Large-Scale Parallel Matching of Social Network Profiles

Alexander Panchenko[1], Dmitry Babaev[2], and Sergei Obiedkov[3]

[1] TU Darmstadt, FG Language Technology, Darmstadt, Germany
panchenko@lt.informatik.tu-darmstadt.de
[2] Tinkoff Credit Systems Inc., Moscow, Russia
dmitri.lb@gmail.com
[3] National Research University Higher School of Economics, Moscow, Russia
sergei.obj@gmail.com

Abstract. A profile matching algorithm takes as input a user profile of one social network and returns, if existing, the profile of the same person in another social network. Such methods have immediate applications in Internet marketing, search, security, and a number of other domains, which is why this topic saw a recent surge in popularity.
In this paper, we present a *user identity resolution* approach that uses minimal supervision and achieves a precision of 0.98 at a recall of 0.54. Furthermore, the method is computationally efficient and easily parallelizable. We show that the method can be used to match *Facebook*, the most popular social network globally, with *VKontakte*, the most popular social network among Russian-speaking users.

Keywords: User identify resolution, entity resolution, profile matching, record linkage, social networks, social network analysis, Facebook, Vkontakte.

1 Introduction

Online social networks enjoy a tremendous success with general public. They have even become a synonym of the Internet for some users. While there are clear global leaders in terms of the number of users, such as Facebook[4], Twitter[5] and LinkedIn[6], these big platforms are constantly challenged by a plethora of niche and/or local social services trying to find their place on the market. For instance, VKontakte[7] is an online social network, similar to Facebook in many respects, that enjoys a huge popularity among Russian-speaking users.

Current situation leads to the fact that many users are registered in several social networks. People use different services in parallel as they provide complimentary features and user bases. For instance, one common pattern for

[4] http://www.facebook.com
[5] http://www.twitter.com
[6] http://www.linkedin.com
[7] http://www.vk.com

Russian-speaking users is to communicate with Russian-speaking peers with help of Vkontakte and with foreign friends with help of Facebook. Another common pattern is to use LinkedIn for professional and Facebook for private contacts.

Publicly available user information can help in building the next generation of personalised web services, such as search, recommendation systems, targeted marketing, and messaging, to name a few. For instance, Bartunov et al. [1] suggest to use profile matching to perform automatic contact merging on mobile phones. Actually, a similar technology is already integrated in the Android mobile operative system[8]. On the other hand, profile information may be subject to de-anonymization attacks, undesirable for a user [2–4]. No wonder several researchers from information retrieval and security communities recently tried to study methods of user profile correlation across online social networks [5–9, 4, 10].

As information about a single user can be scattered across different networks, integration of data from various platforms can lead to a more complete user representation. Therefore, in many applications it makes sense to build an *integral profile*, featuring information from several sources. In order to do so, it is necessary to perform *user identity resolution*, i.e., to find the same person across various networks. In this paper, we propose a simple, yet efficient method for matching profiles of online social networks.

The contribution of our work is two-fold:

1. We present a new method for matching profiles of social networks. The method has only four meta-parameters. Unlike most existing approaches (see Section 2), the method is easily parallelisable and can be used to process the profiles from an entire social network in a matter of hours. We provide an open-source implementation of the method.[9]
2. We present results of the largest matching experiment to date known to us. While most prior experiments operated on datasets ranging from thousands to hundreds of thousands of profiles, we performed a match of 3 million profiles of Facebook (FB) to 90 million profiles of VKontakte (VK), demonstrating that third parties can perform matching on the scale of entire social network. To the best of our knowledge, we are the first to present a matching of FB to VK.

2 Related Work

2.1 Profile Matching

Bartunov et al. [1] developed a probabilistic model that relies on profile attributes and friendship links. The algorithm was tested on roughly 2 thousand Twitter users and 9 thousand Facebook users. The method achieves F-measure up to 0.89 (precision of 1.0 and recall of 0.8). However, the this is a *local* identity resolution

[8] https://www.android.com

[9] `https://github.com/dmitrib/sn-profile-matching`

method, that requires profiles to be ego-networks of the seed user. From the other hand, in this paper we present a *global* identity resolution method that can potentially match any user of one network with any user of another network.

Veldman [6] conducted a set of extensive experiments with profile matching algorithms. She used profile similarity metrics based on both attributes (name, email and birth date) and friendship relations. The author performed experiments on 2 thousand profiles of LinkedIn and Hyves social networks.

Malhotra et al. [9] used 30 thousand of paired Twitter and LinkedIn profiles to train several supervised models based on attributes, such as name, user id and location. The authors report an F-measure up to 0.98 with precision up to 0.99.

Sironi [5] also used supervised models based on features stemming from similarity of profile attributes. This experiment was done on 34 thousand of Facebook, Twitter and LinkedIn profiles where 2 thousand were paired. Their approach yields precision and recall around 0.90.

Narayanan and Shmatikov [7] proposed an approach that establishes connections between users based on their friendship relations. This incremental method requires a small initial number of matched profiles and access to a graph of friendship links. The authors used the method to match 224 thousand Twitter users with 3.3 million Flickr users and observed an error rate of 12%.

Balduzzi et al. [2] showed that matching can be done effectively based on email addresses.

Jain et al. [10] developed a system that takes as input a Twitter account and finds a corresponding Facebook account. The system relies on profile, content, self-mention and network-based similarity metrics.

Goga et al. [4] present a comprehensive study on profile matching technology. The authors try to correlate accounts of Facebook, Twitter, Google+, Flickr, and MySpace to check a feasibility of a de-anonymization attack. They show that up to 80% of Twitter, Facebook and Google+ profiles from their ground truth can be matched with a nearly zero false positive rate. Their matching method is based on features extracted from user names, locations and pictures unified with help of a binary classifier taking as input two profiles. Two key differences of this method from ours are the following. First, Goga et al. [4] perform no candidate selection. Therefore, in this approach all pairwise comparisons should be done, which is not efficient if one deals with the entire social network. Second, this approach uses no features based on friends similarity, which are core of our approach.

2.2 Name Similarity Matching

Our method heavily relies on name similarity matching. In its simplest form a name can be considered as a string. There is a large body of literature on how to define string similarity [11] and use it to extract similar names from a data set with some works focusing specifically on personal names; see a survey and experimental comparison in [12]. According to this survey, one of the best algorithms for approximate name matching is the algorithm from [13], which

uses a prefix tree to efficiently compute the Levenshtein distance. In [14], three generations of name matching methods are identified, with only third-generation methods showing good results in terms of both precision and recall.

3 Dataset

Two social networks were used in our profile matching experiment. One is the biggest Russian social network VKontakte; the other is Facebook, which is also very popular among Russian-speaking users.

In our experiments, we used publicly available data from VK and FB. The matching algorithm is based on name similarity and the friendship relation: each profile is represented by the first and/or last name of the user and by a list of names of his or her friends in the social network. No other characterising features of profiles were used.

3.1 VKontakte

We collected about 90 million VK profiles that set Russia as their current location. We gathered first and second name of each user along with list of her friends using the "users.get" method of the social network API[10]. Therefore, we can assume that in our experiment VK friend lists are *complete*.

3.2 Facebook

We deal with 3 million public Facebook profiles from Russia. User's name can be obtained via the official API[11], but not list of her friends. That is why friend lists were generated from events displayed in user's feed. Users A and B were considered as friends if a message "A and B are now friends" appeared in feeds of A and B. Profile feeds were collected via the "user/feed" method of the FB API. The problems with this approach is that (1) access to users's feed can be restricted by privacy settings; (2) one need to download all wall posts to gather list of friends, which is not always possible due to API restrictions and requires multiple API calls. Therefore, we should assume that in our experiment FB friend lists are *incomplete*.

3.3 Test data

VKontakte provides a field where a used can specify a link to her FB page. We gathered about 850 thousand known VK-FB profile pairs. However, only 92,488 Facebook users were found in our Facebook dataset out of these 850 thousand profiles. These pairs were used as a ground truth to check correctness of the matching algorithm. A subset of the test data used in our experiments is publicly available[12].

[10] `https://vk.com/dev/users.get`
[11] `https://developers.facebook.com/docs/graph-api`
[12] `https://github.com/dmitrib/sn-profile-matching`

3.4 Name romanisation

Names of Russian FB and VK users can be spelled in both Latin and Cyrillic alphabets i.e. "Alexander Ivanov" or "Александр Иванов". To enable correct name matching, all user names in both networks were converted to Latin script using the Russian-Latin BGN transliteration rules[13].

4 Profile matching algorithm

The algorithm consists of three phases:

1. *Candidate generation.* For each VK profile we retrieve a set of FB profiles with similar first and second names.
2. *Candidate ranking.* The candidates are ranked according to similarity of their friends.
3. *Selection of the best candidate.* The goal of the final step is to select the best match from the list of candidates.

Each profile from VK network is processed independently and hence this operation can be easily parallelised (we rely on MapReduce framework[14]). It is possible to perform matching in both directions (VK→FB and FB→VK). However, all profiles from the target network must be stored at each computational node. Therefore, direction of matching VK → FB minimises the memory footprint of such nodes. Below we describe each step of the method in detail.

4.1 Candidate generation

It is computationally inefficient to calculate similarity of each VK profile with each FB profile. This operation would require about $1.3 \cdot 10^{20}$ pairwise comparisons. This first step reduces the search space retrieving FB users with names similar to the input VK profile. Two names are considered similar if the first letter is the same and the edit distance [14] between names is less than two. This should be true for both first and last names.

We use an index based on Levenshtein Automata [15] to perform fuzzy match between a VK user name and all FB user names. In particular, we relied on the Lucene implementation of this approach[15].

However, the edit distance does not provide a complete solution for name matching, since many first names have several rather different variants, e.g., "Robert" and "Bob", or "Mikhail" and "Misha". One way to address this problem is to use a dictionary of proper names prepared by linguists, e.g., [16], to decide whether two names are synonyms. However, such dictionaries often skip some name variants. For example, the entry for the Russian name "Alexander" in [16]

[13] `http://earth-info.nga.mil/gns/html/romanization.html`
[14] `http://hadoop.apache.org`
[15] `org.apache.lucene.util.automaton.LevenshteinAutomata`

includes "Sanya", but not "Sanek". In addition, they do not include variants based on similarity with names from other languages: e.g., "Alejandro" is not in the entry for "Aleksandr" in [16].

Therefore, we decided to build our own dictionary using pairs of profiles known to belong to the same person. We do this by taking the transitive closure of the symmetric binary relation over names given by these pairs. Every two names from the same equivalence class are considered to be synonyms. The fundamental deficiency of this approach is that, being a variant is not an equivalence relation, since transitivity does not always hold. For example, two different Russian names, "Alexander" and "Alexey", are often abbreviated as "Alex". With our approach, this results in declaring "Alexander" and "Alexey" variants of each other, which they are not. Nevertheless, we let this happen and use shared friends to disambiguate between persons erroneously declared to have similar names.

Another problem is that some people use totally unrelated first names (such as "Andrey" and "Vladimir" or even "Max" and "Irina") in different networks. We solve this problem by removing "strange" pairs based on the number of times such a pair occurs in the list (unique or infrequent pairs can safely be removed). The final list of synonym clusters was quickly checked manually.

While candidate generation step greatly reduces search space, a person that indicated different name or a pseudonym in two social networks will not be recognised with our approach. From the other hand, in this situation a person is probably prefers to hide his or her identity and therefore it is more appropriate to perform no matching for this user at all.

4.2 Candidate ranking

The higher the number of friends with similar names in VK and FB profiles, the larger the similarity of these profiles. Two friends are considered to be similar if:

- First two letters of their last names match, and
- The similarity between their first names and the similarity between their last names are both greater than thresholds α and β, correspondingly. We empirically set α to 0.6 and β to 0.8. String similarity sim_s is calculated as follows:

$$sim_s(s_i, s_j) = 1 - \frac{lev(s_i, s_j)}{\max(|s_i|, |s_j|)},$$

  where lev is edit distance of string s_i and s_j. At this step we use the standard algorithm for calculation of Levenstein distance[16], not Levenstein Automata.

Matching friends with rare names should be weighted higher than a match of friend with matching friends with common names. Indeed, two unrelated profiles can easily have several friends with similar common names.

Probability of a user with first name s^f and second name s^s, provided than these events are independent is $P(s^f, s^s) = P(s^f)P(s^s) = \frac{|s^f|}{N}\frac{|s^s|}{N}$. Here $|s^f|$ and

[16] org.apache.lucene.search.spell.LevensteinDistance

$|s^s|$ are frequencies of respectively first and second names and N is the total number of profiles. Thus, expectation of name frequency equals to $\frac{|s^f| \cdot |s^s|}{N}$. In our approach, contribution of each friend to similarity sim_p of two profiles p_{vk} and p_{fb} is inverse of name expectation frequency, but not greater than one:

$$sim_p(p_{vk}, p_{fb}) = \sum_{j: sim_s(s_i^f, s_j^f) > \alpha \wedge sim_s(s_i^s, s_j^s) > \beta} \min(1, \frac{N}{|s_j^f| \cdot |s_j^s|}).$$

Here s_i^f and s_i^s are first and second names of a VK profile, correspondingly, while s_j^f and s_j^s refer to a FB profile.

4.3 Best candidate selection

FB candidates are ranked according to their similarity sim_p to an input profile p_{vk}. There are two thresholds the best candidate p_{fb} should pass to match:

- its score should be higher than the *similarity threshold* γ:
$$sim_p(p_{vk}, p_{fb}) > \gamma.$$
- it should be either the only candidate or score ratio between it and the next best candidate p'_{fb} should be higher than the *ratio threshold* δ:
$$\frac{sim_p(p_{vk}, p_{fb})}{sim_p(p_{vk}, p'_{fb})} > \delta.$$

The δ threshold enforces the fact that a VK user has only one account in FB. On the other hand, one FB profile can be linked with several VK profiles. Still, in this case only the match with the highest score is kept.

5 Results and Discussion

We performed matching of VKontakte and Facebook profiles (c.f. Table 1) with the approach described above. Results of the candidate ranking step were saved. At this point, we conducted a series of experiments varying the similarity threshold γ and the ratio threshold δ. Results of these experiments in terms of precision and recall with respect to the test collection (see Section 3.3 are presented in Figure 1. The bold line denotes the best precision at certain level of recall.

Table 1. Statistics of VKontakte and Facebook.

	VKontakte	**Facebook**
Number of users in our dataset	89,561,085	2,903,144
Number of Russian-speaking users[17]	100,000,000	13,000,000
User overlap	29%	88%

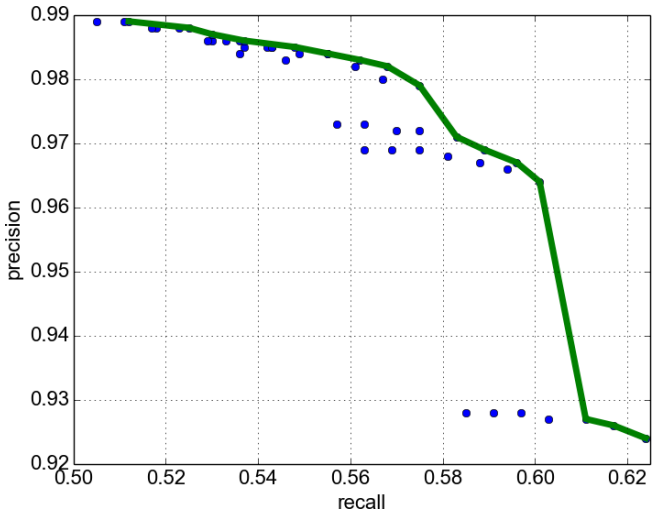

Fig. 1. Precision-recall plot of our matching method. Here we perform a grid search of two method parameters: profile *similarity threshold* $\gamma \in [1; 4]$ and profile similarity *ratio threshold* $\delta \in [3; 6]$. The bold line denotes the best precision at a given recall.

As one may observe, our method yields very good results achieving precision of 0.97 at recall of 0.58. Furthermore, a configuration of the approach yielding 99% precision recalls roughly 50% of relevant profiles.

In order to perform the final matching of VK and FB we chose a version of the algorithm that provides precision of 0.98 and recall of 0.54 (see Table 2). Results were obtained in 4 hours on a Hadoop cluster with 100 nodes of type `m2.xlarge` (2 vCPU, 17 GB RAM) on the AWS EC2 cloud[18]. The mentioned above configuration of the method mentioned above retrieved 644,334 VK profiles of FB users. Thus, we found corresponding VK pages of 22% Facebook users present in our collection.

While our approach makes only few errors, reaching precision of 0.98, it is not able to match a significant fraction of 40-50% of user profiles. The key factors hampering correct retrieval are the following:

- In our method, we perform fuzzy search with name synonyms that can lead to semantic drift. For instance, "Maria" is expanded with its alias "Masha". According to fuzzy search "Masha" and "Misha" are related. But the latter is a shortcut for "Michael" in Russian.
- Implementation of the Levenstein Automata used in our experiments retrieves candidates with distance lower or equal than two. Thus, people with long names and surnames can be missed during candidate generation.

[17] `http://www.comscore.com/Insights/Data-Mine/Which-Sites-Capture-The-Most-Screen-Time-in-Russia` and `http://vk.com/about` provide statistics on number of Russian-speaking users.

[18] `http://aws.amazon.com`

Table 2. Matching of user profiles of Facebook and VKontakte social networks. The upper table presents four main parameters of our profile matching method. The lower part of the table presents results of the final matching of the two networks.

Parameter	**Value**
First name similarity threshold, α	0.8
Second name similarity threshold, β	0.6
Profile similarity threshold, γ	3
Profile ratio threshold, δ	5
Number of matched profiles	644,334 (22% of 2,903,144 FB users)
Expected precision	0.98
Expected recall	0.54

- People often intentionally indicate different names in two social networks or use different aliases. Our approach is not designed to identify and match such profiles.
- First letter mismatch. Different variants of the same name/surname in Russian can start from different letters. Furthermore, transliteration can lead to such mismatches as well, e.g. surname "Ефимов" can be spelled in Latin as "Efimov" or "Yefimov".
- People often use transliterated versions of their names in one network, but stick to the original Cyrillic versions in the other. The method always works with transliterated names, but our transliteration can be quite different from the one done by a user.
- Due to nature of the friend collection method used, some FB friends can be absent in our dataset.

In order to improve performance of the method, one would need to tackle the problems mentioned above.

6 Conclusion

In this paper, we presented a new user identity matching method. Unlike most previous approaches, our method is able to work on the scale of real online social networks, such as Facebook, matching tens of millions of users in several hours on a medium-sized computational cluster. The method yields excellent precision (up to 98%). At the same time it is able to recall up to 54% of correct matches.

The method was used to perform the most largest-scale matching experiment up to date. We matched 90 millions of VKontakte users with 3 million of Facebook users.

A prominent direction for the future work, is to use supervised learning in order to improve *candidate ranking*. One way pionered by [4] is to use a binary classifier predicting if two profiles match; profile similarity in this case would be the confidence of positive class. Learning to rank methods [17] is another way to

cast profile matching as a supervised problem. The supervised models provide a convenient framework where name similarity features, used in our method, can be mixed with attribute-, network-, and image-based features.

Acknowledgements

This research was conducted as part of a project funded by Digital Society Laboratory LLC. We thank Prof. Chris Biemann and three anonymous reviewers for their thorough comments that significantly improved quality of this paper.

References

1. Bartunov, S., Korshunov, A., Park, S.T., Ryu, W., Lee, H.: Joint link-attribute user identity resolution in online social networks. In: Proc. of the Sixth SNA-KDD Workshop at KDD. (2012)
2. Balduzzi, M., Platzer, C., Holz, T., Kirda, E., Balzarotti, D., Kruegel, C.: Abusing social networks for automated user profiling. In: Recent Advances in Intrusion Detection, Springer (2010) 422–441
3. Wondracek, G., Holz, T., Kirda, E., Kruegel, C.: A practical attack to de-anonymize social network users. In: 2010 IEEE Symposium on Security and Privacy (SP), IEEE (2010) 223–238
4. Goga, O., Perito, D., Lei, H., Teixeira, R., Sommer, R.: Large-scale correlation of accounts across social networks. Technical report, International Computer Science Institute (2013)
5. Sironi, G.: Automatic alignment of user identities in heterogeneous social networks. Master's thesis, Politechnico di Milano, Italy (2012)
6. Veldman, I.: Matching profiles from social network sites: Similarity calculations with social network support. Master's thesis, University of Twente, Italy (2009)
7. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: Security and Privacy, 2009 30th IEEE Symposium on, IEEE (2009) 173–187
8. Raad, E., Chbeir, R., Dipanda, A.: User profile matching in social networks. In: 13th International Conference on Network-Based Information Systems (NBiS), IEEE (2010) 297–304
9. Malhotra, A., Totti, L., Meira Jr, W., Kumaraguru, P., Almeida, V.: Studying user footprints in different online social networks. In: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), IEEE Computer Society (2012) 1065–1070
10. Jain, P., Kumaraguru, P., Joshi, A.: @I seek 'fb.me': Identifying users across multiple online social networks. In: Proceedings of the 22nd international conference on World Wide Web companion, International World Wide Web Conferences Steering Committee (2013) 1259–1268
11. Boytsov, L.: Indexing methods for approximate dictionary searching: Comparative analysis. Journal of Experimental Algorithmics (JEA) **16** (2011) 1–1
12. Du, M.: Approximate name matching. NADA, Numerisk Analys och Datalogi, KTH, Kungliga Tekniska Högskolan. Stockholm: un (2005)
13. Navarro, G., Baeza-Yates, R., Marcelo Azevedo Arcoverde, J.: Matchsimile: a flexible approximate matching tool for searching proper names. Journal of the American society for Information Science and Technology **54**(1) (2003) 3–15

14. Lisbach, B., Meyer, V.: Linguistic Identity Matching. Springer Verlag. Heidelberg. (2013)
15. Schulz, K., Mihov, S.: Fast string correction with Levenshtein-automata. International Journal of Document Analysis and Recognition **5** (2002) 67–85
16. Petrovsky, N.: Dictionary of Russian personal names. `http://www.gramota.ru/slovari/info/petr`. M.: In Russian Dictionaries (2000)
17. Trotman, A.: Learning to rank. Information Retrieval **8**(3) (2005) 359–381