Identity Linkage Across Diverse Social Networks

Youcef Benkhedda
Ecole nationale Supèrieure
d'Informatique , BP 68M, 16309,
Oued-Smar, Algiers, Algeria
Email: y_benkhedda@esi.dz

Faical Azouaou
Ecole nationale Supèrieure
d'Informatique , BP 68M, 16309,
Oued-Smar, Algiers, Algeria
Email: f_azouaou@esi.dz

Sofiane Abbar
Qatar Computing Research Institute,
HBKU, Doha 5825, Qatar
Email:sabbar@hbku.edu.qa

***Abstract*—User identity linkage across online social networks has gained a significant interest in the last few years in diverse applications such as data fusion, de-duplication, personalized advertisement, user profiling, and expert recommendation. Existing techniques investigated the use of personal discrete attributes such as user name, gender, location, and email which are not always available. Other techniques explored the use of network relations. In our proposal, we attempt to design a generic framework for user identity linkage across diverse social networks based exclusively on the widely available textual user generated content. We intentionally selected two social networks, Twitter and Quora, which have different contribution models and serve different purposes, and explore different supervised and unsupervised techniques for matching profiles as well as different language models ranging from simple tf*idf vectorization to more sophisticated BERT embeddings. We discuss the limits of different choices and present some encouraging preliminary results. For example, we find that prolific users can be identified with 84% accuracy. We also present a framework we designed to create the largest publicly available annotated dataset for profile linkage in social networks.**

I. Introduction

User Matching approaches have broadly employed text similarity metrics on the user Personal Discrete Attributes (PDAs), such as name phonetic similarity, location format variations or topology matching. However, using PDAs is not always effective due to inconsistency and inaccessibility of such attributes from one Social Network to another. We shed lights on the benefits and limits of using textual generated content for the user matching task. Exploring the user's intrinsic linguistic features for user matching may prove useful for a wide scope of other user-related applications, such as *user profiling*, *online privacy preserving* and *social innocuous activity detection*.

We highlight in this paper our methodology for user profile linkage across diverse social networks using different language modeling techniques to represent profile textual content, such as bag of words, generative probabilistic and deep learning-based models. We also assess the effectiveness of using textual pattern recognition such as named entities (i.e., persons, organizations, locations) and URLs. The hypothesis here is that the same user might be interested in the same set of entities across different social network platforms. Given that user interests may shift over time, we investigate the importance of taking into account the temporal aspect when creating user representations and report some interesting observations. We finally present a system we built from scratch to automate the task of collecting ground linked user datasets. Our experiments are conducted on a real data set we curated from Twitter and Quora, two social platforms that have various differences. Our key contributions include the followings:

- Generation of the largest public user profiles linked dataset with more than 27k account pairs.
- Creation of an end-to-end process to generate ground truth data for the complex task of user matching in social media platforms.
- Evaluation of different matching approaches, combined with different categories of NLP models, we show that with a bag of words and good classification algorithm we can achieve an accuracy rate of 84 percent, without using any elimination process or discrete personal data.

The rest of this paper is organized as follows. Section II formulates user profile linkage problem. Section III explains the setup of the matching approaches we tested. In Section IV we present the dataset we constructed to implement and evaluate our matching applications. Section V presents the preliminary experimental results and the impact of user prolificacy and time features on the matching accuracy. We conclude the paper and discuss some future research directions in Section VI.

II. Problem Formulation

Given two social network platforms SN_1 and SN_2, the user profile linkage problem aims at finding all pairs of accounts $(a_u, a_v) \in SN_1 \times SN_2$ such that $u = v$, i.e. the two accounts $a_u \in SN_1$ and $a_v \in SN_2$ belong to the same user.

Without loss of generality, we assume that each user account a_u in SN_1 has one and only one counter part matching account a_v in SN_2.

III. User Profile Linkage

The fist step in content-based user profile linkage is to create representations of users based on the content they generated. For instance, we consider the user profile as the concatenation of all his generated tweets/answers, respectively in Tw and Qu. Once the representations are established, there are several techniques than can be used to identify the true matching pairs, i.e., those belonging to the same user. In the following, we describe two approaches that we explored: unsupervised user matching and supervised classification. We also show how

we use clustering to improve the time performance of these techniques.

A. User matching approach

The user matching approach performs an exhaustive pair-wise similarity calculation between all the possible account pairs from $\mathcal{SN}_1$ and $\mathcal{SN}_2$. The true matching account for each user is the one having the highest similarity score amongst the rest. In other words, the pair of accounts (a_u, a_v) such that $a_u \in SN_1$ and $a_v \in SN_2$ is considered as belonging to the same real user if and only if $sim(a_u, a_v) >= sim(a_x, a_y), \forall (a_x, a_y) \in SN_1 \times SN_2 \wedge (a_u, a_v) \neq (a_x, a_y)$. The highest score is considered when dealing with similarity metrics such as cosine. For distance metrics such as Jaro–Winkler we consider the lowsest score. The technique is quite effective when dealing with relatively small size of user datasets as shown in Section V-A. However when the number of users exceeds the range of few thousands, the matching performance decreases significantly due to quadratic calculation time.

B. Classification approach

Here we consider the user profile linkage problem as a binary classification problem. We aim to train a classifier that takes as input two user account representations and outputs a binary result, 1 if the two accounts belong to the same user, 0 otherwise.

The labeled data for this supervised learning approach is generated as follows: For each positive example, i.e., true matching pair $(a_u, a_v) \in SN_1 \times SN_2$, we randomly sample N false examples $(a_u, a_y) \in SN_1 \times SN_2 | a_y \neq a_v$. Different values of negative pairs N are tested to study the impact of unbalance ratio on the accuracy of the method.

C. Clustering optimization

The major problem of the user matching approach is the number of comparisons which can be prohibitive. Indeed, for each user account $a_u \in SN_1$, we need to compute similarities with all other accounts $a_v \in SN_2$. One way to tackle this problem is to pre-process user accounts using unsupervised clustering technique. A good clustering will ensure that similar user accounts end-up as members of the same cluster. At run time, given a user account $a_u \in SN_1$ we first identify the cluster to which it belongs, then compute similarities with only members of that cluster to find the candidate matching account a_v. Note that it is perfectly possible to use different types of representations for the clustering and the matching steps. For instance, one could use LDA topics based vectors to perform the clustering (low dimensionality), then use word2vec or TFIDF based representations to perform the matching.

IV. Dataset

We collected the ground truth dataset using an automated crawling and matching framework, based on a username and profile photo matching scheme[1]. An initial dataset of 64k user pairs from Quora (Qu) and Twitter (Tw) were obtained from which we removed all users having less than 10 tweets or zero Quora answer. Pairs with users having private or inaccessible profiles were also removed. As shown in table I, the final collected dataset consist of 27.049 user linked profile pairs with more than 1,295,111 Quora answers and 39,166,748 tweets, containing a total of more than 5.7 billion words. From which 8,212,340 unique word in Twitter and 1,212,090 unique word Quora. Note that we performed a human qualitative evaluation of all pairs to make sure they indeed belong to the same user.

The dataset is structured into two folders (`/Twitter/`, `/Quora/`). Each user account is represented by a single document in each folder with the same 7 digit filename identifier. The file contains posts (tweets or answers) collected from the user profile and ordered by their publication date. Each tweet comes with a timestamp and the tweet text. Similarly, each Quora answer comes with an answer date and text. It is worth noting that the first line in each Quora/Twitter file represents the bio of the user. The dataset is publicly available for test or training use[2].

TABLE I
COLLECTED DATASET CHARACTERISTICS

	#users	#posts	#words	#uniq words
Twitter	27049	39 M	4.4B	8.2 M
Quora	27049	1.29 M	1.3 B	1.2 M

V. Experiments

In order to present a comprehensive comparison of all models, we run all the experiments on the same subset of 500 users (with their corresponding 1000 accounts in Quora and Twitter), randomly selected from the dataset described in the section above. The reason being that we are still running some experiments with larger number of users involving complex representations such as BERT which takes tremendous amount of time.

A. User matching results

We first report the results of user matching technique using different representations.

BERT. Our first attempt was to use the modern and powerful pretrained model BERT [1]. Bert has two main tasks that are next sentence prediction and masked language modeling. The model input size is limited to 512 tokens, which makes predicting raw large documents non possible. One address this issue we can use text summarization techniques to reduce the size of the user documents to Bert readable format (i.e, 518 or less tokens). However, application of such approach is generally expensive on large corpus. Instead, we adopted the approach presented by [2]. In order to classify user reviews based on Bert model sentence prediction, they tested three simple truncation techniques: Head, Tail and Head+Tail, where they selected respectively the first 518 tokens, last 518 tokens

[1]https://github.com/banyous/Quora-and-Twitter-crawler-and-user-matcher

[2]https://zenodo.org/record/4011647.X09yPYZRVH4

and first 128 + last 372 tokens. They achieved the best classification performance results using H+T truncation and the last layer of BERT model as the feature embedding model. In our case, the best accuracy results were obtained using the H+T with a simple Bert sentence prediction model. We got top-1 accuracy of 22%. We concluded for now that BERT might not be a good fit for very large text, as in profile linkage where each user is represented with all their contributions (i.e., tweets and/or answers.)

TFIDF. Next, we tested more simpler embedding models such as TFIDF, LDA, and doc2vec. Our evaluation has shown that the simple TFIDF outperforms all other models by a large margin. We trained the TFIDF model on the union of all documents (tweets and Quora answers). A startling top-1 accuracy of 61% was obtained using low value of maxdf=10, which is a fairly interesting outcome given that a random matching would have achieved an accuracy of $0.2\% = 100 \times 1/500$. To reduce the dictionary size we set mindf=2 to delete noisy terms that occur in only one document in both datasets. Interestingly enough, trying the n-gram range to (1,2) increased the top-1 resulted in a 6% improvement of the accuracy bring it up to 67%. However, it is important to note that n-grams also increased the size of the vectors from 73K to 692k dimensions (unique tokens), which makes it hard to apply to very large datasets. We also observed that pre-processing text, like removing stop-words and stemming, had a negative impact on results. This can be explained by the fact that limiting the max-features or pre-processing the text leads to the disappearance of rare words (misspellings) that are key to distinguish between users textual footprints.

B. Classification results

We tested different document classification algorithms such as logistic regression and support vector machines on TFIDF and LDA user account vectors. Different negative sampling sizes were tested, with values of N set to 1, 3, 5 and 10. Sampled pairs were constructed through mathematical addition of the two accounts feature vectors. For each value of N, we generate 30 different randomly sampled datasets that are split into training set (80%) and test set (20%). We consider the average accuracy of the 30 iterations as the classifier accuracy. In our case, we are more interested in the positive pairs prediction accuracy, which is the classifier precision or the true positive accuracy (TPA). In Figure 1, we present the classifiers confusion matrix for $N = 1$. We notice that the accuracy of true positives are relatively low compared to the results obtained with user matching technique, with KNN and RF having the highest TPA of respectively 34% and 30%. Increasing N values decreases significantly the TPA to less than 2% for $N = 10$. This is the result of the classifier predicting all new pairs as negative matching, which is the dominant class in the imbalanced setup.

C. The impact of clustering

We tested different clustering algorithms such as k-means and hierarchical clustering [3] with different values (k) of

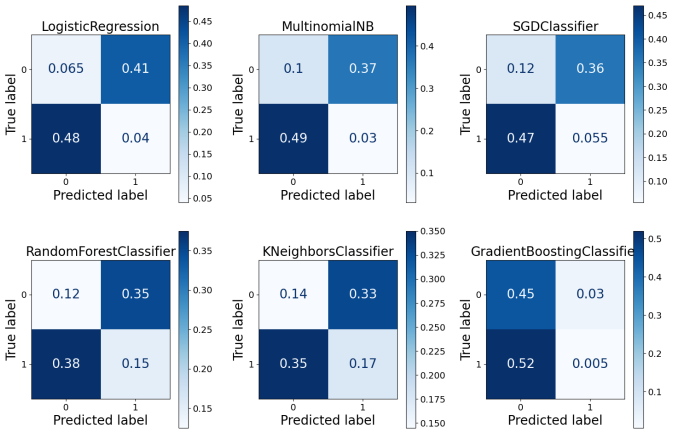

Fig. 1. Percentage accuracy confusion matrix of 6 classification methods applied on S1 sample for ratio of false pairs (N=1) and TFIDF vectors

number of clusters. We used LDA embeddings to compute similarities during the clustering step. We observed that small numbers of clusters lead to better overall linkage accuracy. However the clustering distribution were highly imbalanced. On the other side, increasing the number of clusters reduces significantly the size imbalance among clusters, but affects the overall accuracy of correctly clustered pairs. The best results were obtained for $k = 100$ with sub-space clustering technique which yielded an accuracy of 40%. Using the combination of other embedding techniques such as Bert or Doc2vec produced much lower accuracy.

D. Impact of user prolificacy

Users in our dataset have different volume of activity in Quora and Twitter. In order to understand how user prolificacy (i.e, volume of produced content) affects the effectiveness of profile linkage, we tested the best performing technique, i.e., TFIDF user matching on four user groups: (a) users with at least 10 tweets and 1 answer, (b) users with at least 100 tweets and 10 answers, (c) users with at least 1000 tweets and 30 answers, and finally (d) users with at least 5000 tweets and 100 answers. The number of users of (a),(b),(c) and (d) groups is respectively 27049, 13064, 1501 and 241 users. Figure IV shows the top-1, top-3, top-5, and top-10 accuracy results achieved for different values of maxdf (1, 5, 10, 50). One striking results is that the accuracy of profile linkage significantly increases when the amount of available data increases. Indeed, we notice that linking accounts in group (*d*) can be done with 84% accuracy for maxdf=5. This result is explained by the fact that active users have more available textual features that makes tfidf more efficient in distinguishing between their content. The worst results are observed in group (a).

In order to check the reliability of the previous results, we tested the previous matching algorithm on a greater dataset containing the top active users from (d) merged with normal users from (a),(b), and (c). Using cross-validation, we generated 20 test samples with each sample containing the 239 top

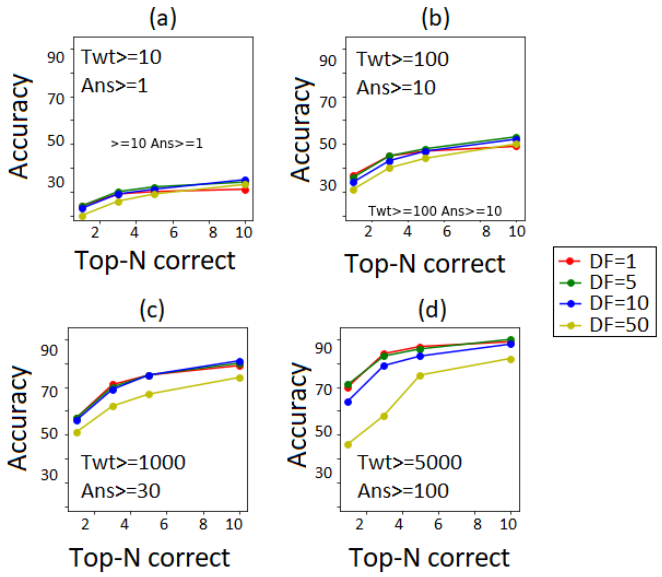

Fig. 2. Different top-k accuracy values for TF-IDF matching with MAXDF varying in range of 1,5,10,50

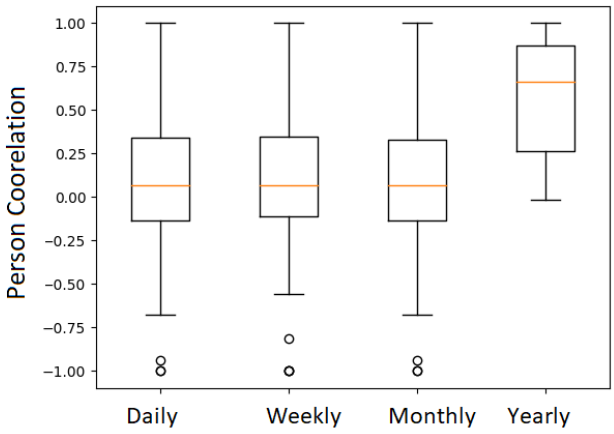

Fig. 3. Periodic Pearson Correlations between Qu and Tw users (S1 users).

active users and 1195 randomly selected users having less than 500 tweets and 10 answers, which makes a rate of 1 active user for each 5 normal user in the dataset. Using this approach we could identify users in group (d) with a top-1 accuracy of 73 %, which is a result close to what we observed earlier when only users from group (d) were considered as candidate matches. This shows that highly active users can still be highly identifiable even when they are part of a considerably larger group of less active users.

E. The impact of temporal aspect

We discuss in this section some ideas that we tried but did not yield to significant improvement on the overall quality of profile linkage task. First, we tried to use the posting dynamics of users as a filter to reduce the search space. Our intuition was that the Tw/Qu time-series correlation of the same user is more likely to be higher than the time-series coorelation of two accounts that belong to two different users.

We tested the correlation relations between each user's accounts. We perform this task by applying Pearson correlation on the posting frequency of the common time periods between the two accounts. The selected periods are weeks, months and years, and the tests were applied to the four group of users (a), (b), (c), and (d) from section above. As shown in Figure 3, top average correlation between real users accounts were obtained from yearly frequencies for the four groups. The weekly and monthly correlations turned to be much less significant, which can be due to the difference in nature between posting behaviours on Twitter and Quora.

We also tried for each pair of accounts $(a_u, a_v) \in SN_1 \times SN_2$ to compare, to build the representation vectors only from documents (tweets and answers) posted in the same time period range. We then take the average accuracy of all the common periods as the matching accuracy score for the pair. We tried several period range units such as same week, same month, and same year. But this did not yield any improvement on the results.

VI. Conclusion

This paper presented the results of a preliminary work on linking users accounts across diverse social networks based on textual generated content. Promising results were obtained showing a high precision linkage of user accounts using simple language models such as tfidf with accuracy scores of 67%. Furthermore, based on the analysis we performed on different categories of active users, we found that more data we have about a given user, the higher the chances to correctly link her two profile accounts. Indeed for users who posted more than 5000 tweets and 100 Quora answers, we could correctly link their profiles with an accuracy of 84%.

One major technical issue we faced when learning the user features is the sparsity of extracted data. To address this problem we used feature selection with different ratio parameters. We also implemented feature selection based on Chi-Square method. Although it was not illustrated here, both feature selection methods had a negative impact on the matching accuracy as they tend to eliminate important features such as misspelled words. Another option we would like to consider in the coming weeks is to adapt dimensionality reduction techniques such as PCA for the textual feature selection problem.

Having tested the effectiveness of different language models and pattern analysis techniques, it would be interesting to explore the directions in which we can use these models at larger scales. External similarity frameworks such as Facebook Faiss [4], can play a critical role in narrowing the matching space with a low computation cost instead of unsupervised clustering.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *China National Conference on Chinese Computational Linguistics*. Springer, 2019, pp. 194–206.

[3] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3918–3927.

[4] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, 2019.