Online Social Network Profile Linkage

Haochen Zhang[1], Min-Yen 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

Abstract. Piecing together social signals from people in different online social networks is key for downstream analytics. However, users may have different usernames in different social networks, making the linkage task difficult. To enable this, we explore a probabilistic approach that uses a domain-specific prior knowledge to address this problem of online social network user profile linkage.

At scale, linkage approaches that are based on a naïve pairwise comparisons that have quadratic complexity become prohibitively expensive. Our proposed threshold-based canopying framework – named OPL – reduces this pairwise comparisons, and guarantees a upper bound theoretic linear complexity with respect to the dataset size.

We evaluate our approaches on real-world, large-scale datasets obtained from Twitter and Linkedin. Our probabilistic classifier integrating prior knowledge into Naïve Bayes performs at over 85% F_1-measure for pairwise linkage, comparable to state-of-the-art approaches.

Keywords: social media, user profiles, canopies, profile linkage

1 Introduction

The Online Social Network (OSN) is a ubiquitous feature in modern daily life. People access social networks to share their stories and connect with their friends. Many netizens commonly participate in multipl social networks, to cover their social needs of reading, researching, sharing, commenting and complaining. There are a wealth of choices available to link their real-world social networks virtually, and to extend and enhance them online.

The variety of social networks are partially redundant but each has a niche focus that can provide different slants on an individual's virtual lifestyles. People

* 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.

may communicate with friends in Facebook, share their opinions in Twitter, exhibit artistic photographs in Flickr and maintain business relationships in LinkedIn. Rarely do individuals use a single OSN to cover all such facets. As such, studies that seek to understand the virtual netizen that capture a user's participation from only a single OSN will necessarily have a strong bias. To gain a holistic perspective, an understanding of an online individual is derivable from piecing together all of the myriad aspects of his online footprints. Furthermore, when the same users posts the same opinions in different OSNs, such *user linkage* is needed, to avoid double-counting and to accurately estimatef social signals.

Variants of this problem – commonly referred to as *record linkage* – have been investigated in the database community for decades. Relatively recent work has re-examined this problem in light of linkage within OSNs. Many works [4, 13, 15, 19] adapt standard supervised machine learning for linkage. These works focus on accuracy and rely on pairwise comparisons (i.e., Is profile x in OSN a the same as profile y in OSN b?). However these works are infeasible to apply to large-scale real-world datasets: as they do not address how to deal with the unbalanced ratio of positive to negative instances, and the enormous number of necessary pairwise comparisons. [12] carefully studies usernames and proposes a prior knowledge approach which significantly improves name-disambiguation performance in the user linkage task. [14, 3] leverage the graphical structures of social networks and [20] identifies behavioral features extracted from usernames and user-generated content, both noting that external evidence can help in the linkage task. However, acquiring both forms of external evidence may be expensive, or even inaccessible. In summary, while the prior work we have surveyed here have introduced methods or features for the user linkage problem, few address the difficulties with the necessary quadratic complexity of pairwise comparison. None have reported their results on identifying individuals in real-world large-scale datasets.

In this paper, we address this problem of *large-scale online social network user profile linkage*. We investigate and optimally tune known techniques for record linkage, by applying them to the user profile linkage problem for the purpose of large-scale production use. The key contribution of our work is to link an individual user's user profiles together, exploiting the idiosyncrasies of the problem to achieve accurate, time-efficient and cost-sensitive linkage.

2 Related Work

2.1 Identifying Users across Social Networks

User profile linkage is a research area that has developed in parallel with the development online social networks. At its core, methods compare the similarity between two users' profiles (often, one from one social network and one from another) by carefully investigating their attributes [4, 13, 15, 19]. Vosecky *et al.* [19] and Carmagnola *et al.* [4] proposed linear threshold-based models, which combine each features with weights and determine whether they belong to one identity by comparing to the preset threshold. Malhotra *et al.* [13] and Nunes *et al.* [15] adopt supervised classification to decide on matching.

Aside from attribute comparison, Narayanan *et al.* [14] and Bartunov *et al.* [3] leverage a user's social connections to identify their OSN accounts. The former demonstrated that users can be de-anonymized without personal information, by exploiting the fact that users often have similar social connections in different OSNs. Bartunov *et al.* similarly reported that modeling the user graph improves performance by re-identifying users with similar relationship structures.

Several works also aim to disambiguate users of the same name ("namesake users"), a subtask termed *name disambiguation* [1,18]. Zafarani *et al.* [20] explored how the behavior features of how users express themselves and generate their usernames. Perito *et al.* [16] studied username choice discloses our identities to public, while Liu *et al.* [12] improves name disambiguation by modeling the commonality of usernames, to help better estimate the linkage likelihood.

2.2 Record Linkage and Entity Resolution

User profile linkage is similar to traditional record linkage (or *entity resolution*). Surveys [8,9]. review the various approaches, including named attributes computations [5], schema mapping [2,17] and duplicate detection in hierarchical data[10], all which inform the construction of profile linkage techniques.

Both profile linkage and record linkage face the computational complexity problem. A key insight to reducing practical complexity is to note that many user profile pairs are highly disparate, and unnecessary to compare. Indexing techniques can then be used to find rough clusters for which expensive pairwise comparison can be applied [6]. Canopying [7] is one such techniques, setting up soft constraints to form overlapping clusters (canopies), and only comparing instances within each canopy.

3 Problem Definition

We first define the associated terminology and then formalize the problem of profile linkage:

Identity refers to a unique entity, usually identifiable in the real-world context. Identities usually correspond to individual people, but other physical and virtual entities – such as bands, companies and products – are also possible. The current U.S. president, Barack Obama, is an example of an identity.

Profile refers to a projection of an identity into a particular social network. A profile is a data structure, consisting of a set of **attributes** with values and implicitly belongs to its identity. Identities may participate in multiple social networks, and thus project a profile for each network. For example, currently, Barack Obama is `barackobama` on LinkedIn and Facebook, `BarackObama` on Twitter, and `+BarackObama` on Google Plus.

Intuitively, profiles that are projected from the same identity should have high similarity with each other. Returning to our example, We can see three of Barack Obama's profiles use the same user ID (ignoring capitalization). Profile linkage hinges on this assumption of similarity.

Profile linkage is thus the matching task of determining which profiles are projections of the same identity.

More formally, let $\mathcal{I} = \{I_i\}$ be set of identities, $\mathcal{R} = \{R_k\}$ be set of social networks and $\mathcal{P}_k$ denotes set of profiles in the online social network R_k. I_i has all his profiles $P_{k,i}$ in social network R_k where $P_{k,i} \subset \mathcal{P}_k$. Note that identities ($I_i$) are not observed, so we must infer whether its projections $\hat{I}_i$ in the observed online social networks are linked and represent it, where $\hat{I}_i = \bigcup_{\forall R_k \in \mathcal{R}} P_{k,i}$.

We define the *setwise profile linkage* problem as the task of fully recovering the set of $\hat{I}_i$ given online social networks $\mathcal{R}$ and profiles $\mathcal{P}_k$ of each social network k. At a smaller scale, the *pairwise profile linkage* problem is the task of determining whether two profiles are projected from the same identity.

By repeatedly solving the *pairwise profile linkage* problem for all profile pairs from any two social networks and resolving all transitivity conflicts, we solve the *setwise profile linkage task.* Notice that if there are only two social networks, *setwise profile linkage* reduces to *pairwise profile linkage.*

We address the *pairwise profile linkage* for the case of two social networks. For each query profile in OSN R_α, we retrieve a set of similar n target profiles from R_β, and determine (if any) of the P_β link; i.e., originate from the same identity i that generated the query profile P_α. In this paper, we additionally assume that each identity only projects at most a single profile per OSN.

4 Approach

Given the large-scale and reliance on external data, our OPL ("Online Profile Linkage") approach to profile linkage must consider computation cost at the core. OPL addresses the cost-sensitivity by controlling local computation by employing *canopies* to prune unnecessary pairwise comparisons.

OPL takes an indexing approach to accomplish setwise profile linkage. To avoid redundant comparisons, we sequentially traverse the two pending OSNs, by regarding one as a query profile source and the other as the target to be considered for linking. Our approach is symmetric, as either OSN can be treated as the query source.

4.1 Token-based Canopies

To construct our canopies, we use tokens from usernames and names, as these are ubiquitous sources common in all OSNs. Then we index these profiles by corresponding tokens. Based on our observations, we find that 96.1% matched profiles share at least one token. By "token", we mean continuous letter or digit sequences segmented by intervening spaces or symbols. We make the implicit assumption that two matched profiles must share a common token.

Our detailed examination of tokens shows that they conform to a power law distribution very well (Zipf's Law). Thus, high-frequency tokens do not serve to distinguish truly linked profiles. As such tokens would create canopies of limited use that are large (or equivalently, costly), we filter out high-frequency tokens from consideration: tokens above a frequency threshold θ are discarded.

Canopy Complexity Analysis We prove that token-based canopies yield a linear complexity in this section. Let the size of query profiles $\mathcal{Q}$ be $|Q|$, the size of target profiles $\mathcal{T}$ be $|T|$. Assuming that the set of all tokens is $\mathcal{M}$ and the frequency of token $m \in \mathcal{M}$ is N_m, then the set of tokens (after filtering) is regarded as $M = \{m | m \in \mathcal{M}, N_m \leq \theta\}$.

The total number of comparisons is the summation of each query's candidate profiles retrieved from the canopies:

$$C = \sum_{q \in \mathcal{Q}} \sum_{m \in M_q} |D_m| \tag{1}$$

where $M_q \subset M$ is the query profile q's token set, which is a subset of the tokens M. D_m denotes the set of target profiles indexed by specific token m and $|D_m|$ denotes its size.

Let the profile frequency of token m be $N_{m,Q}$ computed over $\mathcal{Q}$ and $N_{m,T}$ computed over $\mathcal{T}$. Notice that D_m equals to $N_{m,T}$, and that all candidates of m are retrieved $N_{m,Q}$ times and that $N_{m,Q} + N_{m,T} = N_m$. Therefore, we can compute the total number of comparisons C from a token perspective:

$$C = \sum_{m \in M} N_{m,Q} \times N_{m,T} \leq \sum_{m \in M} \frac{1}{4} \times N_m^2 \tag{2}$$

Since the tokens' distribution follow a power law (Zipf's Law), we have:

$$N_m \approx H \times r_m^{-s} \tag{3}$$

where s and H are parameters that characterize the distribution and r_m is the rank of m. Substituting Equation 3 into Equation 2, we derive:

$$C \leq \frac{1}{4} \sum_{m \in M} {N_m}^2 \approx \frac{H^2}{4} \int_{r_\theta}^{\infty} r^{-2s} \mathrm{d}r \tag{4}$$

where r_θ is the rank of the token(s) with frequency θ, which equals to $(H/\theta)^{s^{-1}}$.

By employing linear regression, we estimate approximate value of s to be 1.053, which follows the empirical observations that $s \to 1$ and $H \propto (|Q| + |T|)$ when applied to human language[11]. We derive a final, concise relationship:

$$C \leq \zeta \times \theta \times (|Q| + |T|) \tag{5}$$

where ζ is a constant to ensure equality. We can thus tune θ for a particular application scenario, knowing that we will have a complexity on the order of $O(|Q| + |T|)$, i.e. linear in size of Q and T.

4.2 Feature Selection

OPL uses a simple battery of features for linking in a supervised manner. We employ both local features extracted directly from profile attributes, and (optionally) external features acquired from the Web. All features are normalized to a range of $[0, 1]$ to simplify computation.

Local Features (5 Features) **Username**: Name comparison is a well-studied problem and many fuzzy matching approaches have been designed and evaluated for it. We adopt the Jaro Winker metric, as it been reported to be one of the best performing [5] metrics for name-like feature.

As many identities may have similar or even identical namesakes, the usernames alone are not sufficiently discriminative. When linking across the entire web dataset or treating person names with high namesake conflicts such as Chinese, name disambiguation techniques become more important.

Language: This attribute refers to the language(s) spoken by the user. This attribute is a set of enumerated types, taking on values from a fixed finite set. We employ the Jaccard similarity for this set attribute to compute the feature.

Description: The description is a free-form short text provided by the user, commonly mentioning their associations to organizations, their occupations and interests. We calculate the vector-space model cosine similarity with TF×IDF weighting, a commonly-used standard, for this feature.

URL: Some profile attribute values are URLs, while other URLs can be extracted from free text descriptions (e.g., `descriptions`). URLs pointing to specific pages (i.e., homepages, blogs) can be helpful. We split URLs into tokens, using cosine similarity with TF×IDF weighting of the tokens for comparison.

Popularity: We utilize the profile's friend or connection count. This value reflects the popularity and connectedness of the profile. OSNs often cap the total number of connections that are displayed; so to make two values comparable, we omit counts beyond this maximum limit. We adopt a normalized formula akin to Jaccard set similarity for popularity comparison:

$$F_{popularity} = \frac{|friend_q - friend_t|}{|friend_q + friend_t|} \tag{6}$$

where $friend_q$ ($friend_t$) is the count of friends for profile q (t).

External Features (2 Features) **Location**: Locations come in a variety of forms – detailed addresses, lat-long coordinates, or bare city names – such that standard string similarity fails here. We rely on the Google Maps API (GeoCode) to convert arbitrary locality strings into geographic coordinates, calculating spherical distance d in kilometers for comparison. We employ $e^{-\gamma d}$ to normalize the distance similarity within $[0, 1]$, where the scale parameter γ is assigned to be 0.001.

Avatar: is an image to represent the user, given as a URL in the profile. After downloading the image, we use a gray-scale χ^2 dissimilarity to compare the images. Our implementation is a bin-by-bin histogram difference based [21], which has been proved effective for texture and object categories classification, defined as:

$$F_{avatar} = \frac{1}{2} \sum_{i \in Bins} \frac{(H_{q,i} - H_{t,i})^2}{(H_{q,i} + H_{t,i})} \tag{7}$$

where $H_{q,i}$ and $H_{t,i}$ represent the ith bin of the query profile q and target profile t's image gray-scale histograms.

4.3 Probabilistic Classifier

As previously stated, the token type distribution obeys Zipf's law. This allows us to estimate the utility of a shared token for matching profiles based on its frequency within the collection. A shared rare token gives a larger probability of matching. We codify this evidence into a probablistic model.

To determine whether the query profile q and target profile t are from the same identity, we estimate its probability modeled as conditioned on the joint probabily of the similarity of the features and the set of shared tokens: $Pr(l_{q,t} = 1|F_{q,t}, M_{q,t})$ where $l_{q,t} = \{0, 1\}$ denotes whether q and t are matched, $F_{q,t}$ denotes similarity features and $M_{q,t}$ denotes the shared tokens between q and t.

We make the assumption that the feature similarity and overlapping tokens are independent of each other, yielding:

$$Pr(l_{q,t}|F_{q,t}, M_{q,t}) = \frac{Pr(l_{q,t}|M_{q,t}) \times \prod_{f_k \in F_{q,t}} Pr(f_k|l_{q,t})}{\prod_{f_k \in F_{q,t}} Pr(f_k)} \tag{8}$$

where $Pr(l_{q,t}|M_{q,t})$ is the prior token distribution knowledge, and $Pr(f_k|l_{q,t})$ is the probability of observing feature k, conditioned on profile match or not.

Unfortunately, $Pr(l_{q,t}|M_{q,t})$ is difficult to measure in practice. We estimate it roughly as the fraction of profiles that have all observed tokens in q:

$$\hat{Pr}(l_{q,t} = 1|M_{q,t}) = \frac{1}{|\bigcap_{m \in M_{q,t}} D_m| + \beta} \tag{9}$$

where D_m is all corresponding profiles indexed by token m and β (empirically set to 0.5) is a smoothing factor that prevents $Pr(l_{q,t}|M_{q,t})$ from being 1. By applying the equality $Pr(l_{q,t} = 0|\cdot) + Pr(l_{q,t} = 1|\cdot) = 1$ to Equation 8, we derive:

$$\begin{aligned} &p_{q,t} = Pr(l_{q,t} = 1|F_{q,t}, M_{q,t}) = \\ &\frac{1}{1 + (|\bigcap_{m \in M_{q,t}} D_m| + \beta - 1) \times \prod_{f_k \in F_{q,t}} \frac{Pr(f_k|l_{q,t} = 0)}{Pr(f_k|l_{q,t} = 1)}} \end{aligned} \tag{10}$$

where $\hat{Pr}(f_k|l_{q,t})$ is calculated over the training data. Since the feature counts are sparse, it is difficult to properly model their distribution, we employ the kernel density estimator to estimate the features' distributions. Given these estimates, we thus declare q to match t when $\hat{Pr}(l_{q,t} = 1|F_{q,t}, Mq, t) > 0.5$.

5 Experiment

We set up our experiments on linking over 150,000 users across two well-known social networks: Twitter and LinkedIn. We aim to answer the following questions: (1) How well does our approach perform on the real world large-scale dataset compared to other state-of-the-art approaches? (2) How does the setting of the canopy threshold θ practically impact performance and efficiency?

5.1 Dataset and Evaluation metric

We describe our approach to construct a realistic dataset for the profile linkage problem. We consider the problem of linking user profiles from Twitter and LinkedIn. We first collected tweets from Twitter for one week, 9–16 October 2012. Then we sampled 152,294 Twitter users from these tweets and downloaded their profiles. LinkedIn users are randomly sampled from LinkedIn directory[1]. In total, we obtained 154,379 LinkedIn user profiles.

It is impossible to obtain the full ground truth for the dataset, short of asking each tweeter to disclose their LinkedIn profile. Instead we use public data already provided in third party websites, such as About.me and Google+, which encourages users to manually submit their OSN profiles' links. We assume that all social network accounts filled by one user belong to himself. We randomly crawl 180,000 Google+ profiles and extract this partial ground truth from the overlapping users of our dataset and the Google+ profiles.

The partial ground truth includes 4,779 matched Twitter–LinkedIn users, 3,339 isolated Twitter users and 1,632 isolated LinkedIn users, a total of 9,750 identities. We adopt this partial ground truth to estimate the performance.

We employ the standard IR evaluation metrics: Precision (Pre), recall (Rec) and F_1-measure (F_1) to evaluate the pairwise linkage. We also report the identity-based accuracy (*I-Acc*), which the accuracy of setwise linkage restricted to true positive matches (i.e., correctly identified identities divided by the total number of ground truth identities).

5.2 User Profile Linkage

We apply several approaches to link the Twitter–LinkedIn dataset using a canopy framework. To the best of our knowledge, no related work has attempted the linkage of complete OSN profiles on two real-world large-scale datasets. Both Bartunov *el at.* [3] and Vosecky *el at.* [19] executed experiments on a small-scale datasets. In related work, Liu *el at.* [12] focuses only on disambiguating profiles with identical username namesakes and Malhotra *el at.* [13] studied the linkage effectiveness on an artificial dataset. No works have yet to benchmark profile linkage on a real-world large scale dataset.

However, such studies are relevant as they describe comparable pairwise classifier to ours. [13] shows that with similar features, simple classifiers like C4.5,

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

SVM and Naïve Bayes perform well in the artificial, balanced dataset scenario. [12] provides an improved model that combines SVM and username n-gram probability. We use these methods as comparative baselines. We use the WEKA3[2] library for its implementations of C4.5, SVM and Naïve Bayes classifiers. We re-implement Liu *et al.* (2013) approach following their work's description.

In our experiment, we set Twitter as the query dataset and LinkedIn as target dataset. For our canopy threshold, we set $\theta = 200$, as our parameter tuning results. We randomly sampled 1,000 query instances from the ground truth, then retrieve all corresponding target instances by canopying to generate the training set.

Table 5.1. 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
Liu *et al.*	0.910	0.567	0.698	0.767
OPL	0.866	**0.846**	**0.856**	**0.865**

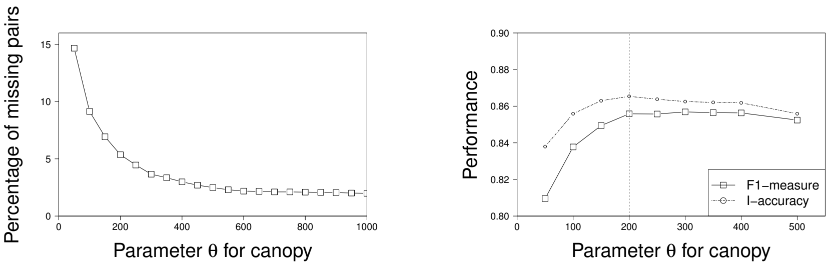
Table 5.1 shows the experimental results. Our approach achieves the best performance, in both F_1 and *I-Acc*. The standard Naïve Bayes classifier outperforms SVM. While not strictly comparable, our Naïve Bayes-based approach also betters Liu's SVM-based method [12]. This validates the same conclusion in [13]. We believe the reason for SVM's subpar performance is caused by missing features in a large proportion of the profiles, which we have described as quite significant an issue for profile linkage.

By reviewing the evaluation results, we observe that simple classifiers perform better in precision but underperform on recall. Although Malhotra *et al.* reports good performance on an artificially-balanced scenario, on real data, naïve classifiers prefer classifying instances as negative, as there is a much larger imbalance of negative instances. Both Liu *et al.* and our approach address this problem by employing prior knowledge about the rarity of tokens, that may carry stronger signals for matching. However, Liu's work adopts this prior in a simple linear way, while OPL embeds it directly within its probabilistic model.

5.3 Canopy Performance and Efficiency

Canopy settings affect both efficiency and linkage performance. Recall is reduced when θ is set too small, preventing correct potential profiles from being in the same canopy. Figure 5.1–(a) illustrates the relationship between θ and missing pair. These missing pairs are indeed matched but at least one profile was pushed out of a matching canopy, as all of its tokens' frequencies are greater than the given θ. The miss percentage reaches a steady state when $\theta \geq 500$, and we feel is already insignificantly different when $\theta \geq 200$.

[2] `http://www.cs.waikato.ac.nz/ml/weka/`

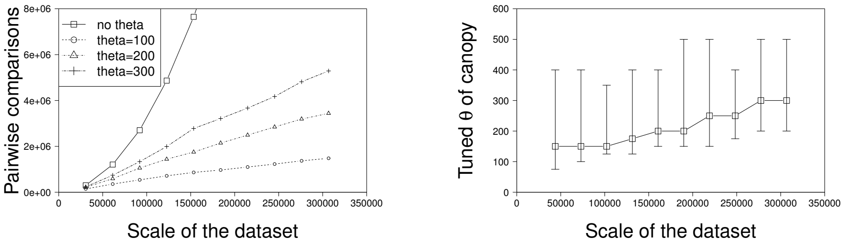


(a)

(b)

Fig. 5.1. Tuning parameter θ on the full dataset: (a) Missing pairs when varying θ, (b) Performance when varying θ.

On the other hand, too large a setting of θ brings in noise that confuses classifiers. A large setting of θ also increases computational overhead (*cf* Section 4.1). Figure 5.1(b) above illustrates the correlation between performance and θ on the full scale of our dataset. Figure 5.1(b) shows that F_1-measure converges after $\theta \geq 200$, so we set $\theta = 200$ for our experiment.



(a)

(b)

Fig. 5.2. Tuning OPL for scalability and efficiency: (a) Comparisons at different dataset scales, (b) Tuned θ for different dataset scales.

While correct parameter setting primarily depends on the requirement of whether precision is more important than recall, θ's value also influences running time. The number of pairwise comparisons used by OPL over different scales of θ is shown in the Figure 5.2(a). We see that OPL using threshold-based canopies is approximately linear in computations to θ. As the computation complexity depends on θ, we also tune the performance against θ, at different dataset scales. Each square in Figure 5.2(b) represents the θ with the best performance, and the respective vertical interval gives the acceptable range of θ values, for which the resultantq loss in F_1 is less than 0.5%. From these results, we can see that

OPL is largely insensitive to dataset scale, a good signal that OPL adequately constrains the linkage task to an approximate linear complexity.

The reason why the optimal θ values show only a neglible increase when the dataset size is scaled up is due to our choice of canopying on username tokens. To avoid conflicts, we find that users prefer to select fairly unique usernames, that may incorporate rarer tokens whose frequencies are less than a most useful choices of θ.

6 Conclusion

We investigate the problem of real world large-scale profile linkage and propose OPL, a probabilistic classifier to address this. OPL caters to specific characteristics of this problem that differentiate it from toy linkage datasets: handling a) the unbalanced nature of the dataset and b) the largeness of the dataset scale. To link the hundreds of thousands of profiles, we employ threshold-based canopies, which directly manipulate and control the resultant linear complexity of the linkage task, allowing an operator a higher degree of flexibility and control over expected run times.

In our experimental results, we show effective performance with 85% F_1-measure and 86% I-accuracy, comparable to previous work. Our cost-sensitive framework also has the ability to prune unnecessary pairwise comparisons while keeping the loss in performance to an acceptable level.

In future work, we plan to improve OPL in two ways: first, to investigate more robust methods for linking OSNs when provided with other heterogeneous data. For example, linking an SNS user to a forum user, by way of the forum content. Second, to leverage the automatically identified set of users to build and test applications where the holistic user profile serves to better aggregate evidence for downstream applications, such as product sentiment estimation.

References

1. Anwar, T., Abulaish, M.: An MCL-Based Text Mining Approach for Namesake Disambiguation on the Web. In: Proceedings of the 2012 IEEE/WIC/ACM International Conference on Web Intelligence (2012)
2. 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 '05. p. 906. ACM Press (2005)
3. 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)
4. Carmagnola, F., Cena, F.: User Identification for Cross-system Personalisation. Inf. Sci. 179(1-2) (2009)
5. 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)

6. Christen, P.: A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. Knowledge and Data Engineering, IEEE Transactions on 24(9) (2012)
7. Cohen, W.W., Richman, J.: Learning to Match and Cluster Large High-Dimensional Data Sets for Data Integration. In: Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining. ACM (2002)
8. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate Record Detection: A Survey. IEEE Trans. on Knowl. and Data Eng. 19(1), 1–16 (2007)
9. Köpcke, H., Rahm, E.: Frameworks for Entity Matching: A Comparison. Data Knowledge Engineering 69(2) (2010)
10. 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)
11. Li, W.: Random Texts Exhibit Zipf's-law-like Word Frequency Distribution. IEEE Transactions on Information Theory pp. 1842–1845 (1992)
12. 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)
13. 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)
14. Narayanan, A., Shmatikov, V.: De-anonymizing Social Networks. In: Proceedings of the 2009 30th IEEE Symposium on Security and Privacy. IEEE (2009)
15. 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)
16. Perito, D., Castelluccia, C., Kaafar, M., Manils, P.: How Unique and Traceable are Usernames? In: Privacy Enhancing Technologies. Springer (2011)
17. Qian, L., Cafarella, M.J., Jagadish, H.V.: Sample-driven schema mapping. In: Proceedings of the 2012 international conference on Management of Data - SIGMOD '12. p. 73. ACM Press (2012)
18. Qian, Y., Hu, Y., Cui, J., Zheng, Q., Nie, Z.: Combining Machine Learning and Human Judgement in Author Disambiguation. In: Proceedings of the 20th ACM international conference on Information and knowledge management. ACM (2011)
19. Vosecky, J., Hong, D., Shen, V.: User Identification Across Multiple Social Networks. In: Networked Digital Technologies. IEEE (2009)
20. 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, NY, USA (2013)
21. 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)