Linking Knowledge Bases to Social Media Profiles

Yaroslav Nechaev
Fondazione Bruno Kessler
University of Trento
nechaev@fbk.eu

Francesco Corcoglioniti
Fondazione Bruno Kessler
corcoglio@fbk.eu

Claudio Giuliano
Fondazione Bruno Kessler
giuliano@fbk.eu

ABSTRACT

Social media have become an invaluable source of data for a wide variety of tasks. Unfortunately, this data is hard to gather and process due to low amount of machine readable attributes, API limitations and noisiness. In this paper we propose a system that aligns knowledge base entries of people and organisations to the corresponding social media profiles. The motivation is twofold: (i) on the one hand, we facilitate processing of social media data by allowing the import of rich entity descriptions from knowledge bases; (ii) on the other hand, we are enabling an automatic enrichment of a knowledge base with additional data from the social media. We used this system to create a resource of 893,446 alignments between DBpedia entities and Twitter profiles. This resource allows, effectively, to connect Twitter to the Linked Open Data cloud.

CCS Concepts

•**Information systems** → *Entity resolution;* •**Computing methodologies** → *Ranking;*

Keywords

Social Media, Profile matching, Machine Learning, Knowledge Bases, DBpedia

1. INTRODUCTION

With social media now being used universally across all countries and communities, the probability of finding a particular person in some social network is higher than ever. This is especially true for public persons and companies, e.g., the ones having a page on Wikipedia, that have to engage with their supporters online and share their work in order to maintain and expand their popularity, public image, or gather feedback. Thus, social media have become a primary source of information about such entities, providing user profile data (e.g., location, job, interests), social connections, user-generated content (e.g., text, images), and so on. Importantly, this source is generally more up-to-date than collaboratively curated resources like Wikipedia, where new information can be added with a significant delay. By extension, popular public knowledge bases (KB) derived from Wikipedia, such as DBpedia,[1] as well as KBs resulting from voluntary collaborative efforts such as Wikidata,[2] cannot be reliably used as sources of live knowledge, since changes in such KBs do not occur instantly. Entries about recent events, such as deaths, elections, interviews, can lag behind from hours to months, depending on the amount of people in the community caring about those topics. Such delay can prevent using these KBs in some areas.

In order to integrate up-to-date social media with KBs, a first step is to establish reliable and comprehensive alignments between social media user profiles and KB entities. These alignments can be leveraged to augment the KB with data from the social media, or to inject background knowledge from the KB in social media analysis tasks such as user profiling, named entity linking in user-generated content and others. Although such alignments do exist for a few entities in DBpedia and Wikidata, they only cover a very small portion of all the persons and organisation entities they contain.

In this paper, we aim to bridge this gap by introducing a significant amount of explicit links between social media profiles and KB entities for people and organisations. In order to do that, we propose an approach for aligning KBs to social media that consists of two steps, exemplified in Figure 1. In the *candidate acquisition* step, given a target KB entity we query the social network API to retrieve a list of matching candidate profiles, using different query assembly strategies that aim at maximizing recall and efficiently using the request quota provided by the API. In the *candidate selection* step, a machine learning-based classifier (SVM or deep neural network) selects the best possible candidate, using handcrafted features derived from attributes in the KB and the candidate profile that can be reliably found in any KB or social media: names, descriptions, and types (person vs organisation) for KB entities; name, username, description, and profile metrics (e.g., the "verified" flag) for profiles.

To train and evaluate our approach we collected a gold standard dataset of 35,149 alignments for persons and organisations in DBpedia that are already linked to Twitter. A baseline leveraging the Twitter API for user profile search was implemented. Our approach outperforms the baseline in terms of F1 (0.644 vs 0.604) and precision (0.849 vs 0.718), with comparable recall levels (0.519 vs 0.521).

SAC 2017, April 03 - 07, 2017, Marrakech, Morocco

ISBN 978-1-4503-4486-9/17/04…$15.00

DOI: http://dx.doi.org/10.1145/3019612.3019645

[1] http://wiki.dbpedia.org/
[2] http://www.wikidata.org/

Figure 1: Aligning DBpedia entities to Twitter profiles

Using our trained approach, we have aligned 893,446 entities from English DBpedia (version 2015-10) to Twitter and we are releasing the obtained alignments, the gold standard, and all the developed code as publicly available resources on our website;[3] intermediate results such as queries and lists of candidates are included as well, so to enable third parties to easily reproduce, improve, or build on our approach.

By producing the aforementioned alignments, we are effectively creating a bridge between Twitter and DBpedia. Twitter is currently one of the most popular social networks. The way it is designed encourages its users to share quick short messages that reflect current events. Profiles in this network are typically public, messages contain a decent amount of machine-readable metadata including external links, hashtags, mentions of people, media and location data. Various data can be inferred from this vast array of information and potentially be brought back to DBpedia. For example, a significant amount of related images, which are very sparse in DBpedia, can be automatically introduced for many entities, or new types of data can be added, such as citations, for a particular person.[4] Our alignments also directly enable various marketing activities. Using a simple SPARQL query, companies can already acquire a list of potential competitors from DBpedia. Now, using our resource, they can start automatically tracking their activities on Twitter, following the social graph to discover influencing users or potential customers. Similarly, one can easily aggregate the followers of the companies in a particular domain to infer their interests and their opinions about those companies, including weaknesses and strengths of their products and services. In sports, companies can track the social impact of teams and individual athletes to efficiently allocate their marketing budgets among them.

The remainder of the paper is organized as follows. Section 2 discusses the addressed alignment problem in more details. Section 3 presents our alignment approach, while Section 4 reports on its evaluation on the gold standard. The generated alignment resource is presented in Section 5. Section 6 reports on related works while Section 7 concludes.

2. PROBLEM DEFINITION

Our goal is to find a profile of an entity (person or organisation) in a particular social network given the knowledge base (KB) entry about this entity.[5] The KB entry is defined as a set of attributes describing the entity. We consider KBs and social networks in general in our approach, although we later instantiate it for DBpedia and Twitter.

Table 1: Information in DBpedia 2015-10, our resource, and the gold standard, including avg. # names and description length (chars) per entity (types and names always available).

	Entities (per,org)	Persons share	Have desc.	Temp. info	Avg. names	Avg. desc.
DBpedia	1636373	83.38%	93.69%	84.40%	2.53	694
Resource	893446	78.86%	95.52%	77.42%	2.54	663
Gold standard	35149	78.02%	99.97%	92.16%	2.86	712

The information available in a KB entry depends on the KB considered and, within the same KB, may be different from entity to entity. DBpedia itself, although based on Wikipedia that is being updated by millions of people every day, can have various issues including inconsistency, noisiness, obsolete knowledge and unavailability of entity attributes. An entry about the President of the United States can, for example, contain a vast amount of attributes from different domains, while an entry for a regional-level politician in a non-English speaking country can basically contain a name, a description and the occupational class. This inconsistency requires us to develop an approach that works with the bare minimum of information known about the target entity. In this paper, we assume that a KB entry at least contain the name and a textual description, person vs organisation type information, and provides some temporal information allowing distinguishing alive/existing entities from non-existing ones. Table 1 shows the information available when using DBpedia as KB,[6] covering (i) all the person and organisation entities in DBpedia; (ii) the entities included in our alignment resource; and (iii) the entities already aligned to Twitter in DBpedia[7] that form our gold standard.

Working with social media from the outside also imposes a number of challenges. First of all, it is generally not feasible to crawl the entire social network due to API limitations and its typically enormous size. Given a target KB entity, a list of candidate profiles can be obtained by querying the social network API. There is no guarantee that this list is complete or contains the right answer. We also cannot be

[3] http://alignments.futuro.media/

[4] In our resource we only release the identifiers of aligned profiles and our resource does not contain any other data gathered from social media to respect users' privacy.

[5] We start from KB entries as they are entirely known in advance, differently from social network profiles that can be only queried or (partially) acquired via expensive crawling.

[6] We get names from properties `foaf:name` and `rdfs:label`; descriptions from `dbo:abstract` and `rdfs:comment`; types from classes `dbo:Person` and `dbo:Organisation`; time information from temporal properties like `dbo:deathDate`, `dbo:deathYear`, `dbo:extinctionYear`, `dbo:extinctionDate`, `dbo:closingYear`, `dbo:closed`, `wikidata:P570`, `wikidata:P20`, `wikidata:P509`, or properties implying death like `dbo:deathPlace`, `dbo:deathCause`, `dbo:causeOfDeath`.

[7] Alignments are provided by properties `foaf:isPrimaryTopicOf` (linking to Twitter website) and `wikidata:P2002`.

Table 2: Example of queries built by strategies for an entity.

URI	`http://dbpedia.org/resource/Max_Meyer_(footballer)`
Names	`Meyer`, `Max`, `Max Meyer`, `Maximilian Meyer`

Strategy	**Constructed query**
S_{all}	(Max Meyer) OR (Maximilian Meyer)
S_{base}	Max Meyer
S_{quotes}	"Max Meyer"
S_{topic}	Max Meyer footballer

sure that the entity even has a profile in a particular social network. Because of that, we have to operate under the open world assumption: instead of just selecting the most probable answer in the list of candidates, we need to make sure that this answer is correct and reject it otherwise.

Secondly, profiles can be private, have limited attributes available, and/or contain confusing or inaccurate information. Our approach has to use the attributes that are typically available in social media. They include, for example, user name, size of his/her social graph, posting behaviour, textual description and a special "verified" flag issued by the social media that certifies the identity of the profile owner.

Third, for famous people and organisations, there typically exist impersonating and fan accounts that can be very similar to the real one. Since most of the entities do not try to acquire the "verified" flag, it can be hard even for a human to distinguish them. Moreover, certain groups of people, e.g., politicians and athletes, tend to have multiple accounts that correspond to various periods in their life. A politician might create a new account if he was elected, an athlete can do the same when changing teams. Some famous people tend to have an official and a personal account.

Another difficulty arises from the request limitations that are typically present in social media and search engine APIs. In order to find candidates for each KB entity we have to use the request quota available for us sparingly, which limits the amount of information that we can acquire for each candidate. In our particular use case, we can reasonably perform one Twitter API search request per entity, which provides us with the general candidate profile and a single tweet.

The task that we solve in this paper is similar to the well-known problem of profile matching on social media, if we look at a KB entry as a special kind of profile. However, KBs do not contain attributes that were vital to matching profiles in previous studies, such as usernames, user-generated content, and social graph. Therefore, the techniques outlined in such studies cannot be directly applied in our case and cannot provide a baseline for evaluating our approach.

3. METHODOLOGY

In order to align KB entities to social media profiles we have developed a system that: (i) acquires a set of candidate profiles for each entity; (ii) constructs a feature vector for each entity-candidate pair; and (iii) uses a machine learning-based approach to score each pair and choose the appropriate result. Our approach is scalable and uses attributes likely to be available in any KB and social network, although in the following we focus on DBpedia and Twitter.

3.1 Candidate Acquisition

Since we cannot observe the whole social network, we have to query the social media API to retrieve possible candidate profiles for each target KB entity evaluated by our approach.

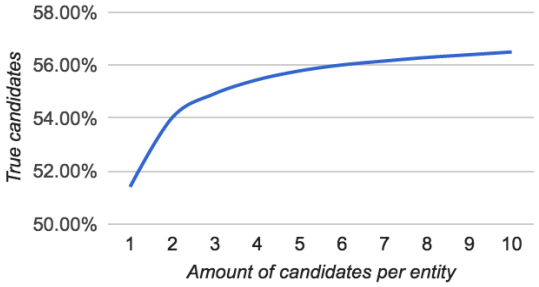

Figure 2: True candidates coverage by amount of candidates.

The choice of the search query provided to the API is significant: we want to maximise the probability of finding a set of candidates that includes the right one, without introducing too much noise. We have developed and tested four different strategies to construct the query, which aim at balancing completeness and noisiness of results:

All names (S_{all}) Known names and aliases for the target entity are found in the KB. For DBpedia, we use property `foaf:name`. Names are normalized, filtering out duplicates and names that are too short or too long. Remaining names are concatenated into a query.

Strict (S_{base}) The same procedure as before is used to construct the list of names, but only the name most frequently used to refer to the entity is selected and used as a query. For DBpedia, we choose this name as the one most frequently used in different DBpedia languages and name-related properties. This transfers well to the names that are observed in social media.

Strict with quotes (S_{quotes}) Like S_{base}, but quotes are added from both sides so that only exact matches of the name are returned by the social network API.

Strict with topic (S_{topic}) As S_{base}, but ambiguous names associated to multiple KB entities are augmented with a disambiguating topic extracted from the KB entry. For DBpedia, we use the occupational class or title encoded in the entity `rdfs:label` (e.g., "fooballer").

Examples for all the strategies are shown in Table 2. We did not add any additional information into the queries because our experiments showed that social media APIs, like Twitter one, do not handle complex queries well (they mainly target the lookup of profiles by name, rather than general-purpose profile search). More sophisticated candidate acquisition strategies can be studied. For example, we could make multiple requests cycling through all known names of an entity. However, in order to process almost a million entities for our resource in a reasonable amount of time,[8] we have limited the amount of queries spent per entity to one.

In the end, we save the top k results returned by the API as candidates for the entity, trying both not to miss the right answer and introduce less noise in the candidate selection algorithm. For Twitter, we evaluated each threshold value against our gold standard, plotting the amount of queries having a correct answer among their candidates in Figure 2. As a result, we set $k = 10$ as a reasonable threshold.

3.2 Feature Extraction

Given a KB target entity, a feature vector is extracted for each candidate acquired from the social network. Features

[8]The whole run takes 52 days with one Twitter account.

are derived from the entity-candidate pair or from the entity or candidate only. Feature vectors are scaled to unit variance and zero mean and all unique combinations of every two features are added. We consider different feature types:

Names (`NAME`) Just like in query assembly strategies, we construct the filtered list of names from the KB that is then compared to the name and the username in the candidate profile. The Jaro-Winkler and Levenshtein distances are used as similarity metrics. The social media API most likely returns candidates exactly matching the queried name, but this is not guaranteed. Such features help discarding candidates with a large edit distance from the target entity. They also allow taking into account the username of the person, which is known to be useful when aligning profiles [8, 7].

Descriptions (`DESC`) Another reliable attribute, that a KB entity almost always contains, is a free text description of the entity. On social media, users can describe themselves by providing a short description. For Twitter, we combine it with the text of the last status written by a user and his location attribute. All these data are tokenised and transformed to vector representation as a bag of words. Then sparse vectors are constructed using TF/IDF (IDF computed on Wikipedia corpus) and the cosine similarity is calculated between a KB description and all the combinations of attributes from Twitter, producing a set of similarity features.

Core profile metrics (`PROF`) We construct various features based on the amount of posts, friends, followers, and other profile statistics. Those features measure how popular a particular candidate is and how active he or she is. They capture the intuition that the right candidate profile for a KB entity is often the most popular and/or active one (this is especially the case for famous KB entities). If a user has been "verified" by the social media, such information is also included. Even though the percentage of verified entities is rather low, it is still one of the most effective features to help distinguishing the real profile from a fake one.

Wikipedia-specific features (`WIKI`) Similarly to the core profile metrics, we add certain "popularity" metrics for the KB entity itself. These features rely on a grounding of KB entities to Wikipedia pages (as occurs for DBpedia). For each entity, we consider these features: Wikipedia page length; average page visits per time unit; indegree and outdegree computed based on Wikipedia links among pages.

Entity type (`TYPE`) Rules that work well when aligning people do not necessarily work well with other entity types. To help the system distinguishing between such rule sets we map each entity to a top-level type: Person or Organisation.

Homepage links (`LINK`) Entities in a KB may be linked to external homepages, like DBpedia entities with `foaf:homepage`. We crawl entities homepages and scrape links to their Twitter profiles. Extracted profiles are then searched to find links that go back to the website from which they were extracted. From this information three binary features are constructed: (i) if a candidate profile is contained in the list of profiles scraped for the target entity; (ii) if this candidate profile is the only one extracted; and (iii) if there is a link back to the website from which this candidate was extracted.

3.3 Candidate selection

To score the candidates we formulate a classification problem where the classifier has to provide a probability of a candidate profile being a match of a target entity. We train two supervised models on the DBpedia gold standard described earlier: `SVM` and `DNN`. `SVM` is a simple SVM-based model with a linear kernel that returns probabilities as a result. `DNN` is a deep neural network that uses stacked densely-connected layers with *tanh* as activation function. On each layer random dropout was applied to prevent overfitting. The $softmax$ function is then applied on top to acquire probabilities. Cross-entropy is employed as a cost function and the *Adagrad* algorithm is used to train the network.

After acquiring each candidate score, the most probable candidate is selected. The system abstains from selecting the candidate if its probability or the difference in probabilities from the runner-up candidate is below two predefined thresholds, which can be tuned to balance precision and recall depending on the desired properties of the final system.

Table 3: Query assembly strategies vs gold standard.

Strategy	Average candidates	Has some candidate	Has true candidate
S_{base}	6.86	88.96%	**56.50%**
S_{topic}	6.34	85.90%	53.22%
S_{quotes}	6.32	85.54%	53.98%
All names	3.46	50.15%	29.64%

Table 4: Precision, recall, and F1 of candidate selection.

	Persons			Organisations			All entities		
	P	R	F1	P	R	F1	P	R	F1
Base	0.732	**0.921**	0.816	0.679	**0.889**	0.770	0.718	**0.916**	0.805
SVM	**0.866**	0.891	0.878	0.851	0.862	0.856	**0.860**	0.885	0.872
DNN	0.858	0.905	**0.881**	**0.867**	0.863	**0.865**	0.854	0.900	**0.876**

4. EVALUATION

We evaluate our approach on the DBpedia gold standard, since there are no publicly available datasets for our task.

As baseline, we consider a straightforward approach that aligns a DBpedia entity to the top ranked Twitter profile (if any) returned by Twitter when queried with the entity name (S_{base} query assembly strategy). This baseline simulates a user's search for profiles on Twitter, and leverages the proprietary search algorithm developed by Twitter for this purpose. Since this baseline never abstains from making an alignment decision (when there is at least a profile matching the search) it tends to favour recall over precision.

We first consider candidate acquisition and candidate selection separately, and then assess the performances of the overall approach and the impact of different feature types.

Candidate Acquisition We evaluate the query assembly strategies against the whole gold standard dataset, considering as performance metric the percentage of entities having the correct candidate in the profiles returned by a strategy.

Table 3 reports the results (best value in bold) and also lists the percentage of entities having some candidate and, for those entities, the average number of candidates returned. The table shows that the more sophisticated the query gets, the less candidates are returned and the less entities have the correct candidate. This is due to the nature of the Twitter API search algorithm: it does a great job matching real names, usernames, and hashtags if there is an exact match, but performances drop significantly once the query gets more complex. Therefore, hereafter we use the S_{base} strategy.

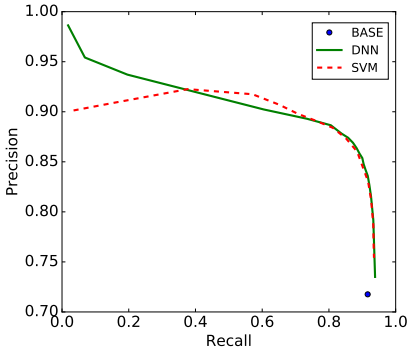

Figure 3: P/R curves for candidate selection – DNN vs SVM vs Baseline.

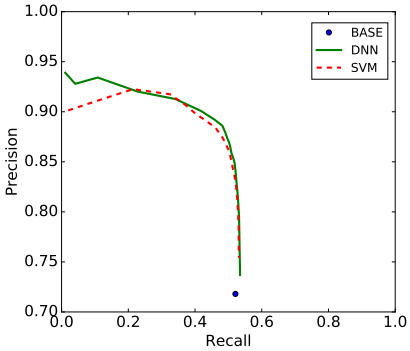

Figure 4: P/R curves for overall system – DNN vs SVM vs Baseline.

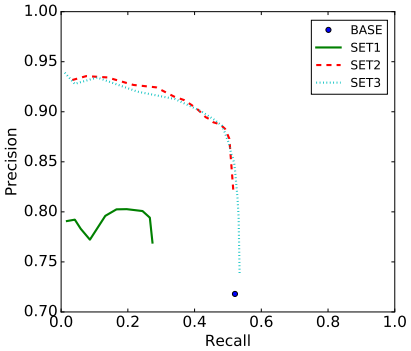

Figure 5: P/R curves for overall system – DNN with various feature sets.

Table 5: Precision, recall, and F1 of overall system.

	Persons			Organisations			All entities		
	P	R	F1	P	R	F1	P	R	F1
Base	0.732	**0.536**	0.619	0.679	0.510	0.582	0.718	0.521	0.604
SVM	0.837	**0.536**	**0.653**	0.821	**0.519**	**0.636**	0.830	**0.522**	0.641
DNN	**0.845**	0.532	**0.653**	**0.849**	0.505	0.633	**0.849**	0.519	**0.644**

Table 6: Precision, recall, and F1 using different features.

	Persons			Organisations			All entities		
	P	R	F1	P	R	F1	P	R	F1
SET1	0.810	0.260	0.393	0.818	0.369	0.509	0.798	0.269	0.402
SET2	**0.883**	0.518	**0.653**	**0.873**	0.494	0.630	**0.873**	0.504	0.639
SET3	0.845	**0.532**	**0.653**	0.849	**0.505**	**0.633**	0.849	**0.519**	**0.644**

Candidate Selection We now consider the entities for which a non-empty set of candidates is returned by S_{base}, and measure the performances of the SVM and DNN candidate selection models in finding the correct candidate among that set (if any).[9] We use 80% of the dataset for training SVM and DNN with different precision/recall balances, and then we test them on the remaining 20% test set.

Table 4 shows precision (P), recall (R), and F1 of each model (best values in bold) when tuned for maximum F1 over all entities, also listing performances obtained by those models for persons and organisations separately; Figure 3 shows the precision/recall curves of SVM and DNN over all entities. Both SVM and DNN outperform the baseline significantly in terms of F1 and precision, with comparable recall levels. Persons are classified marginally better and SVM and DNN exhibit similar performances, although DNN manages to reach better precision at low recall levels.

Overall approach We evaluate our overall approach on the 20% test set, using the S_{base} candidate acquisition strategy and the previously trained SVM and DNN models, this time considering all the entities and not just the ones for which candidate acquisition gives some results.

Table 5 reports precision, recall, and F1 on the test set when tuning the system for maximum F1 (best values in bold), while Figure 4 shows the precision/recall curves of our approach using SVM or DNN over all entities. While the system still performs better than the baseline (0.644 vs 0.605 F1), we observe a sensible loss of recall with respect to the results obtained for candidate selection alone, caused by an incomplete candidate set returned by the candidate acquisition part (see Table 3). So even if the correct profile exists (as in the case of the gold standard), we are not guaranteed to find it in a candidate list received from Twitter. Again, persons are aligned marginally better than organisations, and the SVM and DNN models perform comparably.

[9]There is a *false negative* for every true candidate not found by the system, and a *false positive* for every wrongly selected candidate; correctly selected candidates are *true positives*.

Table 7: Resource statistics.

	Persons	Org.	All
Entities in DBpedia	721,769	171,677	893,446
Entities with candidates	534,902	95,865	630,767
Avg. candidates / entity	6.98	6.63	6.93
High-quality alignments	136,941	32,807	169,748

Feature analysis We investigate the impact of different sets of features on the performances of the overall approach, employing the S_{base} strategy along with the (slightly) best performing `DNN` model described previously. We devised three feature sets (see Section 3.2 for feature types): `SET1` including just `NAME`; `SET2` including `NAME`, `DESC`, `TYPE`, `PROF` and `LINK` (all but `WIKI`); `SET3` including all available feature types. Table 6 and Figure 5 report the results of those feature sets. `SET1` performs poorly in terms of F1 and recall, both below the baseline, but still achieves a decent precision. `SET2` shows results close to the best and `SET3` offers a very marginal improvement on top of `SET2`, implying that the `WIKI` feature type available only for Wikipedia-grounded KBs (like DBpedia) is not essential to our approach, which can be thus applied effectively to other KBs.

5. RESOURCE

Using the proposed approach, we have built an *alignment resource* linking DBpedia entities to their corresponding Twitter profiles. The resource is publicly available on our website in multiple formats (RDF, CSV, JSON) together with confidence scores, intermediate results (e.g., candidates) and the code needed to build it. As mentioned in Section 3, we extracted from DBpedia 2015-10 a set of 893,446 entities that are either living persons or existing organisations. The S_{base} strategy was used to query the Twitter API, obtaining candidates for 630,767 entities. The DNN candidate selection model was applied to derive "high-quality" alignments for 169,748 entities, using the candidate selection thresholds giving the highest F1. For the remaining

entities, the top alignments (not satisfying the thresholds) were also included together with their scores, as they may still be useful in applications favouring recall over precision. Relevant statistics are provided in Table 7. Our approach requires a single request to the KB and Twitter API. Given that a KB can be installed on premises, scalability is only limited by the Twitter API `users/search` method quota.

6. RELATED WORK

To the best of our knowledge, no one has investigated the task of automatically matching knowledge base entries to social media accounts. This task, however, is closely related to the profile matching problem. Profile matching (or profile aligning) is the task of aligning multiple profiles of the same person in multiple social media. A lot of research has been done for this task, exploiting various attributes in profiles [6, 3, 5], user-generated content [6, 4, 2], and social graphs [5].

Some researchers have pointed out that most of the attributes that could theoretically be exposed in social media are unreliable for profile matching [3]. Attributes might not exist, might contain information of varying granularity, or they might even be false. Attempts were therefore made to use as little information as possible to align profiles (as we do in this work), choosing only the most reliable attributes.

In studies by Zafarani et al. [8, 7] only the username (which is the unique identifier that exists in virtually any social media) was exploited to align profiles. The authors showed that people tend to be very consistent when choosing their usernames, which enable identification even if the rest of the profile is filled with incorrect information. Even though they proved that username is a powerful feature, knowledge bases typically do not contain examples of usernames, which makes this feature unavailable for our task.

Goga et al. [3] explored the reliability of attributes in various social media. According to their study, only few attributes like username and real name are available in social media reliably. For example, they reported that location is present in 54% of Twitter profiles and is not consistent across multiple social media. They exposed various methodological and technical challenges in this area related to the construction of ground truth datasets, attribute discriminability and impersonability. The PhD thesis of Goga [1] provides a more in-depth look into those issues.

Liu et al. [4] reported the largest experiment on profile matching to date using a dataset of 10 millions profiles across 7 social media. Their approach leverages a wide variety of hand-crafted features based on textual and image user-generated content, and demonstrates its importance for profile matching. Note that user-generated content is usually missing in KBs, so it cannot be used in our task as it is used in the profile alignment task.

Goga et al. [2] showed that profiles can be matched robustly even if explicit attributes are hidden or intentionally falsified. Their approach uses only implicit features present in social media but generally unavailable in KBs, such as writing style, messaging behaviour, and location metadata.

Social graph has proven to be hard to acquire in social media. It could be unavailable for crawling or there could be very strict restrictions on API (most notably, in Twitter). Lu et al. [5] were able to gather a small social graph dataset and proved that it can be effectively matched to improve the results of profile alignment. Entities in KBs often contain links to other entities which can be interpreted as a kind of social graph. In this paper we do not use such feature, but in future studies it could be proven useful.

Finally, Peled et al. [6] gave an overview of the profile alignment task and presented their own approach that uses all available information in the profile to perform matching. They presented three main use cases for their system, one of which—searching for a user by similar name—is close to the candidate acquisition part of our system.

To summarize, even though some researchers expressed concerns [3, 1] regarding the usage of some attributes, every piece of profile data contribute towards identifying the user.

7. CONCLUSION AND FUTURE WORK

In this paper we presented a supervised approach for automatically matching a KB entity to the corresponding social media profile, using only a bare minimum of data from the social media that allows scaling to large amounts of entities.

We applied the approach to build a publicly available resource containing alignments for 893,446 DBpedia entities linked to corresponding Twitter profiles. This resource can be used for various social media related activities, such as entity linking in tweets and user profiling, and effectively acts as a bridge connecting Twitter to the LOD cloud. We also released a gold standard dataset of 35,149 alignments to act as a benchmark for further studies in this area.

Our approach can be further improved by using more data from the candidate profile and more attributes from the KB. For example, we can try to infer the occupational class from the messages or exploit the social graph. However, the more data is used the more difficult it is to scale the approach due to the access limitations of social media APIs.

Similar approaches can be applied to perform the same task for other social networks, truly connecting the social media world to the LOD cloud.

8. REFERENCES

[1] O. Goga. *Matching user accounts across online social networks: methods and applications*. PhD thesis, LIP6-Laboratoire d'Informatique de Paris 6, 2014.

[2] O. Goga, H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira. Exploiting innocuous activity for correlating users across sites. In *Proc. of WWW*, pages 447–458. ACM, 2013.

[3] O. Goga, P. Loiseau, R. Sommer, R. Teixeira, and K. P. Gummadi. On the reliability of profile matching across large online social networks. In *Proc. of KDD*, pages 1799–1808. ACM, 2015.

[4] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan. HYDRA: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proc. of SIGMOD*, pages 51–62. ACM, 2014.

[5] C.-T. Lu, H.-H. Shuai, and P. S. Yu. Identifying your customers in social networks. In *Proc. of CIKM*, pages 391–400. ACM, 2014.

[6] O. Peled, Fire, Rokach, and Elovici. Matching entities across online social networks. *Neurocomputing*, 2016.

[7] R. Zafarani and H. Liu. Connecting corresponding identities across communities. In *Proc. of ICWSM*. AAAI Press, 2009.

[8] R. Zafarani and H. Liu. Connecting users across social media sites: A behavioral-modeling approach. In *Proc. of KDD*, pages 41–49. ACM, 2013.