User profiles matching for different social networks based on faces identification

Numerical Methods Project

Kahbod Aeini
Mohammadreza Daviran

Project Essay Explanation

1. Introduction

Today almost every person somehow has registered in at least a couple of social networks. People use these social networks to share contents such as pictures, videos or quotes or mainely what we call information or data. Every social network has a special purpose or atmosphere which reflects in types of published content, conversation style, etc. For example LinkedIn mostly is used as the main page for self-presentation or mainely educational and business purposes but in Twitter or Instagram we may share more private contents and hence these two are used for informal communication.

Following users' behaviours is crucial for these social networks applications, as they can recommande people who to follow in the network or analysis of further application developments.

Users may link some or all of their social networks profiles together, but most of them prefer not to mention their profile in a social network in another one. **Hence we suppose a person as a set of profiles joined from different social networks.**

Previous works in this field mostly have been directed to matching profiles by features such as <u>names, friend-graphs, published textual contents and so on</u>. As we mentioned in the first paragraph, people may share totally different contents in different social networks and hence the friend-graph, style of writing and username may differ from a social network to another one, so these methods may lack precision or recall.

In this work the feature we find for matching by, is users' faces. So we propose a new approach of profiles matching **based on publicly available users' images and faces identification.** The face is a unique attribute for humans, that should keep almost unchanged from network to network. Methods have been developed that allow us to detect faces on photos and compare them. Single face images may suffer from positions, perspective and quality problems. Hence we need a more reliable approach because we have to identify the owner's faces among others, even if there is only one person present in a photo.

The contributions of this paper are the following: (1) we propose a novel approach to user profiles matching using face detection and comparison of face embeddings from different social media; (2) we conduct a set of experiments for two popular in Russia social networks VKontakte and Instagram and investigate limitations of our approach in terms of quality and quantity of data. The latter includes answering the following questions:
1. How much data (photos) does effects matching require?
2. How does efficiency (precision and recall) depend on the quality and the quantity of the data?

2. Related Work

As we mentioned in the introduction, most previous works in this field focused on the features like, usernames, biography or nickname or on the dynamic users behaviour like date of posts or profiles update. These features and kind of information are easy to access which is a decent point but on the other hand, they are very noisy, easily faked, not required. Methods of following users' dynamic behaviour have some major disadvantages, as they require collecting of information during some period of user activities and require an unusual method of data representation in different social media, which can vary in their features. _____It is also noticeable that in these methods we don't need any image processing but this approach may require features which can not be extracted from all social media.

3. The Approach

The main idea of our approach is to form a single defining vector representation of a user's profile based on the embeddings of his faces.

Data Collecting

In the first stage of the approach we must build a labelled dataset. For this purpose we collect a set of profiles from VKontakte, which have a link to the user's Instagram profile. (In the numerical project we use a dataset of users' Twitter and Instagram profile)

Face Detection and Embedding

We process photos by two algorithms of **Face Detection** and **Face Embedding**.

Face Detection Algorithm: We apply Multi-task Cascaded Convolutional Networks (MTCCN), which is efficient and is not affected by scaling of the faces.

Face Embedding Algorithm: We apply FaceNet neural network to construct embeddings of extracted faces.

We apply MTCNN pre-trained on the WIDER FACE dataset and FaceNet pre-trained on the VGGFace2. Then this data is filtered.

WIDER FACE: A face detection benchmark dataset, of which images are selected from the publicly available WIDER dataset. We choose 32,203 images and label 393,703 faces with a high degree of variability in scale, pose and occlusion as depicted in the sample images.

VGGFace2: A dataset for recognising faces across pose and age. The VGGFace2 dataset is made of around 3.31 million images divided into 9131 classes, each representing a different person identity.

Filtering

The extracted face embeddings are filtered by two parameters: Filtering by number of pixels (Quality) and by anchors (child faces removing).

Since FaceNet has limitations on the minimum required image quality, we filter images by the number of pixels of the faces. This control on the number of pixels improves the precision and recall of the matching, Quality control also improves a parameter F1-score which we will discuss later, by 4%.

The other filtering parameter is related to the VGGFace2 limitations. FaceNet was trained with VGGFace2. VGGFace2 contains young and mature faces of people but does not contain the faces of babies and small children. Since children's faces have a very small margin between each other, we must remove them from our collected dataset to avoid mismatching.

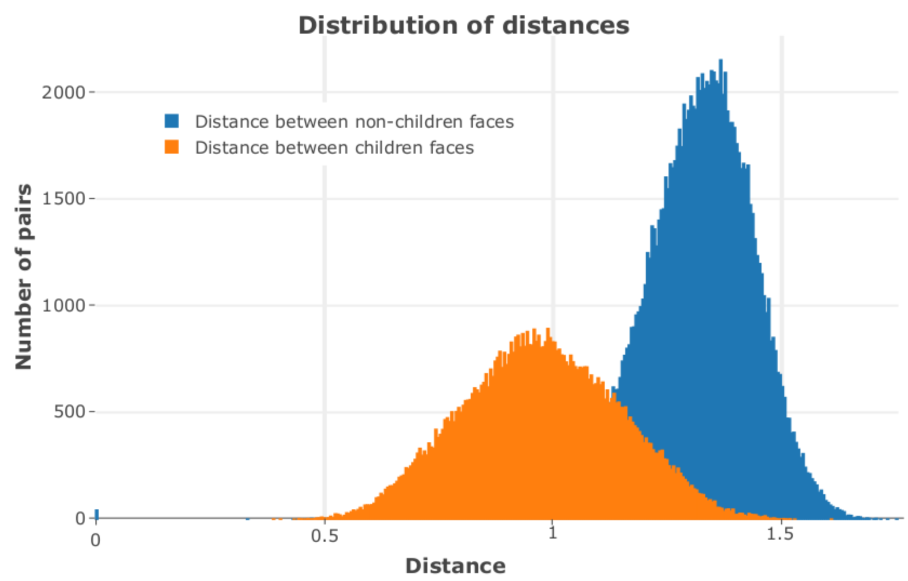


Fig. 1. Distribution of distances between random people faces and between children faces

Above chart reveals that the distribution of distances between embeddings of children's faces has a bias from the distribution of distances between embeddings of random people's faces.

We define anchors to accomplish additional filtering, we use anchors to represent the face of children. An anchor is a vector that represents some space of embedded faces. The anchor was created by collecting faces of children semi-automatically. Then we build an anchor-element-wise mean of all vectors of children's faces. So all face embeddings close to this anchor are removed from the dataset.

Owner Identification

This is the main part of our approach that is performed separately for each profile in each social network. Embeddings of faces are formed in Euclidean space. **We apply hierarchical clustering for each profile separately with the single linkage algorithm and distance threshold**

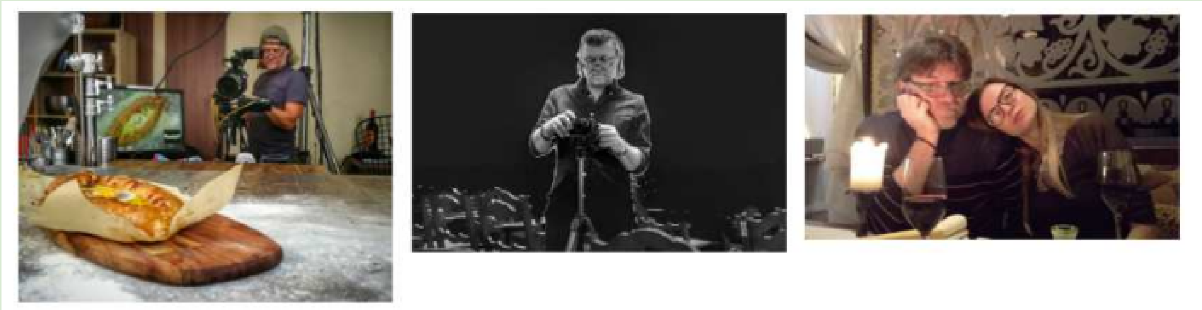
0.8. This algorithm allows us to generate a non-fixed number of clusters based on the Euclidean distance between face embeddings.
<u>Each cluster of the profile should belong either to a single person in the real world, whose faces have slightly different but close embeddings or to persons who look very similar due to distortions introduced by hairstyle, put on glasses, beards and other things which make them look similar.</u>

Most users publish group photos (photos with different people) but we assume that the number of their face occurrences is greater than others. So to find the owners' faces we must choose the largest cluster and combine them into one vector - the Defining Vector (DV) of profile using faces from the chosen cluster. **The DV is an element-wise mean of all generated embeddings with the same dimension** (where V - face embedding, n - number of embeddings of the user).

$$\mathbf{DV} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{V_i} \qquad \mathbf{(1)}$$

Sometimes people publish similar or identical photos, hence it is worth checking the other largest clusters. For example in the below figure, we just have 2 unique images in the cluster and we are not able to match this profile using this cluster.

So we can add for example the below figure, which is the second largest cluster, to form a new DV using five unique face embeddings.



Later experiments show that his assumption and the proposed solution allow us to achieve results that exceed the use of one cluster. Experimental results give us the optimal value - 2 clusters. If after clustering there is only one cluster, we use all photos of the user, if there are all clusters with the same size, we set this profile as ”unable to set the owner” and mark as profiles without a pair.

So then, the DV of each profile in both social media represents the user and will be used for matching. If the size of the largest cluster is less than a given threshold, this user is marked as profiles without pair, because it is not possible to detect the owner’s face correctly.

Profiles Matching

Defining vectors of users from two social media are compared with each other. We calculate the L2 norm between profiles in two social media, for each profile in one social media we find the profile from the other with the smallest distance and mark as a candidate for matching.

$$\textit{argmin}\ \mathrm{L2}\,(DV_i^{VK}, DV_j^{Inst}) = \{\, DV_j^{Inst} \mid DV_k^{Inst} \in DV^{Inst}: \mathrm{L2}\,(DV_i^{VK}, DV_k^{Inst}) > \mathrm{L2}\,(DV_i^{VK}, DV_j^{Inst}) \,\} \qquad \textbf{(2)}$$

If the smallest distance is higher than the given threshold, this means there is no pair in the other social media or we could not find it.

4. Experimental Study

Details of the experimental part

Experimental plan

The experimental plan has three main steps: **1.**Baseline evaluation using real names-based matching; **2.**Evaluation for full profiles without any limitations; **3.**Evaluation with alignment rate reduction and photos number reduction.

Dataset description

The dataset used in the project is Dataset4675, which consists of 4675 profiles from VKontakte and 3100 profiles from Instagram, which simulates working with partially aligned networks - only 3100 VKontakte users have a pair in other social media. Dataset4675 users have from 50 to 500 publicly available photos. (Since the dataset of the project is not accessible for us, we used the given dataset, which consists of 20003 twitter profiles that have the user instagram profile link in the bio)

Metrics

We clarify definitions of **precision**, **recall** and **F1-score** that we use for this classification problem.

With V as a number of all real pairs in our dataset, K^p as a number of the correct predictions of the algorithm (correctly matched pairs of the two social media profiles) and K as a number of all predictions of the algorithm, the **precision** is defined as follows.

$$P = \frac{K^p}{K} \tag{3}$$

And the **recall** is defined as follows:

$$R = \frac{K^p}{V} \tag{4}$$

We need both the recall and precision in order to evaluate our approach, F1-score shows the balance between them and is used to choose the best parameters.

$$\text{F1-score} = \frac{2PR}{P + R} \tag{5}$$

Baseline evaluation. Real names matching

We compare real names of the dataset with the Levenshtein distance metric and analyze sensitivity according to its threshold.

For each user in the first social network we find the closest user from the other network, if the closest distance is greater than the threshold value, the user remains without pair.

The real names are processed in the following sequence: lower case translation, non-alphabetic characters removal, transliteration.

The precision and recall are shown in the below table. The highest F1 of 0.295 is achieved with P=0.765 and R=0.183 and the distance threshold of 4 permutations. This approach achieves a decent precision, but precision decreases by increasing the number of users. The decreasing reason is the fact of the large number of homonyms in the real world.

Table 1. Real name based matching results

Threshold	Precision	Recall	F1-score
1	0.976	0.106	0.191
2	0.972	0.148	0.257
3	0.922	0.169	0.286
4	**0.765**	**0.183**	**0.295**
5	0.511	0.192	0.279
6	0.352	0.198	0.253
7	0.269	0.203	0.231
8	0.235	0.205	0.219

Evaluation for full profiles

Cluster analysis

We first analyze the dependency on the clusters number
in the below table with fixed 0.65 parameter of threshold distance and 6400 image quality 6400.

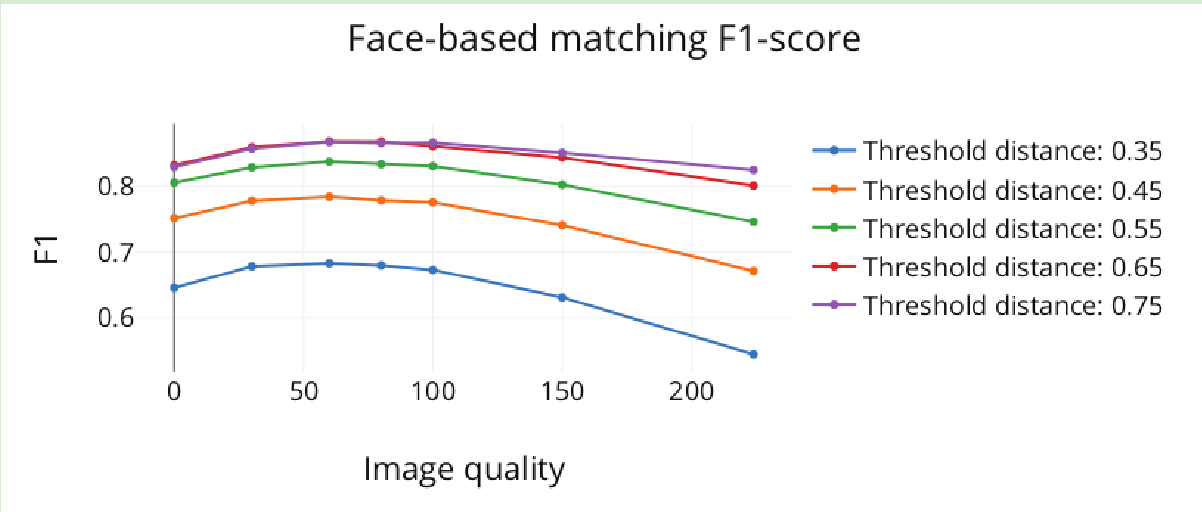
Table 2. Cluster dependence analysis

Number of largest clusters used	Precision	Recall	F1-score
1	0.9617	0.7885	0.8665
2	**0.9782**	**0.7875**	**0.8726**
3	0.9797	0.7839	0.8709
4	0.9793	0.7845	0.8712
5	0.9801	0.7842	0.8713

The above table can be seen as a proof of more than 1 cluster mentioned before. The F1-score in this case is 0.855. The optimal value of the number of the cluster is 2.

Face-based matching

There is a strong dependence between the threshold distance and efficiency, so that by image quality increasing the precision decreases but the recall increases.
The below chart and table were elicited from Dataset4675.



The highest F1 is 0.0868 with image quality 80 and threshold distance 0.65.

Table 3. Face-based matching results

Image quality	Threshold distance 0.35	0.45	0.55	0.65	0.75
			Precision		
0	0.997	0.989	0.976	0.951	0.898
30	1.0	0.999	0.997	0.984	0.933
60	1.0	1.0	1.0	0.995	0.947
80	1.0	1.0	1.0	**0.994**	0.946
100	1.0	1.0	1.0	0.992	0.948
150	1.0	1.0	1.0	0.992	0.948
			Recall		
0	0.478	0.606	0.687	0.739	0.77
30	0.513	0.637	0.709	0.763	0.793
60	0.519	0.645	0.721	0.77	0.8
80	0.515	0.638	0.715	**0.77**	0.798
100	0.507	0.634	0.71	0.761	0.797
150	0.461	0.588	0.671	0.734	0.772

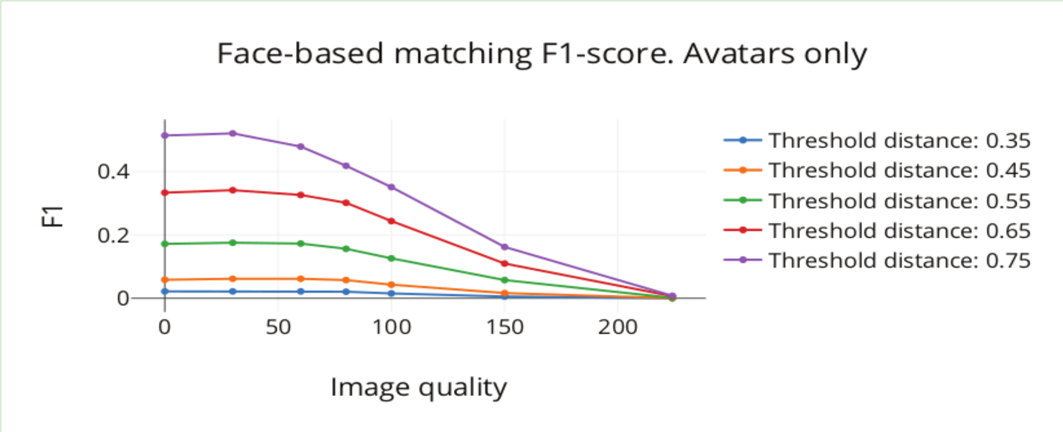
As we can see Precision has an inverse but Recall has a direct relationship with image quality. So since F1-score has direct relationship with precision and recall product and inverse relationship with their sum, we conject the highest efficiency must be around the middle quality images.

Evaluation with the reduced alignment rate and the reduced number of photos

Now we experiment with limited data and rate of alignment of users. Then we can figure out the needed data rate. If the approach requires as much data as possible, it is only applicable for government and law enforcement with social media cooperation.

Avatars only matching

Since avatars are used for presenting the user, only matching avatars, removes the need for the owner detection stage.
The chart below is evaluated from Dataset4675 users' avatars only.



The recall has been decreased in this matching and high value of image quality filter the F1 score is next to zero. The highest F1 achieved score is 0.539 with 0.75 threshold distance and 30 image quality.

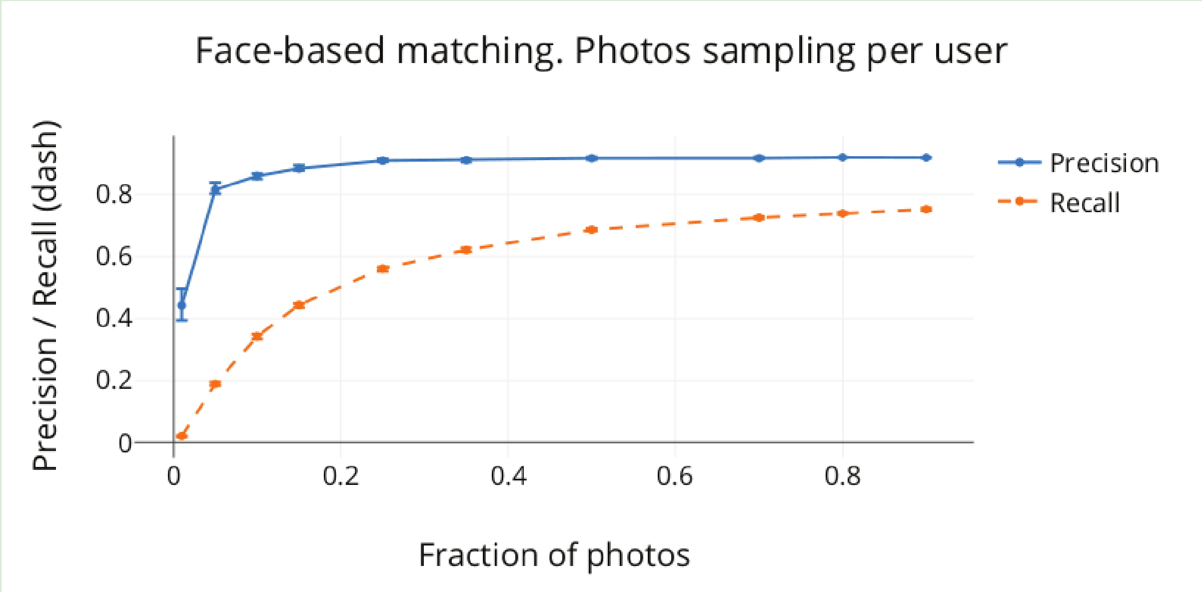
Reducing the number of images for each user

In this section we reduce the number of available photos of each user from Dataset4675 in order to estimate our approach in the condition of greater uncertainty.
So we select X% of the user photos for 10 times. It is interesting that the precision rate remains almost the same even with 10% of data from each user profile of both social media. The reason for the low recall rate is hidden in the owner detection part. The small amount of randomly sampled data does not allow to find the owner's face and to form a good defining vector.

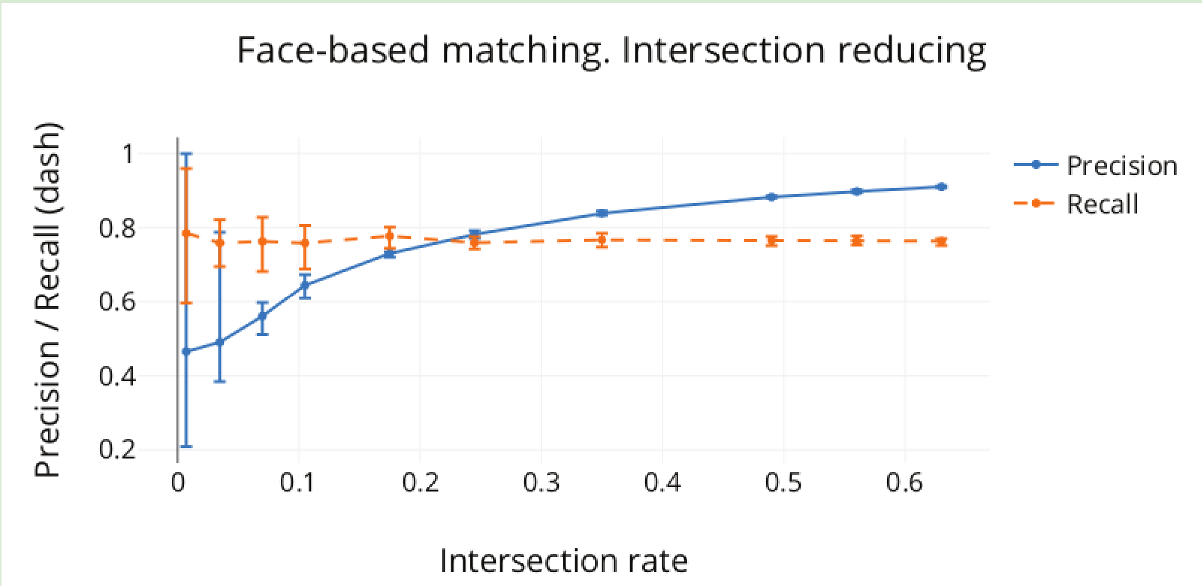
Reducing the rate of intersections. Partial alignment

In the last part of the experiments, we examine the partial alignment of social networks.

The below chart demonstrates the dependence of the efficiency of the algorithm on the proportion of user photos.



In the real world, social media are partially aligned, meaning that for example not all Twitter users have an Instagram account. We can't investigate the real rate of this intersection, but we can consider a number of rate values and create a synthetically reduced intersection.



We match different users, due to random sampling, so some of the chosen users may have more or fewer photos, good or bad (such as biased vectors) defining vectors. Hence the precision and recall have the high variance shown in the above chart. In this chart we can see that the recall is almost stable, which means that the approach can be applied on low-alignment networks and the precision decreases on low-rate alignment because of many false-positive samples, this can potentially be improved by additional filtering.

5. Discussion

The results determine that the faces-based profiles matching with only avatars has a low efficiency and the precision and recall values are 0.375 and 0.963. Reasons behind this low efficiency are the following: **1.** The quality of user avatars are not always enough, this leads to unnecessary filtering and decreasing recall value, there was only 57% of faces from avatars with quality over 80; **2.** Almost 25% of Facebook users have two people on the avatar, so we cannot detect the owner using this kind of images, and the DV is not precise.

Since processing one image generates only one cluster, when we are working only with avatars, we just have one cluster. As it is mentioned before, one cluster gives us less than 0.8665 F1-score. Hence using only users' avatars has low F1-score and efficiency. This aspect and the analysis of results show that very homogeneous clusters lead to mistakes in matching. Using only one image would be a degenerate case of one cluster from one face.

We work with the content of profiles in this study, the recall decreases quickly, but the precision remains almost the same until the 5 to 10 percent of available data, so results indicate that <u>this approach works less efficiently without all available users.</u>

Instagram and VKontakte intersection (Taken from the original essay)

The last thing to discuss the experiments is user sampling. It should be noticed: we do not know the real intersections of people in different social media. According to reports, there are 30 millions of Instagram users in Russia and 80 millions of VKontakte users. Also, we know that 3.3 millions of VKontakte users link their Instagram profile. So, the rough estimate of profile alignment is 3-4%. This value allows us to achieve P=0.49 and the average R=0.758. The alignment rate is probably greater due to historical features: VKontakte is one of the first social media in Russia and it is very popular among active users of the Internet who can be Instagram users. In this case, the alignment is about 30% and the expected precision is 0.8 and the recall is 0.76.

6. Conclusion

This paper was about a method to profile matching across different social media using users' photos. The approach uses photos to form a single feature-vector using embedding techniques and use this vector (the DV) for further profile matching.

When 70% percent of users have profiles in both social medias, the approach achieves precision to 0.994 and recall to 0.76. Profiles also can be matched with limited data, as we discussed in the Experimental Study section.

This approach provides a large number of applications. For example it can match a set of criminals faces from street or security cameras with their profiles in social media. Likewise in scientific purposes, additional information could help to find new features of the user behaviour and open new opportunities in the research of social media impact on the person.