Online Social Network Profile Linkage Based on Cost-Sensitive Feature Acquisition

Haochen Zhang[1], Minyen Kan[2], Yiqun Liu[1], and Shaoping Ma[1,⋆]

[1] State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, China
[2] Web, IR / NLP Group (WING)
Department of Computer Science, National University of Singapore, Singapore
zhang-hc10@mails.tsinghua.edu.cn
kanmy@comp.nus.edu.sg
{yiqunliu,msp}@tsinghua.edu.cn

Abstract. Billions of people spend their virtual life time on hundreds of social networking sites for different social needs. Each social footprint of a person in a particular social networking site reflects some special aspects of himself. To adequately investigate a user's preference for applications such as recommendation and executive search, we need to connect up all these aspects to generate a comprehensive profile of the identity. Profile linkage provides an effective solution to identify the same identity's profiles from different social networks.

With various types of resources, comparing profiles may require plenty of expensive and time-consuming features such as avatars. To boost the online social network profile linkage solution, we propose a cost-sensitive approach that only acquires these expensive and time-consuming features when needed. By evaluating on the real-world datasets from Twitter and LinkedIn, our approach performs at over 85% F_1-measure and has the ability to prune over 80% of the unnecessary feature acquisitions, at a marginal cost of 10% performance loss.

Keywords: social media, user profiles, profile linkage, cost-sensitive.

1 Introduction

Online social network is the most important part of the cyber-life, where netizens share their lives, express their opinions, communicate with their friends and business partners. People use more than one social network to satisfy different social needs of sharing, reading, discussing and communicating. He may

[⋆] This work was supported by Natural Science Foundation of China (61073071). Part of the work was done at the Tsinghua-NUS NExT Search Centre, which is supported by the Singapore National Research Foundation & Interactive Digital Media R&D Program Office, under research grant R-252-300-001-490.

H.-Y. Huang et al. (Eds.): SMP 2014, CCIS 489, pp. 117–128, 2014.

communicate his friends in Facebook, post his comments in Twitter, show his life in Instagram and connect to his business in LinkedIn. To picture a person completely, especially for executive search and recommendation, it is very important to cover all aspects of the person's virtual footprints. Therefore, finding an effective solution to identify users for the same identity has high attractiveness in academic study and commercial value in business.

The similar task, record linkage, has been investigated in traditional database research area for decades. There is also several related work addressing the problem in social network perspective recently[1–7]. However, these approaches barely apply to the large-scale dataset and fail to consider the difficulty dealing with the time-consuming and expensive feature acquisitions. In this paper, we propose an effective and efficient approach taking both features directly extracted from profiles and expensive features acquiring cost-sensitively.

The remainder of this paper is structured as follows. We first describe the related work that informs our task in the next section. In Section 3, we define our problem and describe our analysis of online social network user profiles. This motivates our chosen method to maximally leverage well-populated attributes in profiles for profile linkage, which we present in Section 4. In Section 5, we evaluate our approach on it to examine both effectiveness and efficiency.

2 Related Work

2.1 Profile Linkage across Social Networks

Although profile linkage problem just rises along with the booming development of online social networks, the related task record linkage, also named as entity resolution, has been well studied in traditional database area, including named attributes computations[8, 9], schema mapping for heterogeneous data structures[10–12], probabilistic linkage models[13] and duplicate detection for hierarchical-structured data[14].

Inherited from record linkage task, several work addresses the profile linkage task by applying the intuitive attribute comparison approaches into social network occasions[1–4, 15]. Liu *et al.* [16] and Zafarani *et al.* [7] carefully investigate behaviors of how a user generate his username, and then discover user's characteristics to identify the same individual. Besides attributes comparison, Narayanan *et al.* [5] and Bartunov *et al.* [6] rely on social connections and settle identification by exploiting the assumption that a person has similar social circles across different web sites.

However, these work is not based on the real world dataset, which ignores the problems of time-consuming and expensive feature acquisition procedures. When dealing with large-scale data, the enormous cost has to be considered.

2.2 Cost Sensitive Feature Acquisition

Traditional linkage tasks usually gather all attributes locally and features are easily generated. Thus the cost of acquiring and computing features is omitted.

However, both employing web services results acquiring external resources from web and in extremely high cost comparing to local similarity computation.

Several approaches are investigated to deal with missing attribute values acquisition [17–19]. Lin *et al.* [20] improved probabilistic K-NN with acquiring attribute values of uncertain data objects. Tan *et al.* [21] proposes a hierarchical cost-sensitive approach to acquire search engine results with hierarchical dependencies. However, user profile linkage task acquires various types of features with both time cost and usage limitations. These approaches are not designed to solve the profile linkage tasks in which we should consider the hybrid cost controlling.

3 Motivation

3.1 Profile Linkage

Identity refers to a unique entity, such as individual people, groups and companies, which is usually identifiable in the real-world. **Profile** refers to a particular social network's account for the identity, which consists of **attributes** with values. In different social networks, an identity may register several accounts to cover different social applications. Intuitively, profiles from the same identity should be quite similar to each other. **Profile linkage** is then defined as the task that discovers profiles projected from the same identity. Similar to other linkage tasks, profile linkage has two kinds of solutions: 1) clustering profiles for a certain identity; 2) comparing each pair of profiles to determine whether they belong to the same identity.

We address the profile linkage problem by comparing candidate profile pairs, denoted as *pairwise profile linkage*. Notice that there exist transitivity conflicts when involving more than two social networks. In this paper, we only solve the case of two social networks and leave the transitivity conflicts in future work.

The online social network profile linkage faces unique characteristics, including semi-structured data, multimedia resources, privacy and so on. Therefore, attributes in profiles are often sparse and arbitrary. Meanwhile, profiles in different OSN often prefer different attributes. As an example, Twitter is most public and similar to most other closed social media (i.e., FaceBook), where people share their personality to attract followers.

To take full advantage of online social network profiles, we adopt two external features to solve the problem of sparse and arbitrary features: 1) Geocode is a kind of structured locality information, which is much more precise than comparing textual location. Google Maps API provides web service to convert strings to geocodes. 2) Avatar is the most common multimedia resources in profile. A person may use same portrait across different social networks, which provides a very strong evidence when distinguishing between people with the same name.

3.2 Cost-Sensitivity

Since adopting time-consuming features such as geocode and avatar, we face a trade-off between the feature acquisition cost and the classifier performance.

The acquisition of the complicated features could be very time-consuming. The web services even have limitations or payments for the usage. The expensive feature acquisitions motivate the cost-sensitive approach that carefully selects the most distinguishable features with less cost. We regard the selective feature acquisitions that reduce the time and network cost as the micro-level cost-sensitivity.

In this work, the object of the approach is to achieve better results with less time-consuming within the given usage quotas. To every comparison having available extra feature, we define that the comparison benefits if it is re-classified to be correct by given certain extra feature. Therefore, we predict the expected benefit of a comparison in unit time-consuming and adopt the expectation as the criteria of selecting the most effective feature.

4 Approach

To solve the online social network profile linkage, we propose an indexing framework. We index all profiles by tokens extracted from usernames. The tokens are the continuous letters or digit sequences separated by spaces or symbols. Based on Liu *et al.* [16]'s survey, 79% users prefer same username across different communities. Our investigation also shows that 96.1% matched profiles are connected by at least one token. Therefore, two profiles are very unlikely to be matched without a shared token, which helps to prune unnecessary pair-wise comparisons.

Afterwards, we retrieve all pairs of profiles that share at least one token and adopt a probabilistic classifier to determine whether the given two profiles are from the same identity.

4.1 Probabilistic Classifier

The probabilistic classifier is employed to estimate the probability of whether two given profiles q and t are linked (denoted as $l_{q,t} = \{0, 1\}$), by given similarity features $F_{q,t}$ and shared tokens $M_{q,t}$. By assuming the similarity features and shared tokens are independent of each other, we have:

$$p_{q,t} = Pr(l_{q,t}|F_{q,t}, M_{q,t}) = \frac{Pr(l_{q,t}|M_{q,t}) \times \prod_{f_k \in F_{q,t}} Pr(f_k|l_{q,t})}{\prod_{f_k \in F_{q,t}} Pr(f_k)} \tag{1}$$

where $Pr(l_{q,t}|M_{q,t})$ is approximately calculated by:

$$\hat{Pr}(l_{q,t} = 1|M_{q,t}) = \frac{1}{|\bigcap_{m \in M_{q,t}} D_m| + \beta} \tag{2}$$

where D_m is all profiles indexed by token m and β is a smoothing factor preventing $Pr(l_{q,t}|M_{q,t})$ from being 1. We set $\beta = 0.5$ empirically in our experiments.

By applying $Pr(l_{q,t} = 0|\cdot) + Pr(l_{q,t} = 1|\cdot) = 1$ to Equation 1, the equation is derived:

$$p_{q,t} = \frac{1}{1 + (|\bigcap_{m \in M_{q,t}} D_m| + \beta - 1) \times \prod_{f_k \in F_{q,t}} \frac{Pr(f_k|l_{q,t} = 0)}{Pr(f_k|l_{q,t} = 1)}} \tag{3}$$

where $\hat{Pr}(f_k|l_{q,t})$ is estimated by kernel density estimator [22, 23]. Finally, q and t are regarded as matched if $p_{q,t} > 0.5$.

4.2 Features

CSPL uses features discovered from user's profile to conduct the supervised linkage. These features are consist of local features extracted directly from profile attributes, and external features acquired by web services or web resources. To estimate the benefit for cost-sensitive acquisition, all features are normalized to a range of $[0, 1]$. Table 1 lists all the involved local features and how they are computed.

Table 1. Local similarity features extracted directly from profiles

Feature	Description
user_sim	Jaro Winkler distance between two usernames
language	Jaccard similarity of the written or spoken languages
description	vector-space model cosine similarity of user's biography
URL	cosine similarity of the URL tokens (split by symbols)
popularity	$\frac{\|friend_q - friend_t\|}{\|friend_q + friend_t\|}$ where $friend_u$ is the counts of user u's friends

Besides these easily acquired local features, we also employ two time-consuming and usage-limited features: **avatar** and **geocode** as discussed in section 3:

1. **Avatar** is user uploaded image, given as a URL in the profile. We employ a gray-scale χ^2 dissimilarity, a bin-by-bin histogram difference by [24], to compare avatars. This method has been reported effective for texture classification, and represented as:

$$F_{avatar} = \frac{1}{2} \sum_{i \in Bins} \frac{(H_{q,i} - H_{t,i})^2}{(H_{q,i} + H_{t,i})} \tag{4}$$

   where $H_{q,i}$ and $H_{t,i}$ is the ith bin of the image's gray-scale histograms.
2. **Geocode** is the structured information with lat-long coordinates. We access Google Maps API to convert textual location into geographic coordinates, and then calculate spherical distance d in kilometers for comparison. At last, we use $e^{-\gamma d}$ to normalize the distance within $[0, 1]$ with $\gamma = 0.001$ in our experiments.

4.3 Cost-Sensitive Feature Acquisition

Note that some of our features are externally acquired. For example, obtaining the `Geocode` requires API invocations, while obtaining a users' `Avatar` requires a separate resource request. These external features are expensive to acquire as they incur both network delays and bandwidth, and are much more costly than computation over local features. We wish to manage these costs, so as not to use external resources when the local evidence already overwhelmingly supports a linkage decision.

In fact, compared to to acquiring external features, the cost of computation over all local features is negligible. In our classifier, we thus first utilize all of the local features in the beginning, and iteratively choose the instances that are most probable to be improved by adopting a new external feature. We employ a cost-sensitive classifier derived from Naïve Bayes to prune these unnecessary network operations.

Let $\hat{p}$ denote the probability distribution estimated by the existing set of features, and the $\hat{p}^{+k'}$ be the posterior probability when conditioned on the additional feature k'. Let $f_{k'}$ be value of the extra feature, and we derive the relationship between $\hat{p}$ and $\hat{p}^{+k'}$:

$$\hat{p}^{+k'} = \frac{1}{1 + \dfrac{1-\hat{p}}{\hat{p}} \times \dfrac{Pr(f_{k'}|l=0)}{Pr(f_{k'}|l=1)}} \tag{5}$$

To efficiently improve linkage performance, we need to acquire the k' feature that is most effective. Here, we assume that adding features improves performance at the entire dataset level. We estimate the benefit of acquiring a prospective feature by its utility to raise the certainty of the classifications, either for a match or a non-match:

$$(\hat{p} - 0.5)(\hat{p}^{+k'} - 0.5) \le 0 \tag{6}$$

By solving the inequality, we can restate the post-condition of the classification:

$$\begin{cases} g_{k'} < \hat{p}^{-1} - 1, & \hat{p} > 0.5 \\ g_{k'} > \hat{p}^{-1} - 1, & \hat{p} \le 0.5 \end{cases} \tag{7}$$

where $g_{k'}$ represents the ratio $\dfrac{\hat{Pr}(f_{k'}|l=1)}{\hat{Pr}(f_{k'}|l=0)}$ for convenience.

Note that $f_{k'}$ is unknown and cannot be computed directly ahead of acquisition. We therefore must estimate the probability that $g_{k'}$ satisfies the condition, given $\hat{p}$, which is difficult to compute accurately as the distribution of $g_{k'}$ is unknown. We thus need to develop an approximation method.

Notice that we have already estimated $Pr(f_{k'}|l)$ by using the kernel density estimator during training. Furthermore, all our features have values in normalized range of $[0,1]$. We sample s points $\Delta_1, \Delta_2, \cdots, \Delta_s$ within $[0,1]$ following the distribution of $Pr(f_{k'})$ estimated during training using the kernel density estimation. We then compute the corresponding estimation $\hat{g_{k'}}_j|_{f'=\Delta_j}$. Suppose that $r_{p,k'}$ is the rank of value $\hat{p}^{-1} - 1$ in the ordered list of $\{\hat{g}_{k'}\}$, we can then compute the approximate benefit of acquiring feature k' with equation:

$$E_{q,t}^{k'} = \hat{Pr}(\text{benefit}|p, k') = \begin{cases} \dfrac{r_{p,k'}}{s}, & p > 0.5 \\ 1 - \dfrac{r_{p,k'}}{s}, & p \le 0.5 \end{cases} \tag{8}$$

where $E_{q,t}^{k'}$ is the expectation of benefit given $\hat{p}$ and k'.

In practice, acquiring different features has different time costs. This adds another dimension of complexity to our feature acquisition process, as features have different per time-unit benefit. If the time cost of acquiring feature k' is $c_{k'}$, the *per time-unit feature benefit expectation* of comparison with probability p and added feature k' is $\frac{Pr(\text{benefit}|p, k')}{c_{k'}}$.

We conclude the *per time-unit benefit expectation* of the given comparison $\langle q, t \rangle$ to be:

$$E_{q,t} = \max_{k' \in \mathcal{K}_{q,t}} \frac{E_{p_{q,t}}^{+k'}}{c_{k'}} \tag{9}$$

where $\mathcal{K}_{q,t}$ represents the set of acquirable features of the comparison $\langle q, t \rangle$. By acquirable, we mean that a feature meets the following criteria: 1) have not been acquired; 2) exist in both profiles; 3) its acquisition will not exceed a quota (e.g., a API daily limit). The most effective external feature is the one that maximizes the benefit expectation.

5 Experiment

We set up experiments on linking 150,000 users across Twitter and LinkedIn to evaluate the performance of our linkage approach and the efficiency and effectiveness of the cost-sensitive feature acquisition method.

5.1 Linkage Performance

To evaluate the classifier's performance, we crawled a realistic profile dataset from Twitter and LinkedIn. The Twitter profiles are sampled from tweets posted between 9 Oct. 2012 and 16 Oct. 2012. The LinkedIn profiles are sampled from the directory[1]. In total, we obtained 152,294 Twitter profiles by RESTful API and 154,379 LinkedIn profiles by parsing user profile pages, which are all publicly available.

To discover the relationship between each LinkedIn and Twitter profiles, we employ third party websites Google+ that encourages users to reveal their OSN profiles. We generate the ground truth with the assumption that all corresponding OSN accounts filled by one user belong to themselves. We crawled 180,000 Google+ profiles and extract the overlapping users of our dataset and the Google+ profiles. This partial ground truth contains 9,750 identities: 4,779 matched Twitter–LinkedIn users, 3,339 singular Twitter users and 1,632 singular LinkedIn users.

Besides the standard IR metrics: Precision (*Pre*), Recall (*Rec*) and F_1-measure (F_1), we employ the identity-based accuracy (*I-Acc*), representing as:

$$I\text{-}Acc = \frac{\text{correctly identified identities}}{\text{all ground truth identities}}$$

[1] `http://www.linkedin.com/directory/people/a.html`

We use simple classifiers like C4.5, SVM and Naïve Bayes as the base-line, which have been reported effective in [2]. In our experiment, we choose Twitter as the query dataset and LinkedIn as the target. To generate the training set, We randomly sampled 1,000 query instances and all the corresponding targets. To evaluate the classifier performance adequately, we include all features in this experiment. Table 2 shows that our approach CSPL make the best performance both in F_1 and *I-Acc*. Furthermore, our approach make a significant improvement in recall with slight loss in precision, which discovers more linking relationship between candidate pairs.

Table 2. Linkage performance over our Twitter→LinkedIn dataset with all features

Method	*Pre*	*Rec*	F_1	*I-Acc*
C4.5	0.905	0.658	0.762	0.806
SVM	0.942	0.456	0.614	0.727
Naïve Bayes	0.934	0.625	0.748	0.801
CSPL	0.866	**0.846**	**0.856**	**0.865**

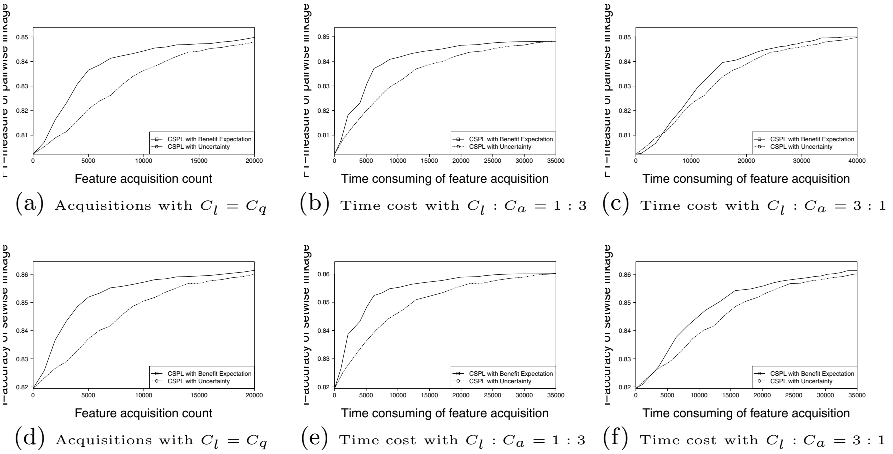
5.2 Cost-Sensitive Feature Acquisition

CSPL is also designed to optimally control for cost in acquiring external features. We denote our cost-sensitive approach based on benefit expectation as described in Section 4.3 as CSPL_BE.

Table 3. Cost-sensitivity with different pseudo time cost settings

	$C_l = C_a = 1$		$C_l = 1, C_a = 3$		$C_l = 3, C_a = 1$	
#I	Acq	Time	Acq	Time	Acq	Time
30%	2,000	2,000	2,000	2,110	4,000	8,273
60%	5,000	5,000	5,000	6,224	6,000	13,202
90%	32,000	32,000	32,000	79,072	32,000	45,824
	Time (Max Acq = 188,590)					
100%	188,590		439,880		314,480	

Since CSPL_BE is based on the time-unit benefit expectation, it is sensitive to different time cost settings. To comprehensively evaluate the performance in different time cost settings, we set up and investigate three pseudo time settings to simulate possible cases: $\langle C_l, C_a \rangle = \langle 1, 1 \rangle$, $\langle C_l, C_a \rangle = \langle 1, 3 \rangle$ and $\langle C_l, C_a \rangle = \langle 3, 1 \rangle$, where C_l and C_a is the time cost of *Geocode* and *Avatar* respectively. The experiment results are sampled per 2,000 acquisitions. To make the results comparable, we set three checkpoints to estimate the approximate number of acquisitions and time cost to the nearest sample. Table 3 gives pseudo time costs over three checkpoints at 30%, 60% and 90% of all external feature acquisitions. Coupled with the results from Figure 1, we see that CSPL_BE achieved 90% of the remaining performance improvement that would be achieved by acquiring all external features, by merely acquiring 17% additional features and between 15–18% more time (depending on cost settings).



(a) Acquisitions with $C_l = C_q$ (b) Time cost with $C_l : C_a = 1 : 3$ (c) Time cost with $C_l : C_a = 3 : 1$

(d) Acquisitions with $C_l = C_q$ (e) Time cost with $C_l : C_a = 1 : 3$ (f) Time cost with $C_l : C_a = 3 : 1$

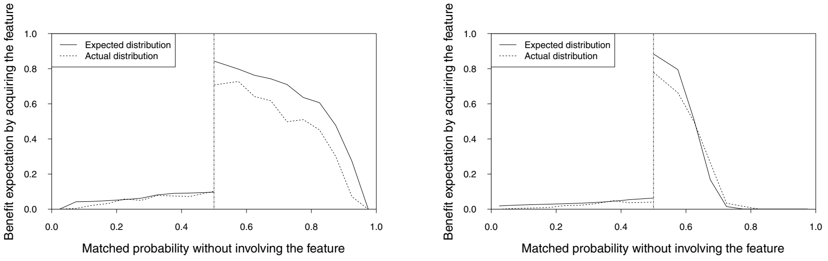
Fig. 1. Performance with different feature acquisition cost by given pseudo time cost. Figures in the first row are evaluated by F_1-measure and ones in the second row are evaluated by I-accuracy.

To specifically evaluate our benefit estimation, we need to compare with other forms to estimating the utility of yet unacquired features. We establish a baseline (CSPL_UN) that acquires the minimal cost feature of the most uncertain comparisons. The uncertainty of a comparison is defined as $1-2\times|\hat{p}-0.5|$, where $\hat{p}$ is the previous estimated probability of the comparison. Figure 1 displays this comparison for the three different pseudo time cost settings and two evaluation metrics. Note that acquiring only about 15% of the external is effective enough. Thus we only investigate the subfigure in which the performance has not reached stable status yet. Illustrated by the figures, our approach represented in solid line increases faster than the baseline represented in dashed line despite different time cost settings, which indicates that our approach performs better with the same feature acquisitions or time cost.

5.3 Benefit Expectation

Our cost-sensitive feature acquisition strategy relies on the estimation to the benefit expectation. We conduct this experiment to evaluate whether our algorithm is able to correctly simulate the expectation before actual feature acquisition.

We evenly split $[0, 1]$ into N buckets and assign the midpoint as the representative probability $\tilde{p}_i$ to the i-th bucket. We first initial the matched probability p of each comparison computed by local features. Each comparison is then dispatched into a certain bucket according to p. We acquire every external feature individually for these comparisons and get updated results. The percentage of beneficial instances within each bucket is then regarded as the actual benefit expectation to the representative probability $\tilde{p}_i$. Meanwhile, we have the set of probabilities $\tilde{p}_i$ and then compute their corresponding $E_p^{k'}$ without actually involving the new features.



(a) Feature *Geocde* (b) Feature *Avatar*

Fig. 2. Evaluation of the benefit expectation algorithm

In Figure 2, solid lines represent the prediction of the beneficial comparisons calculated by our algorithm, while dashed lines represent the actual expectation observed by actually involving new features. Figure 2 ((a)) and ((b)) corresponding to feature *Geocode* and *Avatar* respectively. Figure 2 illustrates that estimation and actual results are consistent, which indicates that our algorithm correctly estimates the benefit expectation.

In addition, we observe that the instances classified as positives, i.e. potential false positives, are more likely to be corrected than false negatives. It is mainly caused by the unbalanced dataset where there are too many negative instances. Meanwhile, the *Geocode* has a better benefit expectation with the same probability comparing to it Avatar, which indicates that *Geocode* is more effective in improving performance. It provides an intuitive way to compare the effectiveness between different features.

6 Conclusion

We investigate the profile linkage problem and propose a cost-sensitive probabilistic approach to reduce time consuming feature acquisitions. To effectively acquire external features, we establish an approximate algorithm to estimate the benefit of involving a new feature with performance time unit cost. The strategy is also able to satisfy the limitation of feature quota.

Our experiment results show the effectiveness of our approach with 85% F_1-measure and 86% I-accuracy compared to base-line. Our cost-sensitive framework also has the ability to prune the unnecessary network acquisitions for external features while keeping the performance loss in an acceptable level. Indeed, with only 10% loss on F_1-measure, we achieve more than 85% network acquisitions reduction.

In the future work, there are two routines to improve the linkage approach: 1. Improve linkage approach with more than profile resources, such as involving social connections, user generated content and mobile footprints. 2. Adopt the linked identities to a practical application such as recommendation to investigate the improvement caused by adequate user study.

References

1. Carmagnola, F., Cena, F.: User Identification for Cross-system Personalisation. Inf. Sci. 179(1-2) (2009)
2. Malhotra, A., Totti, L., Meira Jr., W., Kumaraguru, P., Almeida, V.: Studying User Footprints in Different Online Social Networks. In: International Workshop on Cybersecurity of Online Social Network (2012)
3. Nunes, A., Calado, P., Martins, B.: Resolving User Identities over Social Networks through Supervised Learning and Rich Similarity Features. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. ACM (2012)
4. Vosecky, J., Hong, D., Shen, V.: User Identification Across Multiple Social Networks. In: Networked Digital Technologies. IEEE (2009)
5. Narayanan, A., Shmatikov, V.: De-anonymizing Social Networks. In: Proceedings of the 2009 30th IEEE Symposium on Security and Privacy. IEEE (2009)
6. Bartunov, S., Korshunov, A., Park, S., Ryu, W., Lee, H.: Joint Link-Attribute User Identity Resolution in Online Social Networks. In: Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, Workshop on Social Network Mining and Analysis. ACM (2012)
7. Zafarani, R., Liu, H.: Connecting Users across Social Media Sites: A Behavioral-modeling Approach. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 41–49. ACM, New York (2013)
8. Cohen, W.W., Ravikumar, P., Fienberg, S.E., et al.: A Comparison of String Distance Metrics for Name-matching Tasks. In: Proceedings of the IJCAI 2003 Workshop on Information Integration on the Web (IIWeb 2003), pp. 73–78 (2003)
9. Christen, P.: A Comparison of Personal Name Matching: Techniques and Practical Issues. In: Proceedings of the 6th IEEE International Conference on Data Mining Workshops, ICDM Workshops. IEEE (2006)
10. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and Ontology Matching with Coma++. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD 2005, p. 906. ACM Press (2005)
11. Nottelmann, H., Straccia, U.: Information Retrieval and Machine Learning for Probabilistic Schema Matching. Information Processing & Management 43(3), 552–576 (2007)
12. Qian, L., Cafarella, M.J., Jagadish, H.V.: Sample-driven schema mapping. In: Proceedings of the 2012 International Conference on Management of Data, SIGMOD 2012, p. 73. ACM Press (2012)
13. Ravikumar, P., Cohen, W.W.: A Hierarchical Graphical Model for Record Linkage. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, pp. 454–461. AUAI Press (2004)

14. Leitão, L., Calado, P., Herschel, M.: Efficient and Effective Duplicate Detection in Hierarchical Data. IEEE Transactions on Knowledge and Data Engineering PP(99), 1 (2012)
15. Irani, D., Webb, S., Li, K., Pu, C.: Large Online Social Footprints–An Emerging Threat. In: Proceedings of the International Conference on Computational Science and Engineering. IEEE (2009)
16. Liu, J., Zhang, F., Song, X., Song, Y.I., Lin, C.Y., Hon, H.W.: What's in A Name?: An Unsupervised Approach to Link Users Across Communities. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining. ACM (2013)
17. Ji, S., Carin, L.: Cost-sensitive Feature Acquisition and Classification. Pattern Recognition 40(5), 1474–1485 (2007)
18. Ling, C.X., Sheng, V.S., Yang, Q.: Test strategies for cost-sensitive decision trees. IEEE Trans. on Knowl. and Data Eng. 18(8), 1055–1067 (2006)
19. Saar-Tsechansky, M., Melville, P., Provost, F.: Active feature-value acquisition. Manage. Sci. 55(4), 664–684 (2009)
20. Lin, Y.C., Yang, D.N., Chen, M.S.: Selective Data Acquisition for Probabilistic K-nn Query. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1357–1360. ACM (2010)
21. Tan, Y.F., Kan, M.Y.: Hierarchical cost-sensitive web resource acquisition for record matching. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 01, pp. 382–389. IEEE Computer Society (2010)
22. Epanechnikov, V.: Non-Parametric Estimation of a Multivariate Probability Density. Theory of Probability & Its Applications 14(1), 153–158 (1969)
23. John, G.H., Langley, P.: Estimating Continuous Distributions in Bayesian Classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (1995)
24. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. Int. J. Comput. Vision 73(2) (2007)