Automatic Embedding of Social Network Profile Links into Knowledge Graphs

Hussein Hazimeh, Elena Mugellini,
Simon Ruffieux, Omar Abou Khaled
HumanTech Institute
University of Applied Sciences of Western Switzerland
Fribourg, Switzerland
firstname.lastname@hefr.ch

Philippe Cudré-Mauroux
eXascale Infolab
University of Fribourg
Fribourg, Switzerland
pcm@unifr.ch

ABSTRACT

Recent Knowledge Graphs (KGs) like Wikidata and YAGO are often constructed by incorporating knowledge from semi-structured heterogeneous data resources such as Wikipedia. However, despite their large amount of knowledge, these graphs are still incomplete. In this paper, we posit that Online Social Networks (OSNs) can become prominent data resources comprising abundant knowledge about real-world entities. An entity on an OSN is represented by a profile; the link to this profile is called a social link. In this paper, we propose a KG refinement method for adding missing knowledge to a KG, i.e., social links. We target specific entity types, in the scientific community, such as researchers. Our approach uses both scholarly data resources and existing KG for building knowledge bases. Then, it matches this knowledge with OSNs to detect the corresponding social link(s) for a specific entity. It uses a novel matching algorithm, in combination with supervised and unsupervised learning methods. We empirically validate that our system is able to detect a large number of social links with high confidence.

CCS CONCEPTS

• **Computing methodologies → Knowledge representation and reasoning**;

KEYWORDS

Knowledge graphs, Profile matching, Classification, Clustering, Online Social networks, Data Integration

ACM Reference Format:
Hussein Hazimeh, Elena Mugellini, Simon Ruffieux, Omar Abou Khaled and Philippe Cudré-Mauroux. 2018. Automatic Embedding of Social Network Profile Links into Knowledge Graphs. In *Proceedings of The Ninth International Symposium on Information and Communication Technology (SoICT 2018).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3287921.3287926

SoICT 2018, December 6-7, 2018, Danang City, Viet Nam

ACM ISBN 978-1-4503-6539-0/18/12…$15.00
https://doi.org/10.1145/3287921.3287926

1 INTRODUCTION

Recently, knowledge representation has become more powerful for end-users by releasing knowledge graphs. These graphs allow users to visualize knowledge facts about real-world entities (nodes) and the interrelations between them (edges), stored in the form of RDF triples. It incorporates knowledge from structured repositories such as DBpedia, or by extracting knowledge from semi-structured web resources such as Wikipedia. The term knowledge graph has first appeared in 2012 [1], when Google introduced a so-called knowledge panel in its search results. It allows users to visualize consolidated knowledge from heterogonous data resources such as personal websites and social media channels in a unified and categorized panel. Similarly, other knowledge graphs such as Wikidata [2] and YAGO [3] [12] did construct their KGs. They allow individuals to visualize knowledge facts about entities in a graphical representation. A user must enter an entity name as a keyword query and gets a knowledge graph as an outcome. However, none of the existing knowledge graphs are totally correct or complete. Hence, they encounter several shortcomings. Existing knowledge graphs do not cover all entitiy types nor do they contain complete knowledge about existing entities.

We motivate the contribution beyond this work (see section 2) as follows: first, it is very important to embed OSN profile links of entities in a knowledge graph, as these profiles contain extensive and real knowledge about entities. Second, we shorten the time for searching manually for such OSN profile links, and investigate an automatic approach to embed them into a knowledge graph.

The main research question we address in this work is: given an entity and its knowledge graph, what are the potential corresponding OSN profile links of this entity? which is not easy to answer. For example, there are 75,980 users registered on Facebook under the name "John Smith" [4], with each one of them having the exact screenname, i.e., the exact first name and last name appear on his profile. This poses several ambiguity challenges. A typical way to solve such a problem is by comparing the attributes or the content of both profiles. With each attribute matched to the other attribute using a specific similarity metric (syntactic/attribute or semantic/content). We start our process from a seed knowledge graph, specifically from Wikidata. We extract information about this entity from Google Scholar. We integrate the information from Wikidata and Google

[1] http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html
[2] wikidata.metaphacts.com/
[3] https://gate.d5.mpi-inf.mpg.de/webyago3spotlx/SvgBrowser
[4] https://www.crunchbase.com/organization/facebook

Scholar into a single knowledge base. We utilize this knowledge base in the matching process. We continue by matching the existing knowledge base with Facebook OSN (F-Link algorithm; see section 6). Then, we use the existing knowledge in this matched Facebook profile to find corresponding OSN Twitter profile link (T-Link algorithm; see section 7). To detect other OSN profiles, we utilize a variety of novel features. Despite other existing profile matching algorithms that rely on profile attributes and content, we employ two novel matching features: life events and profile description. To reveal the correct profile, we use both supervised and unsupervised machine learning methods. We first use an unsupervised approach. Then, we use supervised approaches, mainly SVM, Naive Bayes and J48 decision trees.

The rest of the paper is organized as follows: in Section 2 we illustrate the motivation of the work, in Section 3 we discuss the related work, in Section 4 we define and explain our framework, in Sections 5 to 9 we present our framework and both the F-Link and T-Link matchers, in Section 10 we present experimental results, in Section 12 conclusion and potential future work.

2 MOTIVATING EXAMPLE

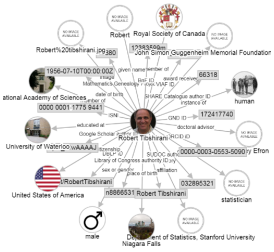

Figure 1: Wikidata knowledge graph.

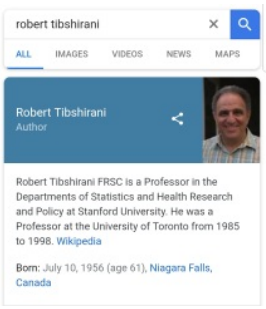

Figure 2: Google knowledge panel.

In Figure 1, we show a knowledge graph from Wikidata [5] of a researcher named "Robert Tibshirani". Although there is a good amount of existing knowledge, none of his social profile links are available. The same goes for Figure 2, where we show a Google knowledge box. It includes basic information such as: profile picture, date and place of birth and finally a short description from Wikipedia. However, his OSN profile links are also missing. In addition to both figures, we illustrate how many OSN profile links are available on Wikidata and Google for different entity types. In Table 1 also we show an example of 100 entities/each type (6 types shown in the table) from Wikidata and Google. For each type, we show the corresponding number of potential social profiles available. As we observe, the lowest count is for the scholarly profiles (type: academic person).

3 RELATED WORK

We divide the related work section into two areas: approaches for knowledge graph completion (KGC) and approaches for profile matching (PM) in social networks.

[5]https://wikidata.metaphacts.com/resource/wd:Q3938444

Table 1: Availability of social profile links for different entity types on Wikidata and Google knowledge graphs)

Entity type	Wikidata	Google Knowledge box
University	54	64
Actor	76	88
Politician	79	90
Academic person	**2**	**4**
City	28	44

3.1 Knowledge Graph Competion

Approaches for KGC usually aim to increase the coverage of a knowledge graph. It predicts missing knowledge such as entities, type and relations between entities as well. These approaches are composed of internal and external methods. Internal methods use existing knowledge in the knowledge graph itself to predict missing knowledge. On the other hand, external methods use knowledge from other knowledge graphs or large text-corpora. In the work of West et al. The authors in [24] employs web search engines to complete missing relations in knowledge graphs. The authors first formulate a structured query and pass it to a search engine. Finally, they use information retrieval and extraction techniques to complete knowledge graphs. The authors in [14] proposes the use of reinforcement learning to fill missing values in a knowledge base efficiently. The authors mainly address three basic attributes, email, job title and affiliation of professors inside an institution. Similarly to [24], they also uses web search engines and information extraction techniques to extract information from the web. In the studies that use social media to construct knowledge graphs such as [22] work on crawling, parsing, annotating and analyzing social media content to create new knowledge graphs from this content. Works that creates scholarly knowledge graph such as [18] work on integrating data from heterogeneous resources and metadata from DBLP and Microsoft Academic graph to create new scholarly knowledge graphs. Similarly, [8] consolidates data from different resources to build new knowledge graphs about artists using the LSH technique. NOUS [3] is an approach for constructing domain knowledge graphs through integrating existing knowledge graphs and information from external resources. The most similar work to ours is [19], in this work, the authors suggest social networks for increasing the coverage of knowledge graph. Specifically, they found what people say on social networks about real-world entities in a knowledge graph. They use three resources mainly: Google+, Facebook, and Twitter.

3.2 Profile Matching

Approaches for PM in social networks are similar to approaches for record linkage in traditional databases. Both tackle the problem of finding entities that are the same and merging them. In traditional record linkage, the attributes under comparison are the fields in a database table, and the entities merged are the records in this table. However, in social network sites, the attributes under comparison are the profile attributes or profile content using both syntactic and semantic similarity computations. Approaches based on profile attributes, such as name, locations, profile image, etc. [1, 21] and

the others that are based on profile content such as timestamps, the topic of a content, behavior, etc. [9, 16].
The authors in [17] starts by searching similar profiles using email addresses. However, if this search fails, the authors conduct the matching by relying on a set of public profile attributes, age, gender, location and other attributes as a first step. The second step is to decide on whether the profiles do really match. For this task, they employ boosting machine learning algorithm. In addition to the profile attributes, this research extracts attributes from profile content. Vosecky et al. [23] goal was to identify users across multiple social networks based on profile matching. They represent each profile by a vector of information. The elements of each vector are the profile attributes themselves. In this paper, the authors assign weights for each profile attribute. To detect the similarity between profiles, authors distinguish between three classes of matching: partial, exact and fuzzy matching. Approaches based on content such as [20] correlates user profile information between del.icio.us and Flickr by leveraging tag-clouds used by users across these two networks. The authors in [13] matches users between Delicious, Flickr and Stumble Upon social networks through leveraging tags used by users and usernames. For tag comparison, the authors used TF-IDF and BM25, for username comparison, they used a set of text similarity functions: such as LCS.

4 FRAMEWORK DESCRIPTION

Our framework is composed mainly of four main components detailed in figure 3:

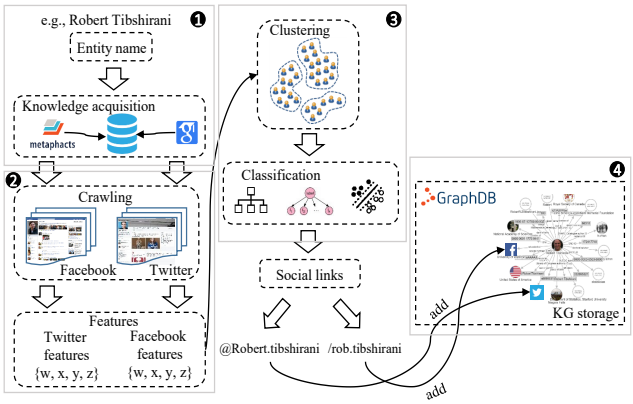

Figure 3: Framework main components.

(1) Knowledge acquisition: we extract knowledge from scholarly resources (Google Scholar mainly), and knowledge graphs from Wikidata Metaphacts. (2) Profile matching: this component matches the knowledge base built in component 1 on the extracted knowledge from social networking sites (Facebook and Twitter). It is composed of two matchers: F-Link and T-Link. (3) Machine learning: this component uses a bottom-up paradigm, it starts by clustering the outcome profiles from component 2 and then uses a binary classifier to find out the matched profile. (4) Enrichment and storage: in this phase, we add the resulting OSN profile links to the knowledge graph (mainly we add all results to the Wikidata knowledge graph stored locally), and we use the GraphDB RDF triple as a storage repository to query and visualize our knowledge graphs.

Table 2: The notation used throughout this paper

Symbol	Description
FB / TW / GS	Facebook / Twitter / Google Scholar
P, a_i	Profile, Attribute
e / b	Life event / Biography
KB, KB_f	Knowledge base, Knowledge base from Facebook
KB	knowledge base
C_i	Cluster

5 KNOWLEDGE BASE CONSTRUCTION

We construct the initial KB by incorporating data from both GS and Wikidata (WD). Hence, KB=GS⊕WD. GS is a scholarly web service that provides information about the scientific community, usually entered by humans. Wikidata is the largest-scale knowledge graph with more than 15 million [6] existing instances.
Our system is keyword-based; it initially requires an author name query, for example: "Robert Tibshirani". We then retrieve the corresponding knowledge graph from Wikidata and data from GS. Knowledge from both resources is integrated into one unified KB. The extracted data from the GS profile is called screen data, i.e., it appears on the profile. For example, the affiliation is: University of Stanford. In this work, we extract further knowledge that does not exist on the profile screen. We extract information from the authors' publications, more specifically, from biographies that exist at the end of each PDF file. Although not all publications include a biography, we verify that at least one of them includes a biography. In table 4, we show the number of biographies found per GS author profile. The information extracted from a biography are: affiliation, position and research interests.

Table 3: The similarity metric used for each feature

Feature	Method	Type
Location	Jaro	Syntactic
Screenname	Jaro, n-gram	Syntactic
Affiliation	Jaro (> 2 tokens, Cosine)	Syntactic
Life event	Cosine, LDA	Semantic
Bio (social networks)	Cosine, JaroWinkler	Syntactic
Bio (publication)	LDA, Cosine, JaroWinkler	Semantic

To extract information from a biography, we use a rule-based information extraction tool, namely GATE (The General Architecture of Text Engineering). It successfully annotates and extracts main attributes inside a text. It has proven an average precision and recall of 90-95% on extracting this kind of information [4].
In Table 3, we show the similarity metric for each attribute used along this work (Bio is biography in the table).

[6] http://www.wikidata.org/wiki/Wikidata:Statistics

Table 4: Total number of biographies found per each entity (using GS as a resource)

Full name	#of papers found	#of biographies
Robert Tibshirani	783	125
Trevor Hastie	555	77
Chris Pal	133	13
Ben Shneiderman	1301	195
Petter Bae Brandtzaeg	86	10

6 F-LINK MATCHER

The information extracted from the FB profile P is represented in the knowledge base KB_f. We crawled only the information that is visible to the public. It mainly includes the following attributes: screenname (the first name and last name), a biography (a short description about the profile owner. This attribute is always public), a content (collection of posts, sharing, etc.), and other structured profile information, such as workplace and living place. The outcome of F-Link matcher is a matched profile that has a link L_f, where $L_f = KB \otimes KB_f$.

6.1 F-Link feature extraction

The features extracted are the scores of the similarity between two attributes, for example, the Jaro similarity between a name from KB and a name from KG_f. We use both syntactic similarity and semantic methods to compute the similarity between attributes. However, although these methods are good enough to give us a matching score, in some cases, the match does not occur. Let us consider the following example: an entity has a name "Robert Tibshirani" on GS while its corresponding name on FB is "Rob Tibshirani"; in this case, the score of matching between two names using the Jaro similarity, for example, is 0.6. Such score is not high enough to decide that they can be matched. Hence, in this case, we must use alternative or complementary metrics such as n-gram. It finds the occurrence of a substring inside the other one, i.e., the occurrence of "Rob" inside the string "Robert".

6.2 Syntactic matching

We compare the syntax of attributes that have no semantic context, such as names. Usually, this type of information is composed of one or two strings (e.g., when the first name and last name appears on a GS profile). Another example is the affiliation or the place of work, for instance, "Stanford University", or complete affiliation such as, a = "Professor at the University of Stanford". In this case, if we have two or more tokens inside the text, we extract the named entities. Let us consider the previous example with affiliation a, we employ Stanford NER to find the entity list (e) in a, so, e will contains the following entities e = ["University of Stanford": Organization, "Professor": Object]. We employ the JaroWinkler distance metric to find the similarity score among strings in the list e.

Before we conduct a similarity computation across two affiliations, we normalize our data by removing stop words from the text. For example, if a = "professor of machine learning at the University of Fribourg", the stop words removed are: "of", "at" and "the". In case the affiliation consists of more than two tokens, we use the cosine similarity (1). Where s_1 and s_2 represents two affiliations respectively, and s_1 and s_2 are > 2.

$$\cos(\mathbf{s_1}, \mathbf{s_2}) = \frac{\mathbf{s_1}\mathbf{s_2}}{\|\mathbf{s_1}\|\|\mathbf{s_2}\|} = \frac{\sum_{i=1}^{n} \mathbf{s_1}_i \mathbf{s_2}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{s_1}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{s_2}_i)^2}} \quad (1)$$

6.3 Semantic matching

When the attribute has a semantic context, we conduct a semantic matching between two pairs of information as follows: we first detect and extract the topics and then find the textual similarity between them; this done (textual similarity) using Jaro similarity (in case we have 1 string) or Cosine similarity (in case we have 2 or more strings). An example of topics that can be extracted are: a biography contains a research interest "cloud computing", and on FB, a life event describes a conference event and on TW also a life event about conference. Hence, all the aformentioned topics must be detected. For this task we employ Latent dirichlet allocation (LDA) [2] to find the topics in a biography, as well as for finding the topic of the events.

6.4 F-Link matching algorithm

In Algorithm 1, we show how the procedure of matching the existing KB with a FB profile works. We start by querying the FB with the name from KB. We get a list of profiles that have the exact name or similar names. For each profile, we get a list of public information containing attribute information and content information. The information retrieved from a FB profile is represented by KB_f. Then, we start by matching each possible attribute between KB and KB_f using both syntactic and semantic methods. Then, we get a vector of scores that represents the similarity score for each pairwise attribute comparison. Finally, we cluster and classify these vectors to find the corresponding TW profile link L_f.

Algorithm 1 F-Link algorithm

```
function F-Link(KB, FB)
KB ← GS ⊕ WD
KB_f = GetKB_f(KB.name)
foreach: P in KB_f
    foreach: a_i in P
    if isSyntactic(a_i) then
        x = MatchSYN(A, KB.A)
    if isSemantic(a_i) then
        y = MatchSEM(a_i, P.a_i)
v_i = [x, y]
v.add(v_i)
C_i =Cluster(v)
P =Classify(C_i)
return P
function GetKB_f(String Name)
    ProfilesList = queryFB(Name)
foreach: P in ProfilesList
    new KB ← P
    KB_f.add(KB)
return KB_f
```

7 T-LINK MATCHER

In a previous work, we have proposed SocialMatching++ [11] a work specifically focusing on matching FB to TW OSN profiles using two novel features (biographies and life events). In this work, we prove how these two features can enhance each other to yield more potential matchings than other approaches. In a previous work to SocialMatching++, we have investigated a thorough review of the related state-of-the-art profile matching algorithms [10].
In this section, we match data between FB and TW by using basic profile attributes, such as screennames, location, or affiliations. However, some of these attributes might be useless due to missing information or privacy issues. In this context, we propose the employment of novel matching attributes that cannot be the same for two entities holding the same name. The two proposed features are both syntactic (attribute based) and semantic (content based). For the syntactic, we use the profiles biographies because its always public [11]. In addition, other basic profile attributes are set private by users. Therefore, using this profile attribute would result in more efficient matching outcomes. For the semantic one, we used life events. A life event is a temporal event that usually social networking users publish on their social accounts. Despite other approaches that use the semantic similarity between profile pairs to detect the matching across entities, their approaches are based on only comparing profile content, neglecting that the activity rate is totally different among two different OSNs.
Consequently, in our TW to FB matcher, we used two additional key attributes which are the biography and life events.

7.1 T-Link feature extraction

The features we use represent similarity scores between two pairs of attributes from FB and TW. In T-Link, we calculate and obtain three features, the syntactic similarity score between screennames using JaroWinkler and n-gram similarity metrics, the syntactic similarity score between locations and affiliations (if it exists) using JaroWinkler and cosine similarity metrics, and finally the semantic similarity scores of biographies and life events (also if exists) using LDA to extract the event topic.

7.2 Syntactic and semantic similarity

In our description of the F-link matcher, we distinguish between when to usesemantic and when we to use syntactic similarities. Similarly, the matching between FB and TW acts in the same way. Syntactic methods are used to conduct a similarity computation among sytactic attributes (screenname, location and affiliation (< two tokens)). According to the semantic similarity, we calculate the semantic similarity of two pairs of life evens from FB and TW. For instance, if an event published on FB is "New job", we search for a similar life event on the TW profile. By using LDA, we detect if the topic of these two events is the same or not. A number of algorithms already published in this area [5–7, 15] have proposed a solution for finding social events on TW.

7.3 T-Link matching algorithm

In Algorithm 2, we start by searching a name from KB_f in TW. Then we get a list of TW profiles that have a similar or exact screenname. For each profile in this list, we compare its attributes with each attribute in KB_f (if it exists). Unlike the state-of-the-art approaches that rely on the content only, they does not take into consideration the amount of activity between two social networks which could be different. We leveraged life events and profile biographies that contain key information on each profile. Using the syntactic and semantic similarity methods discussed in the previous section, we obtain the scores. For each matching pair, we build a vector with the similarity scores. Then we cluster and classify these vectors to find the corresponding TW profile link L_t.

Algorithm 2 T-Link algorithm

```
1:  function T-Link(KB, FB)
2:  KB_t = GetKB_f(KB_f.name)
3:  foreach: P in KB_t
4:      if isSyntactic(a_i) then
5:          x = MatchSYN(P.a_i, KB_f.a_i)
6:      if isSemantic(a_i) then
7:          y = MatchSEM(P.a_i, KB_f.a_i)
8:  v_i = [x, y]
9:  v.add(v_i)
10: C_i =Cluster(v)
11: P =Classify(C_i)
12: return P
13: function GetKB_t(String Name)
14:     ProfilesList = queryTW(Name)
15: foreach: P in ProfilesList
16:     new KB ← P
17:     KB_t.add(KB)
18: return KB_f
```

8 PROFILES CLUSTERING

To detect the matched profile, we establish a bottom-up machine learning paradigm. We begin by clustering profiles (each profile represent a feature vector). First of all, we cluster all the profiles into 3 clusters. Only the cluster with the highest confidence (C_f) value is considered while the others two are ignored because they have very low similarity scores between profile attributes. We deduce C_f through ranking the clusters based on the average similarity scores. For instance, if we have C_1=0.2,0.2,0.5 and C_2=0.6,0.4,0.5 and C_3=0.8,0.7,0.9, the average of C_3 is the highest = 0.8, hence, this cluster will be considered. Finally, each profile vector from C_f is classified in the classification layer. To chose the C_f we check the highest similarity vectors among each cluster. The formal description of the classification process is described in Algorithm 3.

Algorithm 3 Clustering algorithm

```
1: function Cluster(VectorList v)
2: foreach: v_i in v
3: CompareEachElementIn_v
4:     C_i = new cluster
5: foreach: C_i in C
6:     if C_i.hasHighestSimilarity then
7: return C_i
```

9 PROFILES CLASSIFICATION

Methods to profile matching usually detect the matching profile following a rule-based or machine learning approache. In rule-based approaches, we typically assign a set of rules and a threshold score to obtain the match profile among all other profiles. In machine learning approaches, they treat this problem as a supervised learning problem. They build a feature vector which consists of syntactic and semantic calculations, normalized to a single vector. This task is considered as a binary classifier task, i.e., match or not match. Some supervised algorithms employed to treat this problem are SVM, Decision trees, Bayes naive, KNN and Logistic regression.

Algorithm 4 Classification algorithm

```
1: function Classify(Cluster C_i)
2: foreach: v_i in C_i
3:     if C_i.v_i = "Match" then
4:         P = v_i.profileLink
5: return P
```

In this work, we employ three supervised learning algorithms: support vector machine, Naive bayesian classifier and J48 decision tree. Each profile candidate in the cluster C_f is classified using a binary classifier. The two classes are evenly match or not match. The hierarchy of the bayesian classifier is composed of one parent (class = match/not match) and four children (features). We build a training set containing five hundred feature vectors v = $[x_1,x_2,...,x_n]$. Where x_n is the similarity score among two attributes from different resources. We used this vector as an input to our classification algorithm which results in a match or non-match profile. The formal description of the classification process is described in Algorithm 4.

10 IMPLEMENTATION

All implementations were done using Java. To crawl user profiles from GS, FB and TW, we use the Selenium web crawler [7]. To extract knowledge graphs from Wikidata, we use their API. To add links to Knowledge Graphs, we use the GraphDB RDF repository, as well as to visualize them. A demonstration of our system has been uploaded to Youtube [8], presenting the complete process. The system takes a full name as a query and finally produces a knowledge graph from Wikidata including OSN profile links. The code of biography extraction from PDF publications is also available on Githib [9], and finally the T-Link matcher is available on Github [10].

11 EXPERIMENTAL RESULTS

In this section, we describe the results of this research. We compare our results with baseline methods and systems. We compare both matchers with Wikidata and YAGO and show that our method is capable of finding FB and TW links better than both Wikidata and YAGO. Additionally, we compare the FB to TW (T-Link) matching algorithm with baseline approaches that match user profile across social networking sites.

[7]https://www.seleniumhq.org/
[8]https://www.youtube.com/watch?v=JAZ_VM4u92g
[9]https://github.com/HusseinSwiss/author-name-disambiguation
[10]https://github.com/HusseinHESSO/ProfileLinking_v1.0

11.1 Dataset collection

To test and evaluate the performance of both matchers. We use data from the following sources: a one scholarly resource (GS), a one knowledge graph resource (Wikidata), and two social networks (FB and TW). From GS, we collect the top cited GS profiles on four domains: computer science, physics, chemistry and medicine. For each domain, we collect 200 profiles, and for each profile we extract screen information such as: Full name, affiliation (position and workplace), location (from affiliation - if available) and the number of citations. In addition to the screen information, we extract complementary information that resides inside publications (biography section). For each profile in the GS dataset, we found its corresponding knowledge graph on Wikidata, and collect the existing knowledge graph triples. For the social networks, we collect information on user profiles such as : screenname, living place, workplace, affiliation, in addition to biographies and life events.

11.2 Performace evaluation

We evaluate the performance of finding the correct profile links from FB and TW using both the precision and recall, where precision = $\frac{\#of\,Found\,and\,Correct\,profilesmatches}{Total\#of\,profilesfound}$ and recall = $\frac{\#of\,Found\,and\,Correct\,profilesmatches}{Total\#of\,correct\,profilesmatches}$.

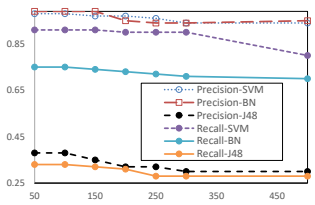

Figure 4: Precision of finding FB links on three classifiers (essential + auxiliary).

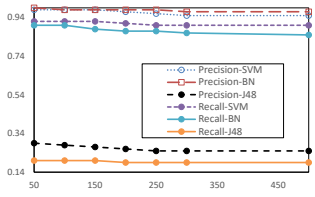

Figure 5: Precision of finding TW links on three classifiers (essential + auxiliary).

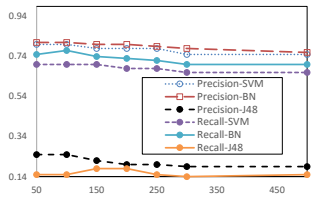

Figure 6: Recall of finding FB links on three classifiers except auxiliary.

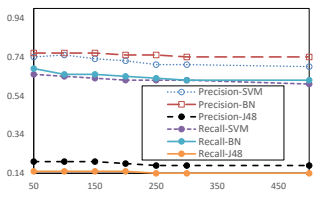

Figure 7: Recall of finding TW links on three classifiers except auxiliary.

In Figures 4 and 5, we illustrate the precision values for finding FB links and TW links respectively when using together essential attributes (profile attributes) and auxiliary attributes (biography and life events), and only essential without auxiliary (Figures 6 and 7). Three methods have participated in this experiments for measuring the potential benefits of our method, SVM, NBC, and J48. The y-axis represent the precision value. We observe that Naive Bayesian Classifier (NBC) has the highest precision and recall values

Table 5: Comparing our system using three diffirent classification methods, with Wikidata and YAGO, on four different domains to find both FB and TW profile links.

	Our approach									
	SVM		NBC		J48		Wikidata		YAGO	
Domain	F	T	F	T	F	T	F	T	F	T
Computer Science(CS) (matched)	121	118	122	118	120	116	3	6	4	7
Physics (P) (matched)	130	138	130	136	129	138	4	8	0	1
Chemistry (C) (matched)	121	120	121	121	121	120	5	8	6	0
Medicine (M)(matched)	132	138	133	138	135	135	12	16	10	6
Existing profiles, CS (F = 125, T = 131), C (F = 134, T = 139), P (F = 130, T = 122), M (F = 138, T = 141), all profiles (F = T = 200), F = Facebook, T = Twitter										

with a comparison to SVM and J48. The precision and recall seems to be higher on FB than TW (see next section). Its shown clearly in the evaluation that auxiliary attributes have raised smoothly the precision and recall values. Therefore, both auxiliary attributes have played a vital role in enhancing the matching probabilities.

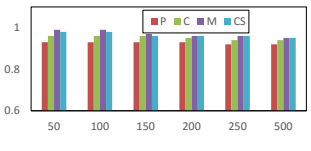

Figure 8: Precision of finding FB links on four domains.

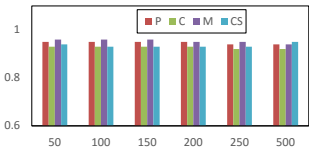

Figure 9: Precision of finding TW links on four domains.

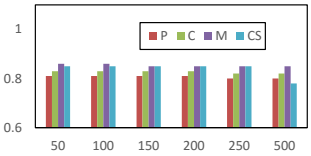

Figure 10: Recall of finding FB links on four domains.

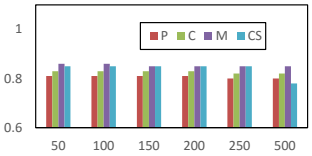

Figure 11: Recall of finding TW links on four domains.

In Figures 8, 9, 10 and 11, we present the precision and recall results of finding FB and TW links compared on four different domains (CS, P, C, M) where the x-axis represents the number of profiles and the y-axis represents the values. By analyzing the graphs, we conclude that our system is more precised on FB compared to TW. This is because: (1) FB information is more systematic, i.e., the profile is more organized than TW. For example, life events on FB can be found under "life events" section inside user profile. (2) FB content is more rich than TW, i.e., the amount of shared posts on FB is more than the amount of shared tweets on TW. The precision as well as the recall values between the 4 domains records the highest in "Medicine" domain. We found that profiles related to physicians usually contained more information than the other profiles. The precision and recall values are evaluated on a small number of profiles. Thus, as we have more number of profiles, the precision and recall will be more accurate, and the classifiers as well, because the training set will be enriched. In the current version of our system, with each found FB or TW link, we add the matching result to the training set to enhance its accuracy.

11.3 Knowlege graph baselines

In Table 5, we present a comparison between our method using three classifiers and the baselines Wikidata and YAGO. We illustrate a comparison between the number of profiles we were able to match on four scholarly domains: computer science, physics, chemistry and medicine. The domains are categorized based on GSś categorization, i.e., to find a set of profiles on GS that have the physics domain, we use the following query for GS (label:physics). We show that our method produces a number of profiles highly exceeding the ones in Wikidata and YAGO. In physics for instance, we have zero YAGO FB profiles. However, there are 122 existing profiles out of the 200 GS profiles tested. According to the performance of the three classifiers, all perform approximatly the same, with a slight positive difference for NBC, and slight negative difference for the J48 algorithm.

11.4 T-link evaluation metrics

To evaluate the performance of the T-link algorithm, we use both precision and recall metrics.

11.5 T-link baselines

We compare in table 6 the precision and recall of T-Link algorithm with the following state-of-the-art approaches:

(1) HYDRA [16]: a system for linking identical user accounts by analysing and comparing the behaviour of users. (2) BM25 [13]: an approach for identifying a user across social networks by comparing their tagging practice and usernames . (3) MOBIUS [25]: they connect user profiles across social networks by comparing the behavioural charecteristics such as timestamp between posts . (4) OPL [26]: an approach for connecting social networking user profiles using internal and external features . In Table 6, we present the results of comparing our approach with these baselines. Its shown that our approach outperforms better than the 3 baselines on precision and recall, and not very far from acheving the same perfrmance of HYDRA [16].

Note: the domains other than "Medicine" performs nearly the same.

Table 6: Comparing the precision and recall of T-link with four baselines

Baseline	Precision	Recall
T-link	0.97	0.88
HYDRA	0.98	0.90
MOBIUS	0.95	0.84
BM25	0.81	0.73
OPL	0.86	0.84

We stored this refined graph in GraphDB repository. To visualize the graph, we use GraphDB visualization tool (Figure 12 illustrates an example). To query the graph we use SPARQL.

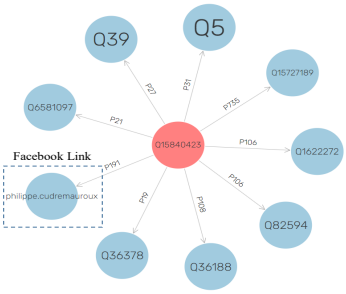

Figure 12: A knowledge graph from Wikidata including two social profile links from FB and TW, visualized on GraphDB. Note: only outdegree relations is shown.

12 CONCLUSION AND FUTURE WORK

In this paper, we presented an algorithm for knowledge graph refinement, by adding social profile links to academic entities. The algorithm works as follows: (1) it takes a name as a query and retrieves its corresponding Wikidata knowledge graph and its corresponding GS profile. (2) It combines information from these two resources into one knowledge base. (3) It matches this KB with the FB social networking platform to derive the corresponding profile and its link (F-Link 1 algorithm). (4) We extracted the information from the FB profile and use it as a knowledge base to match it with TW and found the corresponding TW profile link (T-Link 2 algorithm). The novelty in both matchers 1 and 2 is the use of novel attributes for matching: biographies inside research papers, and life events and descriptions between social networks. Using precision and recall effectiveness measures, we validated the credebility of our system against Wikidata and YAGO. Moreover, we approved that our system outperforms exiting baselines systems in precision and recall. As future work, we plan to find and integrate additional information about entities to further augment the quality of the information contained in the KG, by leveraging additional information from various sites such as personal websites, in order to increase the coverage of a KG.

REFERENCES

[1] N. Bennacer, C.N. Jipmo, A. Penta, and G. Quercini. Matching user profiles across social networks. In *Advanced Information Systems Engineering - 26th International Conference, CAiSE 2014*, pages 424–438.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. In *Journal of Machine Learning Research Volume 3, 2002*, pages 993–1022.

[3] S. Choudhury, K. Agarwal, S. Purohit, B. Zhang, M. Pirrung, W. Smith, and M. Thomas. Nous: Construction and querying of dynamic knowledge graphs. In *33rd IEEE International Conference on Data Engineering, ICDE 2017*, pages 1563–1565.

[4] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: An architecture for development of robust hlt applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL) 2002*, pages 168–175.

[5] T. Dickinson, M. Fernandez, L. Thomas A., P. Mulholland, P. Briggs, and Alani H. Connecting users across social media sites: a behavioral-modeling approach. In *Proceedings of the ACM Web Science Conference, WebSci 2015*, pages 1–2.

[6] T. Dickinson, M. Fernandez, L. Thomas A., P. Mulholland, P. Briggs, and Alani H. Identifying prominent life events on twitter. In *Automatic Identification of Personal Life Events in Twitter*, pages 4–8.

[7] B. Dr Eugenio, N. E. Green, and R. Subba. Detecting life events in feeds from twitter. In *IEEE Seventh International Conference on Semantic Computing ICSE 2013*, pages 274–277.

[8] G. Gawriljuk, A. Harth, C. A. Knoblock, and P. A. Szekely. A scalable approach to incrementally building knowledge graphs. In *Research and Advanced Technology for Digital Libraries - 20th International Conference on Theory and Practice of Digital Libraries, TPDL 2016*, pages 188–199.

[9] O. Goga, H. Lei, S. Hari Krishnan, G. Friedland, R. Sommer, and R. Teixeir. Exploiting innocuous activity for correlating users across sites. In *22nd International World Wide Web Conference, WWW 2013*, pages 447–458.

[10] H. Hazimeh, E. Mugellini., O. Abou Khaled, and P. Cudré-Mauroux. Linking user profiles in social networks: A comparative review. In *International Journal of Social Network Mining (IJSNM) Volume 2 Number 4 2017 (In press)*.

[11] H. Hazimeh, E. Mugellini., O. Abou Khaled, and P. Cudré-Mauroux. Socialmatching++: A novel approach for interlinking user profiles in social networks. In *PROFILES@ISWC 2017*.

[12] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. In *Artificial Intelligence, Volume 194, 2013*, pages 28–61.

[13] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff. Identifying users across social tagging systems. In *Proceedings of the Fifth International Conference on Weblogs and Social Media 2011*.

[14] P. Kanani and A. McCallum. Selecting actions for resource-bounded information extraction using reinforcement learning. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM) 2012*, pages 253–262.

[15] M. Khodabakhsh, M. Kahani, E. Bagheri, and Z. Noorian. Detecting life events from twitter based on temporal semantic features. In *Knowledge Based Systems, Volume 148 2018*, pages 1–16.

[16] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan. Hydra: large scale social identity linkage via heterogeneous behavior modeling. In *International Conference on Management of Data, SIGMOD 2014*, pages 51–62.

[17] M. Motoyama and G. Varghese. I seek you: searching and matching individuals in social networks. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM) 2009*, pages 67–75.

[18] A. Sadeghi, C. Lange, M. E. Vidal, and S. Auer. Integration of scholarly communication metadata using knowledge graphs. In *Research and Advanced Technology for Digital Libraries - 21st International Conference on Theory and Practice of Digital Libraries, TPDL 2017*, pages 328–341.

[19] T. Steiner, R. Verborgh, R. Troncy, J. Gabarro, and R. Van de Valle. Adding realtime coverage to the google knowledge graph. In *International Semantic Web Conference (Posters and Demos) 2012*.

[20] M. Szomszor, I. Cantador, and H. Alani. Correlating user profiles from multiple folksonomies. In *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia 2008*, pages 33–42.

[21] T. Van Le, T.N. Truong, and T. Vu Pham. A content-based approach for user profile modeling and matching on social networks. In *Multi-disciplinary Trends in Artificial Intelligence - 8th International Workshop, MIWAI 2014*, pages 232–243.

[22] J. P. Vargheese, P. Travers, K. Vincet, C. Wallace, and A. Kabedeva. Constructing social media knowledge graphs with social scientists. In *Proceedings of the 30th International BCS Human Computer Interaction Conference, BCS HCI 2016*.

[23] J. Vosecky, D. Hong, and V.Y. Shen. User identification across multiple social networks. In *Networked Digital Technologies, First International Conference 2009*.

[24] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin. Knowledge base completion via searchbased question answering. In *Proceedings of the 23rd International Conference on World Wide Web 2014*, pages 515–526.

[25] R. Zafarani and H. Liu. Connecting users across social media sites: a behavioral-modeling approach. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013*, pages 41–49.

[26] H. Zhang, M-Y. Kan, Y. Liu, and S. Ma. Online social network profile linkage. In *Information Retrieval Technology - 10th Asia Information Retrieval Societies Conference, AIRS 2014*, pages 197–208.