Inferring Web Citations using Social Data and SPARQL Rules

Matthew Rowe

OAK Group, Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello Street, S1 4DP Sheffield , United Kingdom
m.rowe@dcs.shef.ac.uk

Abstract. Web users who disseminate their personal information risk falling victims to malevolent web practices such as lateral surveillance and identity theft. To avoid such practices, web users must manually search for web resources which *may* cite them and then perform analyses to decide which resources *do* cite them; a time-consuming and frequently repeated practice. This paper presents a automated rule-based technique to identify web citations, intended to alleviate this manual process. Seed data is leveraged from user profiles on various Social Web platforms and is then used as seed data from which SPARQL rules are constructed and applied in infer web citations. An evaluation of this technique against humans performing the same task shows higher levels of precision.

1 Introduction

A large proportion of Web platforms now encourage web users to become involved and participate online through the provision of interactive functionalities. Blogging platforms such as Live Journal allow web users who lack the technical expertise to build their own web page to publish their thoughts and musings in an online space and news services such as the BBC now allow web users to become involved by sharing content and being involved in topical discussions. The increased involvement of web users has lead to a reduction in online privacy and an increase in the publication of personal information on the World Wide Web. The sensitive nature of this information has in turn lead to a rise in malevolent web practices such as lateral surveillance [1] and identity theft - which currently costs the UK economy £1.2 per annum[1]. To avoid falling victim to such practices web users are forced to manually search for web resources (i.e. web pages) which *may* cite them and identify which web resources *do* refer to them.

Manually performing the task of identifying web citations is time consuming, laborious and information intensive - given the ever increasing amount of information which is published on the Web. Furthermore the rate at which the Web grows requires this process to be repeated frequently so that new web citations can be identified and the correct action taken (i.e. removing sensitive information). This presents a clear motivation for the application of automated

[1] http://www.identitytheft.org.uk/faqs.asp

techniques to identify web citations. To be effective, such techniques must be provided with *seed data* which describes the identity of the person whose web citations are to be found. However producing this seed data manually is expensive - given the rich identity descriptions which are required to enhance the ability of the technique to detect web citations.

This paper presents an approach to identify web citations using a rule-based technique. Social graphs are exported from user profiles built on individual Social Web platforms using the Resource Description Framework (RDF) and the Friend of a Friend (FOAF) [2] and GeoNames[3] ontologies to describe the semantics of the exported data. Seed data is built by combining several social graphs together, from distributed Social Web platforms, into a single RDF model. SPARQL rules [6] are then constructed from RDF instances within the seed data using a technique inspired by existing general-to-specific rule induction techniques from the state of the art. Web resources are gathered and the SPARQL rules are applied to the resources in order to infer web citations by matching information within the seed data - which describes a person's identity on the Social Web - with a given web resource. One of the governing intuitions behind this approach is that a given person will appear in web resources with other individuals that he/she knows - this is reflected in similar identity disambiguation literature [4, 5]. User profiles from Social Web platforms provide the necessary data to facilitate support the application of this intuition. We present a thorough evaluation of this technique and compare its performance against several baseline measures including human processing.

We have structured the paper as follows: section 2 discusses related work, explaining current rule induction techniques and rule-based classification approaches. Section 3 presents the approach to detecting web citations by describing the generation of seed data, the construction of SPARQL rules and their application. Section 4 presents an evaluation of the technique against humans performing the same task. Section 5 describes the conclusions and plans for future work within this area.

2 Related Work

Identifying web citations has been attempted in [2] using background knowledge provided as a list of names representing the social network of a person, posssible web citations are then gathered by searching the Web and clustered based on their link structures as either *citing* the person or *not*. Similar work by [12] gathers and labels web pages as either containing an entity reference or not, seed data is provided as labelled training examples from which a machine learning classifier is learnt and applied to unlabelled web pages. Several commercial services have been proposed to identify web citations in order to tackle identity. For example Garlik's Data Patrol[4] service searches across various data sources for personal

[2] http://xmlns.com/foaf/0.1/
[3] http://www.geonames.org/ontology/
[4] http://www.garlik.com/dpindividuals.php

information about the individual and uses information when the user signs up to the service to aid the process (includes biographical information such as the name and address of the person). The Trackur[5] service monitors social media on the Web for references and citations with the intended goal of monitoring the reputation of an individual. Similar to Data Patrol, Trackur requires background information about the person whose reputation is to be monitored.

Rules provide a systematic means to infer a web citation and are built using two distinct types of induction technique: *general-to-specific* and *specific-to-general*. The former use a *sequential covering* strategy by gradually specialising a general skeleton rule resulting in coverage of the example set for a given class label. For instance the FOIL [8] algorithm constructs a general rule to match examples from a given set and adds literals which are most *gainful* to the antecedent of the rule thus specialising it. Specific-to-general rule induction algorithms employ a *simultaneous covering* strategy by constructing many rules simultaneously from the examples and then generalising those rules. The C4.5 algorithm [7] creates a decision tree from the labelled examples, rules are then built from paths within the tree. As the produced rules overfit the examples, C4.5 performs post-pruning to generalise the rules by reducing literals.

3 SPARQL Rules: Identifying Web Citations

Identifying web citations of a given person involves matching known information which describes the person's identity which information present within web resources. Our approach to identifying web citations uses the intuition that a person will appear in web resources with other people he/she knows (e.g. work pages, electoral rolls) by gathering seed data which describes both biographical and social network information from the Social Web. Should information appear in a web resource which is related to the given individual - via the seed data - then this information is matched and a web citation is inferred. Fig. 1 presents an overview of the approach which is divided into three stages: first seed data describing the individual whose web citations are to be found is gathered from user profiles on the Social Web. Second, we then query the World Wide Web and the Semantic Web using the person's name to gather possible web citations. Third, we build rules from the seed data and apply them to the web resources to infer web citations. We now explain the various stages of the approach in greater detail.

3.1 Generating Seed Data

We build seed data by exporting individual social graphs from several disparate Social Web platforms - Facebook[6], Twitter[7] and MySpace[8]. An exported social

[5] http://www.trackur.com/
[6] http://www.facebook.com
[7] http://www.twitter.com
[8] http://www.myspace.com

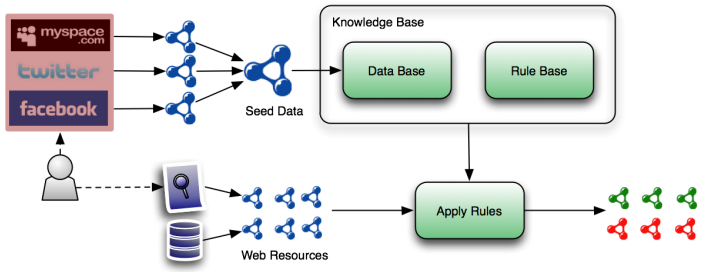

Fig. 1. An approach to build and apply SPARQL rules to identify web citations

graph defines the profile of the user which is visible within a given platform. The digital identity of a web user is fragmented across the Social Web between different platforms, therefore, by exporting various user profiles, we are able to capture a more complete identity representation of the person. Social Web platforms allow access to data through APIs and expose data in proprietary formats (i.e. XML responses using an XML schema unique to the platform). We export social graphs as RDF models using concepts from the FOAF ontology to describe biographical and social network information and concepts from the Geonames ontology to define location information, thereby providing a consistent interpretation of identity information from disparate profile sources.

Social graphs are then linked together to form a single RDF model which is to be used as seed data. Interlinking is performed using reasoning over the available information. We use handcrafted rules which compare instances of *foaf:Person* in disparate graphs and infer a match based on the available data (e.g. same homepage, same location). This provides a more detailed social graph from which SPARQL rules can be built. A full explanation of this technique falls outside of the scope of this paper however, instead we refer the reader to [9]. As Fig. 1 shows the compilation of seed data is finalised once the single social graph is built. This graph contains the features from which SPARQL rules are constructed, a snippet of which would look as follows (using n3 syntax):

```
<http://example.com/foaf.rdf#me>
        rdf:type foaf:Person ;
        foaf:name "Matthew Rowe" ;
        foaf:homepage <www.dcs.shef.ac.uk/~mrowe> ;
        foaf:mbox <m.rowe@dcs.shef.ac.uk> ;
        foaf:knows <http://example.com/foaf.rdf#fabio> ;
        foaf:knows <http://example.com/foaf.rdf#sam> .
<http://example.com/foaf.rdf#fabio>
        rdf:type foaf:Person ;
        foaf:name "Fabio Ciravegna";
        foaf:mbox <fabio@dcs.shef.ac.uk>;
        foaf:homepage <http://www.dcs.shef.ac.uk/~fabio> .
<http://example.com/foaf.rdf#sam>
        rdf:type foaf:Person ;
        foaf:name "Sam Chapman" ;
        foaf:mbox <sam@dcs.shef.ac.uk> ;
        foaf:homepage <http://www.dcs.shef.ac.uk/~sam> .
```

3.2 Gathering Possible Web Citations

To identity web citations for a given person we must gather a set of possible citations so that our rules can then be used to identify which cite the person. We search the World Wide Web using Google and Yahoo and the Semantic Web using Watson[9] and Sindice[10] by querying for the person's name whose web citations are to be found. The results from the queries are then gathered together provide a collection of web resources which may cite the person. In order to apply SPARQL rules we create RDF models describing the knowledge structure of the web resources - this allows triple patterns to be matched which associate information within the social graph to information within the web resource. For XHTML documents containing lightweight semantics such as RDFa we use Gleaning Resource Descriptions from Dialects of Language (GRDDL)[11] to apply XSL Transformations to the documents, thereby gleaning an RDF model. An example gleaned model from the OAK Group member page[12] looks as follows:

```
<http://oak.dcs.shef.ac.uk/people>
        foaf:topic <http://oak.dcs.shef.ac.uk/people#fabio> ;
        foaf:topic <http://oak.dcs.shef.ac.uk/people#matt> ;
        foaf:topic <http://oak.dcs.shef.ac.uk/people#sam> .
<http://oak.dcs.shef.ac.uk/people#fabio>
        rdf:type foaf:Person ;
        foaf:name "Fabio Ciravegna";
        foaf:homepage <http://www.dcs.shef.ac.uk/~fabio> .
<http://oak.dcs.shef.ac.uk/people#matt>
        rdf:type foaf:Person ;
        foaf:name "Matthew Rowe" ;
        foaf:homepage <www.dcs.shef.ac.uk/~mrowe> .
<http://oak.dcs.shef.ac.uk/people#sam>
        rdf:type foaf:Person ;
        foaf:name "Sam Chapman" ;
        foaf:homepage <http://www.dcs.shef.ac.uk/~sam> .
```

For HTML documents we build RDF models from person features by using DOM manipulation to identify context windows and then extracting person information from those windows. A single context window contains information about a single person: his/her name together with or without an email, web address, location. We extract these attributes from the window using Hidden Markov Models trained for the task, and then build an RDF model of the web resource containing these features using the same ontologies as the seed data (i.e. creating an instance of *foaf:Person*). An extensive discussion of this technique falls outside the scope of the paper, instead we refer the reader to [10]. Metadata models returned from querying the Semantic Web are left in tact, given that machine-readable descriptions are already provided.

3.3 Inferring Web Citations using SPARQL Rules

Seed data is provided in the form of a social graph, defined using RDF, describing both the biographical and social network information of a given person. This

[9] http://kmi-web05.open.ac.uk/WatsonWUI/
[10] http://www.sindice.com|
[11] http://www.w3.org/TR/grddl/
[12] http://oak.dcs.shef.ac.uk/people

single RDF model provides the solitary example from which rules can be constructed - given that this is the only information which is known to describe the person whose web citations are to be identified. This limits the ability of state of the art rule induction techniques such as FOIL and C4.5 to function effectively, given their reliance on sufficiently large example sets. Rather than relying on a large example set, rules are instead constructed from RDF instances within the social graph.

RDF Instance Extraction An RDF instance represents a resource within a given RDF model which can either be an anonymous node or identified by a URI. An instance is a unique object, which in the case of the seed data can be a social network member - identified as an instance of *foaf:Person* - or a location related to a given person - identified as an instance of *geo:Feature*. Such instances form a useful basis for inferring a web resource as referring to an individual, given that if information describing the instance is also found within a web resource then the web resource can be identified as citing the person. To leverage RDF instances from a given RDF model we use a *Resource Leaves* (Equation (1)) construct which selects all the triples (<*r,p,o*>) attributed to a given resource (r) from a given RDF model/graph (G) where the object (o) of each triple does not act as the subject of other triples (<*o,p',o'*>). This returns a set of resources and literals which form leaves surrounding the resource (r) such that no paths are beyond those leaves - these in turn provide the features from which rules are built.

$$RLS_G(r) = \{< r, p, o > \mid < r, p, o > \in G \wedge \nexists p', o' < o, p', o' > \in G\} \quad (1)$$

Building SPARQL Rules SPARQL provides a mechanism for querying RDF models by matching graph patterns. SPARQL rules allow a given graph to be derived - denoted by the triples within the `CONSTRUCT` clause of the rule - by matching triples within the `WHERE` clause of the rule. SPARQL rules are built using a general-to-specific strategy inspired by FOIL [8] and only a single positive example - the social graph - to construct general skeleton rules which are then specialised. The strategy for building rules works according to Algorithm 1, which will now be explained.

Each resource is extracted from the supplied social graph (seed data) (line 2). A skeleton SPARQL rule is created for each resource (line 3) comprised of the name of the social graph owner together with triples identifying a person within a web resource with the same name. The *Resource Leaves* construct (RLS_G) is then used to extract a tripleset for the resource (r) from the social graph (line 4) - this forms the information which is used to build the rules. The algorithm then goes through the tripleset and builds the rules as follows: if the resource (r) is an instance of *foaf:Person* and is not the social graph owner - and a social network member - then the triple (<*s,p,o*>) is added to the `WHERE` clause of R (line 7). Additional triples are added to relate the social graph owner with the

Algorithm 1 buildRules(r_p,G) : Induces rules from RDF instances. Input to this algorithm is the social graph (G) and the person whose web resources are to be disambiguated (r_p). Rules are induced and added to the rule base (RB).

Input: G, r_p
Output: RB

1: $RB = \emptyset$
2: **for each** resource $r \in G$ **do**
3: R = CONSTRUCT {<r_q foaf:page ?url>} WHERE {<r_p foaf:name ?n>,<?url foaf:topic ?p>,<?p foaf:name ?n>}
4: $T_r = RLS_G(r)$
5: **for each** <s,p,o> $\in T_r$ **do**
6: **if** (<r rdf:type foaf:Person>) && (r != r_p)) **then**
7: Add <r_p foaf:knows ?q>,<?q p ?o>,<?url foaf:topic ?x>,<?x p ?o> to WHERE clause of R
8: **if** (<p rdf:type owl:inverseFunctionalProperty>) **then**
9: Q = CONSTRUCT {<r_q foaf:page ?url>} WHERE {<r_p foaf:name ?n>,<?url foaf:topic ?p>,<?p foaf:name ?n>}
10: Add <r_p foaf:knows ?q>,<?q p ?o>,<?url foaf:topic ?x>,<?x p ?o> to WHERE clause of Q
11: $RB = RB \cup Q$
12: **end if**
13: **else if** ((<r rdf:type foaf:Person>) && (r == r_p)) **then**
14: Add <r_p p ?o>,<?p p ?o> to WHERE clause of R
15: **if** (<p rdf:type owl:inverseFunctionalProperty>) **then**
16: Q = CONSTRUCT {<r_q foaf:page ?url>} WHERE {<r_p foaf:name ?n>,<?url foaf:topic ?p>,<?p foaf:name ?n>}
17: Add <r_p p ?o>,<?p p ?o> to WHERE clause of Q
18: $RB = RB \cup Q$
19: **end if**
20: **else if** ((<r rdf:type geo:Feature>) && (<r_p foaf:based_near r>)) **then**
21: Add <r_p foaf:based_near ?g>,<?g p ?l>,<?p foaf:based_near ?h>,<?h p ?l> to WHERE clause of R
22: **end if**
23: **end for**
24: $RB = RB \cup R$
25: **end for**
26: **return** RB

person and relate a person within a web resource with the object of the triple. The predicate (p) of the triple (<*s,p,o*>) is also checked. If it is inverse functional (line 8) then another rule (Q) is created using the same earlier skeleton rule and the addition of triples to the WHERE clause of Q. Q is then added to the rule base RB. A predicate is said to be inverse functional when it defines a unique property of the subject (e.g. an email address of a person is unique to that person). Q resembles a rule which is specific to the resource (r) such that the discovery of the value (o) of the property (p) uniquely identifies the resource in a web resource.

If r is the social graph owner (line 13) then the triple (<*s,p,o*>) from the tripleset of r is added to the $WHERE$ clause of R (line 14) by directly relating the social graph owner with the triple. Similar to before, if the predicate of the triple defines an inverse functional property then a separate rule Q is created using the skeleton rule from before and <*s,p,o*>, and is added to RB. If the resource which is being analysed denotes a location (line 20) and the location is the same as the social graph owner's location then the details of the location are added to the rule. The cycle of populating the rule R is repeated until all the

triples of the resource r have been analysed. Essentially the strategy functions by populating the rule with all the triples such that the structure of the social graph can be utilised to match information within web resources. The rule (R) once complete is added to the rule base (line 24). Once all resources within the seed data have been analysed and rules compiled then the rule base is returned. The following section presents example rules which are created from this technique.

3.4 Example Rules

SPARQL rules are built to match triple patterns which associate known information about a given person - present within the seed data - with information within a given web resource and infer the existence of a web citation. The body of the `WHERE` clause of a given SPARQL rule contains the triples which make this association possible. An example rule constructed using Algorithm 1 matches the presence of a person's name, together with the presence of the name of a social network member.

```
CONSTRUCT { <http:/example.com/foaf.rdf#me> foaf:page ?url }
WHERE {
        <http://example.com/foaf.rdf#me> foaf:name ?n .
        <http://example.com/foaf.rdf#me> foaf:knows ?p .
        ?p foaf:name ?o .
        ?url foaf:topic ?q .
        ?url foaf:topic ?r .
        ?q foaf:name ?n .
        ?r foaf:name ?o
}
```

The graph patterns included within the body of the rules depends on the features of the seed data - i.e. the RDF instances and the triples within the instance descriptions. The advantage of using SPARQL rules is the use of variables within the rules' body, thus allowing the graph patterns to be built from different RDF instances. The intuition behind this strategy is that information within distinct RDF instances will vary such that one instance may contain only the person's name, whereas another will contain their name and email address. By using only the structure of this information both of these patterns are covered in order to provide more general rules for matching information.

Fig. 2 shows an example of how SPARQL rules can be applied to infer a web citation. In this case a SPARQL rule is built from the seed data which first matches the social graph owner's name within a given web resource, and then detects the homepage of a member of the social graph owner's social network. The rule then constructs a triple which associates the social graph owner with the URL of the web resource as a web citation.The metadata model of the web resource provides a common interpretation of information with the seed data and therefore allows the rule to match information found within the social graph.

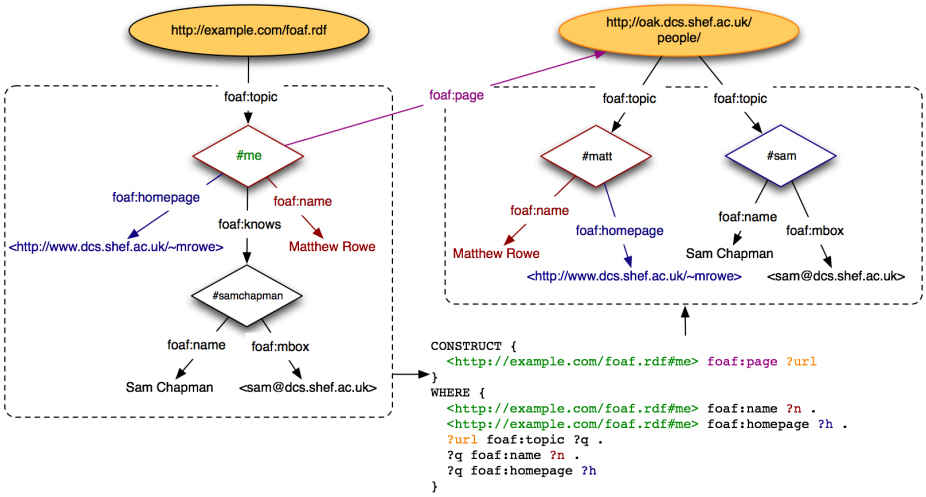

Fig. 2. Applying a SPARQL rule to the RDF model of *oak.dcs.shef.ac.uk/people* to match a social network member's homepage. Output is a triple connecting a person which a web resource which cites that person.

4 Evaluation

4.1 Evaluation Measures

SPARQL rules divide a set of possible web citations into two distinct sets; those that refer to a given person (*positive*) and those that do not (*negative*) by either matching known identity information or not. In order to evaluate the performance of the technique, accuracy is assessed using information retrieval metrics [11] featuring two sets of documents: A denotes the set of relevant web resources and B denotes the set of retrieved web resources. Precision measures the proportion of web resources which are labelled as citing a given person and are true references ($precision = |A \cap B|/|B|$). Recall measures the proportion of true references that were retrieved ($recall = |A \cap B|/|A|$). F-measure provides the harmonic mean between the two measures ($f - measure = \frac{2 \times precision \times recall}{precision + recall}$).

4.2 Dataset

The evaluation dataset was compiled using 50 randomly chosen members of the Semantic Web and Web 2.0 communities as participants. For each participant seed data was produced by generating individual social graphs from Facebook and Twitter and interlinking the graphs together. Possible web citations were gathered by querying the WWW and the Semantic Web, and gathering and caching the first 100 results from each query. Following removal of duplicate web resources the resulting dataset contained approximately 17300 web resources

with ~ 346 web resources to be analysed for each participant. Metadata models were then built to represent information found within each of these web resources using the previously described techniques. A gold standard was compiled for the evaluation by manually labeling the dataset. We assess the performance of our technique against the web presence levels of the participants which we define as the proportion of web resources retrieved for a given participant's name that refer to the participant, this is given as a percentage. (e.g. if participant A appears in 50 of the 350 web resources, then the web presence level of that participant is 14%). This allows the technique to be analysed based on its ability to handle varying degrees of web citation density.

4.3 Baseline Measures

We use two baselines for our experiments. *Baseline 1* consists of a basic SPARQL rule which matches the occurrence of a person's name within a given web resources to infer a web citation. *Baseline 2* consists of humans identifying web citations which was derived using a group of 12 raters who manually processed a portion of the dataset. Three different raters identified web citations for each evaluation participant, thereby providing three distinct sets of results, and interrater agreement was used to find the agreement of results between the raters according to techniques in [3]. This provided measures for precision and recall and therefore f-measure. The average of the three separate rater agreement measures for each participant was then computed.

4.4 Results

SPARQL rules achieve high levels of precision with respect to the two baseline techniques (as shown in Table 1). This is due to the strict nature of the rules, requiring a web resource's knowledge structure to exactly match the tripleset of the graph pattern from the `WHERE` clause of the rule. A consequence of this strict matching of knowledge structures is poor levels of recall as indicated within the results. SPARQL rules utilise a *supervised* learning technique such that only *labeled* data (seed data) is used to construct the rules, this inhibits their ability to learn features from identified web citations. With respect to the baseline measures, inference rules yields higher levels of precision but poor recall levels. In comparison to baseline 1, the results indicate that by specialising general rules using RDF instances as features then identified web citations are almost all correct - indicated by the 95% of identified web citations which were true citations. With respect to baseline 2 (human processing) rules returns more accurate web citations yet misses many citations, whereas humans detect such references.

To analyse the relationship between the performance of SPARQL rules and web presence levels, best fitting lines of regression are chosen that maximise the coefficient of determination within the data. Fig. 3 indicates that a linear relationship exists between the performance of rules and web presence levels. This suggests that accuracy levels improve as the web presence of an individual

Table 1. Accuracy of Rules with respect to Baseline techniques

	Precision	Recall	F-Measure
Rules	0.955	0.436	0.553
Baseline 1	0.191	0.998	0.294
Baseline 2	0.765	0.725	0.719

increases. Rules also outperform human processing from the second baseline for low levels of web presence. At such levels information is often sparse and can lead to a "*needle in a haystack*" problem if the web presence level of an individual is very low. As the results indicate the cognitive process of discovering these low-levelled individuals is difficult, however using an automated technique, such as Rules, offers a suitable approach to discovering sparse data with high precision levels.

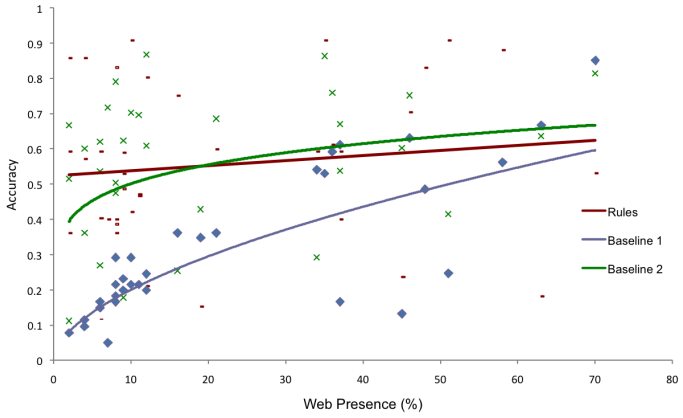

Fig. 3. Accuracy of Rules and Baseline techniques with respect to web presence levels

5 Conclusions

The work presented within this paper provides a technique for identifying web citations which yields high levels of precision. The comparison of the use of SPARQL Rules constructed from seed data against human processing indicates that humans are able to detect a larger number of web citations than rules. The combined f-measure level of rules - given its poor recall levels - achieves less accurate results than humans performing the same task. However it is worth noting that for web users with a low-level of web presence - i.e. only appear in a limited number of web resources - rules outperforms humans. We believe that this

is due to web citation sparsity making it harder for humans to spot web citations, whereas automated techniques are able to. Our technique provides a novel use of combining digital identity information from user profiles on existing Social Web platforms. The results indicate that the use of seed data from such sources is sufficient to identify web citations. One of the limitations of our technique however is its inability to learn from the identified web citations. In such cases features may be present in the web citations which are not within the seed data (i.e. work colleagues, publications), a technique which could learn from such features would yield higher levels of recall than the presented technique within this paper. We plan to explore this line of work in our future research.

References

1. M. Andrejevic. The discipline of watching: Detection, risk, and lateral surveillance. *Critical Studies in Media Communication*, 23(5):392–407, 2006.
2. R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 463–470, New York, NY, USA, 2005. ACM.
3. G. Hripcsak and A. S. Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of American Medical Informatics Association*, 12(3):296–298, 2005.
4. D. V. Kalashnikov, Z. Chen, S. Mehrotra, and R. Nuray-Turan. Web people search via connection analysis. *IEEE Transactions on Knowledge and Data Engineering*, 20(11):1550–1565, 2008.
5. L. Lloyd, V. Bhagwan, D. Gruhl, and A. Tomkins. Disambiguation of references to individuals. Technical report, HP, 2005.
6. E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. Technical report, W3C, 2006.
7. J. R. Quinlan. Learning efficient classification procedures and their application to chess end-games. *Machine Learning: An Artificial Intelligence Approach*, 1983.
8. J. R. Quinlan and R. M. Cameron-jones. Foil: A midterm report. In *In Proceedings of the European Conference on Machine Learning*, pages 3–20. Springer-Verlag, 1993.
9. M. Rowe. Interlinking distributed social graphs. In *Linked Data on the Web Workshop, WWW2009*, April 2009.
10. M. Rowe. Data.dcs: Converting legacy data into linked data. In *Linked Data on the Web Workshop, WWW2010*, 2010.
11. C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.
12. H. Yu, J. Han, and K. C. chuan Chang. Pebl: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16:70–81, 2004.