.

[15] W. W. Cohen and J. Richman, "Learning to match and cluster large high-dimensional data sets for data integration," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 475–480.

[16] Cora dataset. [Online]. Available: http://www.cs.umass.edu/~mccallum/data/cora-refs.tar.gz

[17] T. H. Davenport and L. Prusak, *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, 1998.

[18] X. Dong, A. Y. Halevy, and J. Madhavan, "Reference reconciliation in complex information spaces," in *Proceedings of the 24th ACM SIGMOD International Conference on Management of Data*, 2005, pp. 85–96.

[19] Electronic theses and dissertation system. [Online]. Available: http://etds.ncl.edu.tw

[20] I. P. Felligi and A. B. Sunter, "A theory for record linkage," *Journal of the American Statistical Society*, vol. 64, pp. 1183–1210, 1969.

[21] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," in *Proceedings of National Academy of Science*, 2002, pp. 7821–7826.

[22] F. Jelinek and R. Mercer, "Interpolated estimation of markov sourceparameters from sparse data," in *Workshop on Pattern Recognition in Practice*, 1980.

[23] J. Kleinberg, "Authoritative sources in a hyperlinked environment," in *Proceedings of 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998, pp. 604–632.

[24] V. Lavrenko and W. B. Croft, "Relevance based language models," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001, pp. 120–127.

[25] S. Lawrence, C. L. Giles, and K. D. Bollacker, "Autonomous citation matching," in *Proceedings of the 3rd International Conference on Autonomous Agents*, 1999, pp. 392–393.

[26] C. Macdonald and ladh Ounis, "Voting for candidates: Adapting data fusion techniques for an expert search task," in *Proceedings of the 15th ACM Conference on Information and Knowledge Management*, 2006, pp. 387–396.

[27] M. Maron, S. Curry, and P. Thompson, "An inductive search system: Theory, design, and implementation," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 16, no. 1, pp. 21–28, 1986.

[28] A. McCallum, K. Bellare, and F. Pereira, "A conditional random field for discriminatively-trained finite-state string edit distance," in *Proceedings of 21st Conference on Uncertainty in Artificial Intelligence*, 2005, pp. 388–395.

[29] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Information Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.

[30] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, pp. 169–178.

[31] H. Pasula, B. Marthi, B. Milch, S. J. Russell, and I. Shpitser, "Identity uncertainty and citation matching," in *Proceedings of Advances in Neural Information Processing Systems*, 2002.

[32] D. Petkova and B. W. Croft, "Hierarchical language models for expert finding in enterprise corpora," in *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, Washington, DC, USA, 2006, pp. 599–608.

[33] F. Song and W. Croft, "A general language model for information retrieval," in *Proceedings of the 8th international conference on Information and knowledge management*, 1999.

[34] F. Song and W. Croft, "A general language model for information retrieval," in *Proceedings of the 8th International Conference on Information and Knowledge management*, 1999, pp. 316–321.

[35] TREC. Enterprise Track 2005. [Online]. Available: http://www.ins.cwi.nl/projects/trec-ent/wiki/

[36] C. van der Walt and E. Barnard, "Data characteristics that determine classifier performance," in *Proceedings of Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*, 1992.

[37] J. Wang, Z. Chen, L. Tao, W.-Y. Ma, and L. Wenyin, "Ranking user's relevance to a topic through link analysis on web logs," in *Proceedings of the 4th International Workshop on Web Information and Data Management*, 2002, pp. 49–54.

[38] S. Wasserman and K. Faust, *Social Network Analysis: methods and applications*. Cambridge University Press, 1994.

[39] I. Witten and T. Bell, "The zero-frequency problem: estimating the probabilities of novelevents in adaptive text compression," *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 1085–1094, 1991.