Entity Matching in Online Social Networks

Olga Peled[1], Michael Fire[1,2], Lior Rokach[1] and Yuval Elovici[1,2]
[1]Department of Information Systems Engineering, Ben Gurion University, Be'er Sheva, 84105, Israel
[2]Deutsche Telekom Laboratories at Ben-Gurion University of the Negev,
Email: olgit23@gmail.com, {mickyfi,liorrk,elovici}@bgu.ac.il

Abstract—In recent years, Online Social Networks (OSNs) have essentially become an integral part of our daily lives. There are hundreds of OSNs, each with its own focus and offers for particular services and functionalities. To take advantage of the full range of services and functionalities that OSNs offer, users often create several accounts on various OSNs using the same or different personal information. Retrieving all available data about an individual from several OSNs and merging it into one profile can be useful for many purposes. In this paper, we present a method for solving the Entity Resolution (ER), problem for matching user profiles across multiple OSNs. Our algorithm is able to match two user profiles from two different OSNs based on machine learning techniques, which uses features extracted from each one of the user profiles. Using supervised learning techniques and extracted features, we constructed different classifiers, which were then trained and used to rank the probability that two user profiles from two different OSNs belong to the same individual. These classifiers utilized 27 features of mainly three types: name based features (i.e., the Soundex value of two names), general user info based features (i.e., the cosine similarity between two user profiles), and social network topological based features (i.e., the number of mutual friends between two users' friends list). This experimental study uses real-life data collected from two popular OSNs, Facebook and Xing. The proposed algorithm was evaluated and its classification performance measured by AUC was 0.982 in identifying user profiles across two OSNs.

Keywords—Online Social Networks; Entity Resolution; Machine Learning

I. INTRODUCTION

Social networking has now become one of the most popular and utilized activities on the web. Social network users have access to a "second life" on OSNs, such as Facebook, Twitter, and LinkedIn. People use OSNs for various purposes like meeting new friends, discussing opinions, playing online games, and sharing information. There are hundreds of online social networking sites, each network with its own focus and its own particular services and functionalities offered to its users. To make use of the provided services and functionalities and to keep updated with other members, users tend to create several accounts on various sites. This phenomenon has been studied in the past. For example, in November 2007, Patriquin reported on the member overlap between various OSN services [1]. He showed that 64% of Facebook users also have MySpace accounts and 42% of LinkedIn users also have Facebook accounts. In most cases, to join different OSNs users must register separately to each one by filling out personal information forms. In other cases, people can register to new OSNs using their account credentials from different networks. Users who create several accounts on various sites (i.e., for personal use, for work, etc.) can use the same or different personal information. In many cases, although people provide their personal information to multiple OSNs, they do not intend to provide data for integration with other information sources. Therefore, the integration of personal information can be quite a sensitive issue for privacy reasons.

The process of identifying different profiles which belong to the same real individual is known as Entity Resolution (ER). Today there are various sites, such as Snitch Name and Pipl, which allow the searching of users across multiple OSN sites by name**.** However, cross-referencing users between different sites is not an easy task, since, in many cases, users of different OSNs fill in different details, use different user names, and have different lists of friends. To complicate the problem even further, many different users have similar names and personal details. For example, according to a frequency table of names in Facebook[1], 17,204 out of 100 million people are named John Smith. As a result, even if two user profiles have the same first and last name, it is not enough to confirm that these two user profiles belong to the same real person.

In this study, we present an algorithm for solving the ER problem for profiles from different OSNs. We attempt to find out which user profiles of the first social network correspond to user profiles of the second social network, where we assume a person to have at most one profile per OSN site. Our algorithm matches two user profiles from two different OSNs based on machine learning techniques that use feature extraction on a user's information and on the user's friends list. Different classifiers are trained and used to rank the probability that two user profiles from different OSNs belong to the same individual. These classifiers utilize multiple features of mainly three types: (1) name based features, (2) general user info based features, and (3) social network topological based features, such as the number of mutual friends between the two users' friends list. In contrast to previous studies, our algorithm uses machine learning techniques with feature extractions of many users' data features, like name, birthday, location, professional experience, education, etc. Additionally, previous studies evaluated their methods on small datasets containing about 3,000 users [2], while in this study we evaluated our ER algorithm on large datasets, which included more than 30,000 users obtained from two different popular OSNs, Facebook and

[1] http://www.skullsecurity.org/blog/2010/return-of-the-facebook-snatchers

Xing. The experimental study using real-life data collected from two OSNs, Facebook and Xing demonstrated a high AUC of 0.982 in identifying user profiles across two networks. The main advantage of our matching algorithm is that we used machine learning techniques and suggested many new features.

II. Related work

Our study deals with Entity Resolution in OSNs, thus, in this section we will describe the main approaches to match user profiles across multiple social networks that belong to the same real individual.

Vosecky et al. [2] presented a matching technique in which each user profile is represented as a vector consisting of the values of individual profile fields (e.g., name, date of birth, etc.). The comparison between any two vectors consists of two phases. In the first phase, the algorithm calculates a similarity score between corresponding vector fields using an appropriate string matching function for each field, resulting in a similarity vector. In the second phase, a weighting vector is applied to the similarity vector to calculate the overall similarity. This method was trained to determine an optimal set of profile fields and tested on less than 3,000 users from Facebook and StudiVZ (a German language social network), with 50 mutual users (the database was manually checked). The result had an 83% accuracy rate in identifying duplicated users across two social networks.

Veldman [3] suggested two models to solve the ER problem across two social networks. In the simple model, Veldman compared all of the profiles of the first network against all of the profiles of the second network. Each compared pair received a pairwise similarity score. The higher this score, the higher the probability that these profiles belong to the same real person. The pairs that satisfied the so-called pairwise threshold were the candidate matches and from these candidate matches, the final matches were chosen. In addition to the simple model, Veldman also used the network model, in which for each candidate match the network similarity score is calculated. This is done by determining the overlap in the networks of both profiles of the candidate match. The more the networks overlap, the higher the network similarity score and the higher the probability that two user profiles belong to the same real person. Again, the candidate matches should satisfy a network threshold to remain a candidate match. Then, from the remaining candidate matches, the final matches can be chosen.

In 2011, Carmagnola et al. [4] presented the preliminary results of their research work aimed at uniquely identifying users on different social systems and retrieving their data distributed over the profiles stored in such systems. Their proposed algorithm requires some initial attributes about the user to set up the search. Given a set of input attributes about said user, crawlers search for user profiles that match the input request. Once the user profile has been retrieved, with a matching nickname or full name, the other initial attributes are used to compute a score for this match. Each attribute has its own weight based on how strong a positive match indicator it is. Lastly, a probability is calculated which represents the probability that the newly discovered user attributes actually belong to the searched user. Even though it offers promising results, this approach needs to be further advanced to become really usable.

Iofciu et al. [5] went beyond user matching based on profile information and included content-based information (i.e., tags users assigned to images and bookmarks) when matching Flickr, Delicious, and StumbleUpon user accounts. They suggested combining profile attributes (e.g., usernames) with an analysis of the tags contributed by them to identify users. They also suggested various strategies to compare the tag-based profiles of two users and were able to achieve an accuracy of almost 80% in user identification.

Narayanan and Shmatikov illustrated in two recent papers that information from different data sources can be combined to identify a user [6, 7]. They further showed that statistical methods can be applied to identify micro-data by cross-correlating multiple datasets [6]. They extended their approach to social networks [7] and proved that it is possible to identify members by mapping known auxiliary information on the social network topology. They also demonstrated that a third of users with accounts on both Twitter and Flicker can be re-identified in an anonymous Twitter graph with only a 12% error rate.

In our study, just as in previous work, we consider a range of profile fields and investigate their importance for the matching process. However, unlike previous studies, we used supervised machine learning techniques and constructed classifiers that utilized a vast variety of features. Moreover, we tested our methods on a real and large dataset that contained over 30,000 entities.

III. Proposed method

We propose a method composed of four main components and each component is detailed in the following subsections. Recall that entity resolution in social networks is a task that is represented as a binary classification task. Thus, this process should begin with the generation of a training set. The training set is used as an input to the induction algorithm, which induces a classifier. The classifier estimates the probability that two user profiles from different OSNs belong to the same real individual.

A. Acquiring the Data

To match user profiles from different OSN sites, a large and suitable dataset from social networks is required. However, it is very difficult to generate such a dataset and the only way to get a realistic dataset is to use real data. Suppose S_1 and S_2 are OSN sites where people have profile pages on which they share personal information and lists of friends. The data on these profile pages is retrieved using a crawler. To get access to the public profiles on each of the social networks we had to use a real account for each network. The data on social networking sites can be very diverse, unstructured, and even unsuitable, thus it will need preprocessing. For example, based on the knowledge of the structure of the user profile page and the user's friends' page in a particular network, content can be extracted. Then, irrelevant data can be filtered out. Relevant and non-relevant data depends on the purpose. After preprocessing, the datasets are ready for the matching process.

B. Feature Extraction

After preprocessing, the datasets are ready for the feature extraction process. We extracted 27 features with which we compared two profiles from different OSN sites. There are three main types of features: name based features (i.e., the Soundex value or Edit distance of two names), general user info based features (i.e., the cosine similarity between two user profiles), and social network topological based features (i.e., the number of mutual friends between the two users' friends list). The different features are described in the next section.

C. Constructing the Training Datasets

First, in order to identify which users are members of both networks; we performed a cross reference between the collected user's profiles according to their names. Next, using the profile photos, we manually checked each matched pair of profiles to see whether they belong to the same real person. Each checked pair was labeled as a match or a non-match.
The remaining users who did not have a match by name were used to create negative pairs. Using the labeled pairs, we constructed the training set.

D. Building the Model

This is the last step in our proposed method. In this step the goal is to build a model which can calculate two user profiles from different OSNs and the probability that they belong to the same real individual. To build such a model we used machine learning techniques. In the real world there is a large ratio between people who have several accounts on different social networking sites to people who do not. To simulate this ratio, we used a ratio of one to five; meaning for every positive pair there were 5 negative pairs. One of the main difficulties in building a model is choosing the right algorithms to apply to the training set. In our case, the data was overwhelmingly imbalanced and we therefore, needed to use algorithms that knew how to deal with imbalanced data.

IV. FEATURES

Features are used as indicators which signify whether a candidate pair of users belongs to the same real individual. There are three main types of features: name based features, general user info based features, and social network topological based features. Given two user profiles, we tried to calculate the similarity between the different data fields of each user by calculating the following features.

A. Name Based Features

Name based features are features that represent the similarity between two names. Below we specify the different features that have been used:

Soundex Name Similarity: SOUNDEX codes of the two user full names were compared to see how similar the full names sound when spoken. If the two Soundex codes are equal, then the similarity score is 1 and if the Soundex codes are not equal it is 0. For example, SOUNDEX ("Smith") = SOUNDEX ("Smythe") = S530, therefore the similarity name score is 1.

Difference Name Similarity: The DIFFERENCE function performs a SOUNDEX on two strings and returns an integer that represents how similar the SOUNDEX codes are for those strings. The integer returned is the number of characters in the SOUNDEX values that are the same. For example, SOUNDEX ("Olga") = O420 and SOUNDEX ("Olgit") = O423; notice that there are 3 of the same characters in the SOUNDEX values, therefore the DIFFERENCE ("Olga", "Olgit") = 3/4= 0.75.

LCS Name Similarity: The longest common sub-string (LCS) repeatedly finds and removes the longest common sub-string in the two compared strings up to a minimum length. A similarity measure can be calculated by dividing the total length of the common sub-strings by the average lengths of the two original strings. We use the LCS algorithm to calculate the similarity between the two users' full names. For example, the LCS similarity of "Gail West" and "Vest Abigail" is 7/10.5=0.666.

Compression Name Similarity: The basic idea of compression based similarity is to use the length of the compressed strings to calculate a similarity measure. We used this technique to calculate the similarity score between the two users' full names.

Damerau Levenshtein Name Similarity: Levenshtein or edit distance is defined to be the smallest number of edit operations, inserts, deletes, and substitutions required to change one string into another. The Damerau-Levenshtein distance is a variation of edit distance where a transposition of two characters is also considered to be an elementary edit operation. We used this metric to calculate the similarity between the two users' full names.

Jaro Winkler Name Similarity: Jaro is an algorithm commonly used for name matching in data linkage systems. A similarity measure is calculated using the number of common characters (i.e., same characters that are within half the length of the longer string) and the number of transpositions. Winkler (or Jaro-Winkler) improves upon the Jaro algorithm by applying ideas based on empirical studies which found that fewer errors typically occur at the beginning of names. The algorithm increases the Jaro similarity measure for up to four agreeing initial characters. We used this metric to calculate the similarity between the two users' full names.

N-Gram Name Similarity: N-grams are sub-strings of length n. An n-gram similarity between two strings is calculated by counting the number of n-grams in common (i.e., n-grams contained in both strings) and dividing by either the number of n-grams in the shorter string (called Overlap coefficient), or the number of n-grams in the longer string (called Jaccard similarity), or the average number of n-grams in both strings. We used 2-grams and 3-grams to calculate the similarity between the two users' full names.

VMN Name Similarity: The Vosecky et al. [2] string comparison function, VMN, is used for matching two peoples' names. It is designed for full and partial matches of names consisting of one or more words. VMN supports the case of swapped names and the cases of partial matches. We used VMN to calculate the similarity between the two users' full names.

Names Frequency Similarity: We used the frequency table of names in Facebook to calculate the frequency of a particular name. The similarity was calculated by the average between the frequencies of the user name in each network. We decided not to normalize this score and to let the machine learning model distinguish between high and low frequencies.

B. General User Information Based Features

General user information based features are features that represent the similarity between the different parts of personal information of two users.

Locations Distance: In many social networks, such as Facebook, Twitter, and Google+, the location of the user is known. In cases where the locations are known, this feature represents the distance between the two users' locations. We calculated this distance using Bing maps. Since the location field is a complex characteristic, we cannot simply compare the names of the two locations. Therefore, we decided to calculate the distance, not normalize it, and let the machine learning model distinguish between short and long distances.

N-Gram Current Employer Similarity: We also used the 3-grams method to calculate the similarity between the users' current employers.

Damerau Levenshtein Current Employer Similarity: We also used this metric to calculate the similarity between the two users' current employers.

Jaro Winkler Current Employer Similarity: We also used this metric to calculate the similarity between the users' current employers.

Jaccard Similarity: We used the Jaccard measure similarity to calculate the similarity between two full user profiles without user names, between two user professional experiences and between two user educational backgrounds if the information was provided. We also used Jaccard to calculate the similarity between two users' information page fields. This was accomplished in two steps. In the first step, for each profile in each network, we created text representation of the profile by aggregating all the user's information fields, which appeared in the user's profile page without the name, education and background fields, to a single text. Afterwards, in the second step, we compared the two texts using Jaccard measure.

Semi Vector Space Model Similarity (semi VSM): The Vector Space Model (VSM) [8] is a way of representing documents through the words that they contain and is a standard technique in Information Retrieval. The VSM allows decisions to be made regarding which documents are similar to each other and to a given query. The vector space model procedure consists of three stages. The first stage is document indexing where content bearing terms are extracted from the document text. The second stage is weighing of the indexed terms to enhance the retrieval of documents relevant to the user. The last stage ranks the document with respect to the query and according to a similarity measure. To use the Vector Space Model we needed sets of documents and a query. In our case there are only two documents and since our goal is to calculate the similarity between them, we used one user document as a query and the other user document as only one document retrieved. As we had only one retrieved document, we changed the implementation of each term weight by only calculating the TF (Term Frequency) value rather than multiplying the IDF (Inverse Document Frequency) value. We used the semi VSM similarity to calculate the similarity between two full user profiles without user names, between two user professional experiences and between two user educational backgrounds, if information was provided. Using semi VSM we also calculated the similarity between two users' information page fields. This was done in two steps. In the first step, for each profile in each network, we created text representation of the profile by aggregating all the user's information fields, which appeared in the user's profile page without the name, education and background fields, to a single text. Afterwards, in the second step, we compared the two texts using semi VSM similarity. Since we did not use the IDF value, we refer to this method as semi VSM.

Vector Space Model of Full Profiles Similarity: As described above, at first we did not use the IDF value. To use the SVM method as is, we decided that instead of calculating the similarity between two user documents only, we would calculate the similarity between one user document from the first social network and all the users' documents of the second network. To clarify the process, suppose u_1 and u_2 are two users' documents where u_1 belongs to social network S_1 and u_2 belongs to social network S_2.

Instead of calculating the similarity between u_1 and u_2 only, we calculated the similarity between u_1 and all the users in S_2 and returned a score for u_2; this is the similarity. The same happens for u_2 against all the users in S_1.

C. Social Network Topological Based Features

Social network topological based features are features that represent the similarity between the networks of two user profiles.

Mutual Friends: This feature represents the number of mutual friends of two users. Mutual friends are counted by the number of friends with identical names in both cycles of friends.

Mutual Friends of Friends: This feature represents the number of mutual friends of friends of two users. Mutual friends are counted by the number of friends with identical names in both cycles of friends.

V. EXPERIMENTAL STUDY

A. Setup

To retrieve real data we chose to acquire data from two big and popular social networks, Facebook and Xing, by using dedicated web crawlers. Facebook is a popular free social networking website launched in February 2004. As of October 2012, nearly one billion users have active Facebook accounts. Facebook allows registered users to create profiles, upload photos and video, send messages, and keep in touch with friends, family, and colleagues. Xing was founded in 2003 in Hamburg, Germany and is a European social network for business professionals with more than 12 million users as of September 2012. Xing offers personal profiles, contacts, groups, discussion forums, event coordination, and other common social community features**.** Xing is a platform where professionals from all kinds of different industries can meet up, find jobs, colleagues, new assignments, cooperation partners, experts, and generate business ideas. This site is very popular in Switzerland, Germany, and Austria. Additionally, Xing competes with the American platform LinkedIn for social networking among businesses. We developed a dedicated crawler component to crawl through Facebook and Xing and collect information on user profiles and contact lists accessible to the public. The crawler starts with a user's profile page, downloads it, downloads his friends list, and then continues to crawl each friend, friend of friend, and so on.

To choose the starting profiles of each crawler, we manually found pairs of public profiles, one from each network, that belong to the same real entity. Using these starting profiles ensured that we would have some mutual users across these two social networks. To use the raw data we crawled, the data needed preprocessing. In our case, based on the knowledge of the structure of the social network, content was extracted and irrelevant data was filtered out. We developed a tool that extracted interesting personal data about each user of each network, such as name, gender, professional experience, education, list of friend's names, etc. All the data that was extracted from each network is described in Table I.

We collected 16,561 user profiles from Facebook and 15,430 from Xing. Among the collected users, there were 464 pairs of users, one from Xing and one from Facebook, where the Xing user's full name contains the Facebook user's full name (we will refer to these users as users with very similar names). Each of the 464 pairs was manually checked to uncover whether they represented two user profiles that belonged to the same real entity. Lastly, we received 158 pairs of users with very similar names who belonged to the same real entity and 306 pairs of users with very similar names who did not belong to the same real individual.

TABLE I. EXTRACTED DATA FROM FACEBOOK AND XING

Facebook
UserID
Username
Name
Gender
Birthday
Hometown
Current city
Languages
Friends
Religion views
Political views
Favorite quotes
Relationship status
Websites
Networks
Relatives
Emails
Education (school/collage name, class, concentration, degree, type)
Professional Experience (employer, position, start date, end date, description)

Xing
Username
Name
Hometown
Employer status
Academic grade
Languages
Friends
Groups
Haves
Interests
Organization
Wants
Qualifications
Education(collage name, fields of study, degree, specialized subjects, start date, end date)
Professional Experience (job title, company, industry, start date, end date, company home page, position description)

To build the training set, we randomly divided all the users we crawled from both networks into ten groups and associated each labeled pair with the appropriate group (i.e., the group that contains both of the users of the labeled pair). Note that if the users of a labeled pair belonged to a different group, the labeled pair was thrown out. Next, for each positive pair in each group, we randomly generated ten negative pairs, namely, where $<u_1, u_2>$ is a matched pair, where u_1 belongs to social network S_1 and u_2 belongs to social network S_2. Our main assumption is that a real person has only one user profile page per OSN site. Therefore, there is no user in S_1 who can also be matched to u_2 and there is no user in S_2 who can also be matched to u_1. Using this assumption, for $u_{1,}$ we randomly picked five user profiles from S_2 that are different from u_2; and for $u_{2,}$ we randomly picked five user profiles from S_1 that are different from u_1. These pairs obviously do not match and are therefore, labeled as negative pairs. Lastly, in order to build ten training sets and ten test sets, we took out nine different groups from the ten groups and used them as a training set and the group that was left as a test set.

To compare the various classifiers, the accuracy, true positive rate and false positive rate, AUC measures were used.

We examined Weka's implementation [9] of various supervised learning methods for combining various features, from which we selected the top 6 methods. The following methods have been found to be the most accurate: AdaBoost, Rotation Forest, Random Forest, Logistics Model Tree (LMT), LogitBoost and Artificial Neural Networks. Each classifier outputted the probability that two user profiles belong to the same real individual. For each of these algorithms most of the configuration parameters were set to their default values except for the following: For LMT, the minimum number of instances was set to 30; for LogitBoost the number of iterations was set to 25; for Random Forest the random classification trees number was set to 150 and for AdaBoost the number of iterations was set to 200.

B. Results

Fig. 1 presents the area under the ROC curve that was measured for various learning methods. As we can see, all of the algorithms' AUCs are above 0.972. LogitBoost demonstrates the best performance and significantly outperforms the following methods: Multilayer Perceptron, LMT, and Random Forest. The null-hypothesis that all methods perform the same and the observed differences among their AUC values are merely random, was rejected using the ANOVA test, with F (5,594) =17.495, p<0.01. Even though there are small differences between the six methods' AUC, the differences are still statistically significant. Additional statistical tests have showed that LogitBoost also outperforms other methods in terms of accuracy and false positive. However, LogitBoost is not as good as Random Forest in True Positive rate.

As the LogitBoost algorithm seems to be the most appropriate algorithm to solve our particular problem, we decided to use it to evaluate the contribution of each used feature defined above. In order to evaluate the merit of each feature, we performed two procedures for each feature: "all-but-x" and "only-x". The first procedure "all-but-x" aims to measure how much a certain feature contributes to the entire model. For this purpose we built a LogitBoost model with all features except for one and measured the decline in AUC. We used AUC as single measure for comparing the performance of various features. The second procedure, "only-x", aims to evaluate how each feature performs on its own. Table II presents the performance of all the features.

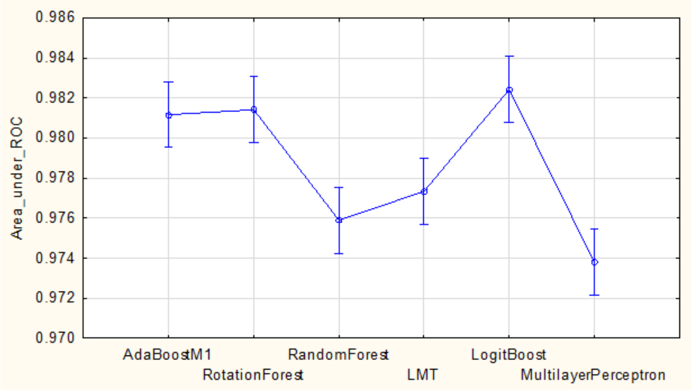

Fig. 1. Area under the ROC results

The best single feature is "Jaro Winkler Name Similarity" with AUC of 0.9504 (based on the "only-x" procedure). Note that the top five features are all name based features. This is indicating that Name based features have a strong direct impact on the classification. Still according to the "all-but-x" procedure, the top two features with the largest drop in AUC belong to two different categories: name based and the general user information based features category. This is indicating that all users' profile fields should indeed to be taken into account when possible and not just users' names.

VI. CONCLUSION AND FUTURE WORK

We presented a supervised learning method to match user profiles across multiple OSNs. Our method is based on machine learning techniques that use a variety of features extracted from a user's profile as well as their friends' profile. Different classifiers were trained and used to rank the probability that two user profiles from different OSNs belong to the same entity. The classifiers utilized multiply features of mainly three types: (1) name based features, such as the Soundex value of two names, (2) general user info based features, such as the cosine similarity between two full user profiles, and (3) social network topological based features, such as the number of mutual friends between two users' friends list. The proposed method was evaluated using real-life data collected from two OSNs, Facebook and Xing and obtained high matching performance. This high performance can be attributed to two main factors: The usage of machine learning techniques and the usage of a large variety of features. This high result is evidence that user identification based on web profiles is conceptually and practically possible. Although we have seen that the name based features are the most important ones, the combination of all of the 27 features achieve best results. Moreover the name based features have a strong direct impact on the classification.

TABLE II. AUC PERFORMANCE FOR "ONLY-X" AND "ALL-BUT-X"

Type	**Feature**	**All-but-X**	**Only-X**
Name Based Features	soundexNameSimilarity	0.9824	0.9284
	differenceNameSimilarity	0.9824	0.9284
	compressionNameSimilarity	0.9831	0.9463
	damerauLevenshteinNameSimilarity	0.9827	**0.9503**
	jaroWinklerNameSimilarity	0.9826	**0.9504**
	twoGramNameSimilarity	0.9827	0.9465
	threeGramNameSimilarity	0.9829	**0.9502**
	lcsNameSimilarity	0.9824	**0.9503**
	vmnNameSimilarity	0.9824	**0.9502**
	namesFrequencySimilarity	**0.976**	0.5976
General User Information Based Features	hometownDistance	0.9813	0.5494
	currentCityDistanceSimilarity	0.9831	0.6839
	threeGramCurrentEmployerSimilarity	0.9819	0.607
	dameraulLevenshteinCurrentEmployerSimilarity	0.9831	0.5898
	jaroWinklerCurrentEmployerSimilarity	0.9824	0.6362
	jaccardOfFullProfilesSimilarity	0.9808	0.7834
	jaccardOfProfessionalExperienceSimilarity	0.9824	0.6324
	jaccardOfEducationalBackgroundSimilarity	0.9815	0.5793
	jaccardOfOtherFieldsSimilarity	0.9815	0.5912
	semiVSMOfFullProfilesSimilarity	**0.9776**	0.7418
	semiVSMOfProfessionalExperienceSimilarity	0.9812	0.6407
	semiVSMOfEducationalBackgroundSimilarity	0.9824	0.5831
	semiVSMOfOtherFieldsSimilarity	0.9825	0.5949
	vsmOfFullProfilesToFacebookSimilarity	0.9824	0.5194
	vsmOfFullProfilesToXingSimilarity	0.9805	0.5432
Social Network Topological Based Features	mutualFriends	0.9815	0.5495
	mutualFriendsOfFriends	0.9835	0.5813
	All features	**0.982**	**0.982**

In future research, we may extend this process to more than two networks. Moreover, in this research we assume that a real person has only one user profile page per OSN site. In future work, we may extend this process to find duplicate accounts of a person in the same OSN. In this study we only use the public profile of a user. In future research we may extend this process to deal with private profiles as well.

REFERENCES

[1] Patriquin A., "Connecting to Social Graph: Member Overlap at OpenSocial and Facebook", The compete.com Blog, 2007.

[2] Vosecky J., Hong D., and Shen V.Y., "User identification across multiple social networks", In Proc. of First International Conference on Networked Digital Technologies, 2009.

[3] Veldman I., "Matching Profiles from Social Network Sites", Master's thesis, University of Twente, 2009.

[4] Carmagnola F., Osborne F., and Torre I., "User data distributed on the social web:How to identify users on different social systems and collecting data about them", Proc. of HetRec 2010, USA, pp. 9–15.

[5] Iofciu T., Fankhauser P., Abel F., and Bischoff K., "Identifying users across social tagging systems", ICWSM 2011.

[6] Narayanan A., and Shmatikov V., "Robust De-anonymization of Large Sparse Datasets", IEEE Symposium on Security and Privacy, 2008.pp. 111-125.

[7] Narayanan A., and Shmatikov V., "De-anonymizing Social Networks", IEEE Symposium on Security & Privacy, 2009, pp. 173-187.

[8] Raghavan V. V., and Wong S. K. M., "A critical analysis of vector space model for information retrieval", Journal of the American Society for Information Science, Vol.37 (5), p. 279-87, 1986.

[9] Witten I. H., Frank E., and Hall M. A., "Data Mining: Practical Machine Learning Tools and Techniques" Elsevier, 2011.