Applying Semantic Social Graphs to Disambiguate Identity References

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. Person disambiguation monitors web appearances of a person by disambiguating information belonging to different people sharing the same name. In this paper we extend person disambiguation to incorporate the abstract notion of identity. This extension utilises semantic web technologies to represent the identity of the person to be found and the web resources to be disambiguated as semantic graphs. Our approach extracts a complete semantic social graph from distributed Web 2.0 services. Web resources containing possible person references are converted into semantic graphs describing available identity features. We disambiguate these web resources to identify correct identity references by performing random walks through the graph space, measuring the distances between the social graph and web resource graphs, and clustering similar web resources. We present a new distance measure called "Optimum Transitions" and evaluate the accuracy of our approach using the information retrieval measure f-measure.

1 Introduction

In the modern web the need to automatically find information about a given individual has many applications such as monitoring web appearances for lateral surveillance, assessing the risk of identity theft, and protecting online visibility [3]. Malicious web practices have motivated average web users with a web presence to monitor their online visibility, and more importantly what aspects of their identity are visible in the public domain. The recent rise in the size of the social web, and particularly in the uptake of social networking sites has given members of the public their first searchable web presence from regular web entry points as sites such as Google[1] begin to index their online profiles. Therefore the need to monitor web appearances has grown, reflected in the rise in online monitoring services such Garlik Datapatrol[2].

One of the main challenges associated with monitoring web appearances is person disambiguation. For example if we are looking for information about a certain person there may be several people all sharing the same name. Therefore

[1] http://www.google.com
[2] http://www.garlik.com/

L. Aroyo et al. (Eds.): ESWC 2009, LNCS 5554, pp. 461–475, 2009.

a web page containing a person's name must be assessed to deduce whether the page contains a reference to the person we are interested in. The grounding for this problem is not uncommon in society where several people share the same name, we refer to such people as namesakes [14].

State of the art graph-based person disambiguation approaches rely on graph structures to measure distances between web resources, represented as sub graphs, according to available paths and edge weightings. These measures are then used to cluster resources where each cluster is representative of a different namesake. In this paper we present an complimentary approach to existing graph-based person disambiguation techniques using a three-stage procedure: We begin by generating individual semantic social graphs from dispersed Web 2.0 services, and link these graphs together to form a complete semantic social graph containing identity and social network information for a given person. Secondly we find possible web resources referencing a given person using a combination of crawling and searching both the web and the semantic web, and convert each resource into a semantic graph describing identity information found within that resource. Thirdly we integrate the social graph and resource graphs into a global graph space, merging instances of semantic concepts and replacing edges in the graph space. We cluster the web resource graphs closest to the social graph, thereby producing two cluster sets: Correct identity references, and ambiguous identity references.

Our intuition is that a given person will co-occur on a web page with people from his/her social network. Thus the focus of our approach is to utilise existing social information sources when disambiguating possible identity referencing web resources. When measuring distances between the social graph and web resource graphs we use two measures; commute time and optimum transitions. We evaluate the accuracy of our approach to clustering web resources using these measures against a baseline latent semantic analysis technique using f-measure.

The paper is structured as follows: Section 2 describes the overview of our approach, and details relevant requirements. Section 3 presents our methodology for social graph generation and linkage. Section 4 describes our methodology for web resource graph generation, and section 5 describes the techniques we have implemented for identity disambiguation in web resources. Section 6 describes our evaluation method; the data set we used, the chosen accuracy measures, and the results and discussions. Section 7 presents the state of the art associated with the three stages of our approach. We conclude the paper in section 8 by discussing the conclusions we have found from our work and future work.

2 Overview of the Approach

The identity disambiguation approach presented in this paper bootstraps an existing graph-based disambiguation process by incorporating a social graph. We believe that the use of the initial social graph when deriving measures among the nodes in the graph space provides a focussed approach. Current approaches rely on comparing every node with every other node in the graph space to derive distance measures, instead we only derive distances between the social graph

and the web resource graphs. Due to the rise in usage of Web 2.0 services such as social networking and blogging platforms, web users build descriptions of their social identity in dispersed information spaces. By harnessing and collating this decentralised information a complete integrated social graph is compiled, providing useful information to correctly match web resources containing identity references. In order to focus our approach to identity disambiguation we have defined several requirements that must be fulfilled:

- Provide machine-readable semantic information from semi-structured Web 2.0 services.
- Handle web resources lacking machine-readable semantics.
- Bootstrap the identity disambiguation process with minimal user input.
- Handle information on a large scale.
- Disambiguate people, not just names.
- Provide accurate identity disambiguation results.

In our approach we use a semantic graph structure. We define a semantic graph as a graph containing nodes represented by classes from an ontology, and edges as the relations and properties between those nodes. The Resource Description Framework (RDF) is well suited to describing graph structures based on the structure of a triple, subject-object-predicate expression, being comparable to an edge in a graph linking two nodes. The advantages of using a semantic graph structure allows inferences to be made about the graph structure, aids with graph integration and node merging, and allows graph formalisation adhering to a formal standard. Figure 1 provides an overview of our approach to identity disambiguation, we divide the approach into three stages:

1. **Social Graph Generation.** Handles the extraction of multiple social graphs from different Web 2.0 services as semi-structured XML. The social graphs are then converted to RDF and linked together into a complete social graph.
2. **Resource Graph Generation.** Responsible for finding possible identity referencing web resources by crawling and searching the web. The resources are each converted into a semantic resource graph as RDF.
3. **Identity Disambiguation.** Integrates the web resource graphs and the social graph into the graph space. Random walks are performed through the graph space to derive distance measures within the graph. These measures are then used to cluster the web resources most similar to the social graph.

3 Social Graph Generation

3.1 Extracting Social Graphs

The rise in usage of Web 2.0 sites and services has encouraged average web users to sign up to multiple dispersed services spreading their identity and their attributed social graph throughout the web. The majority of such services contain social information hidden within a 'walled garden'. Accessing such information

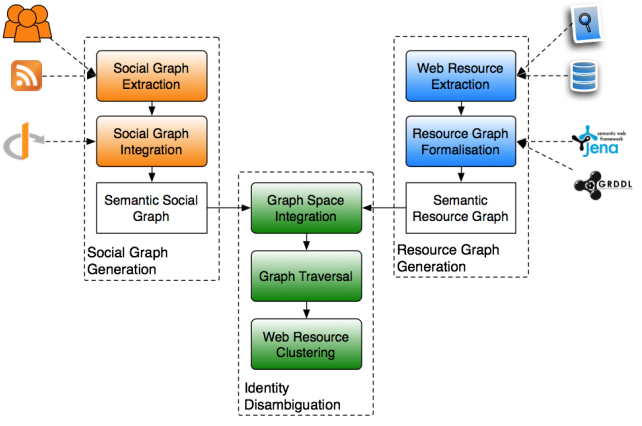

Fig. 1. Architecture of applying Semantic Social Graphs to Identity Disambiguation

is restricted to authorised API calls which - when validated - produce an XML response according to the service's XML schema.

Therefore the task of exporting this information is trivial in that in commonly involves mapping an existing XML schema produced from the API to existing semantic ontologies. We reuse an approach from our previous work [23] to produce RDF from the social networking site Facebook[3] by mapping the identity and social network infromation from the XML schema to concepts from the FOAF ontology [6], and geographical locations to the Geo [7] ontology using the top ranked URI for the location obtained from the GeoNames[4] web service. The selection of the FOAF ontology was a natural choice due to its ability to capture knowledge related to identity features and express social connections between instances of people. Figure 2 shows an example of the graph structure produced from a Facebook account. This social graph captures knowledge concerning the identity details of the person, and the social connections that exist with members of his/her social network described using the `foaf:knows` relation to connect two instances of `foaf:Person`. We replicate this process for social networking sites adopting Google's OpenSocial[5] standards such as MySpace[6], Orkut[7] and Hi5[8]. Available social information from accessing these services is converted into RDF acccording to the FOAF and Geo ontologies.

Many blogging platforms allow access to social information through unauthorised API calls, and return a response according to either the Atom [20] or RSS [28] syndication feed specifications depending on the platform. To generate a semantic graph we convert this information into RDF according to the FOAF

[3] http://www.facebook.com
[4] http://www.geonames.org
[5] http://code.google.com/apis/opensocial
[6] http://www.myspace.com
[7] http://www.orkut.com
[8] http://www.hi5.com

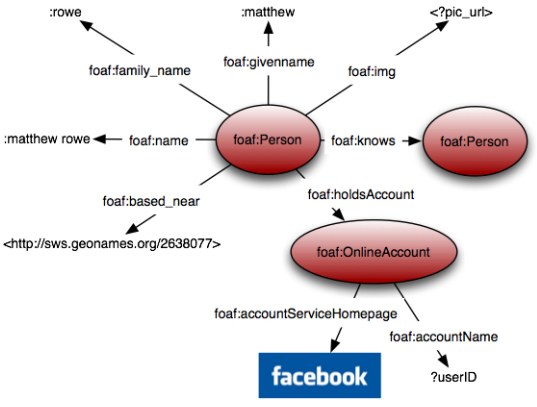

Fig. 2. RDF generated from Facebook according to the FOAF ontology

specification mapping the relevant concepts to capture all of the available social information. In the case of the Atom specification we convert the `atom:Author` concept to an instance of `foaf:Person`, the associated properties are also converted: `atom:name` and `atom:email` are converted to `foaf:name` and `foaf:mbox` respectively.

3.2 Linking Social Graphs

At this stage in the approach we have several social graphs each owned by the same person, expressed as a single unique instance of the `foaf:Person` class in each graph, we refer to this instance as the graph owner. We must link each of the graphs together and merge the structures. The problem we face is the lack of a unique URI that can be assigned to the graph owner, however we refrain from using the `foaf:name` property due to the ambiguity of person names. Therefore we assign an OpenID[9] resource using the `foaf:openid` relation to the graph owner in each of the social graphs. As `foaf:openid` is expressed as an inverse functional property of an instance of `foaf:Person` only one OpenID can be attributed to a given person. We also assume the converse to be true in that one person is restricted to only one OpenID to enforce social graph linkage.

Information in each sub graph is now linked together as figure 3 demonstrates, where all instances of the same `foaf:Person` class are now linked using the `owl:sameAs` property. These instances are then merged together to form one instance of `foaf:Person` for the owner of the complete social graph. We then merge the remaining social network content by comparing instances of `foaf:Person` from separate social networks for matches. We compare identity properties of each `foaf:Person` instance using the Smith-Waterman-Gotoh distance [8] for string literals, and look for matches when comparing identifiers such as `foaf:mbox` and `foaf:homepage`. If a match occurs then we merge the

[9] http://www.openid.net

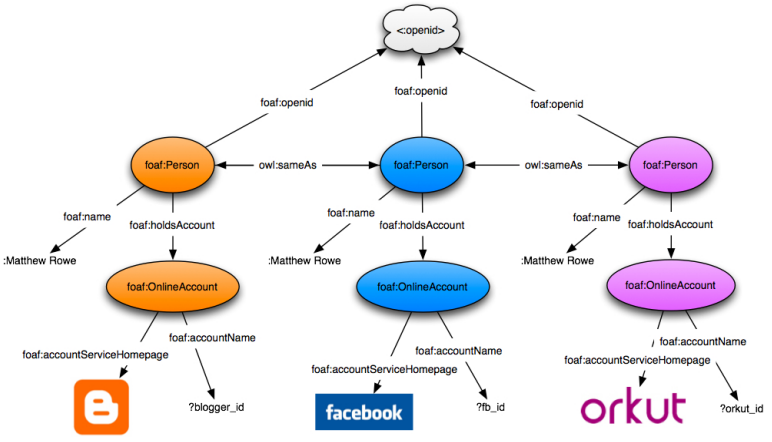

Fig. 3. Linked data across multiple social subgraphs

instances, otherwise we keep them separate in the complete social network. Our technique for linking social graphs is discussed in detail in [24].

3.3 Retrieving a Complete Social Graph

The complete social graph is stored within a triple store for exportation and linking of new triples. This allows SPARQL queries to be used in order to retrieve the complete social graph attributed to a given person using their OpenID. If we are to store social graphs for thousands of people in the same triple store, it is essential to have a URI for each person, hence the selection of OpenID when identiying the graph owners.

4 Resource Graph Generation

4.1 Gathering Resources

Our approach gathers resources using two methods; crawling and searching. We crawl web pages defined by the `foaf:homepage` property in the social graph, and known identity resources such as public telephone directories[10] and listings pages[11]. The crawler runs until a predefined number of web resources have been retrieved, the resources are then added to the resource list for graph conversion.

We search the web using search engines as entry points to generate a list of web resources for a given person using his/her name. URLs from the first 50 results are collected from Google and Yahoo[12] and are added to the resource list for graph conversion. The semantic web is searched using the semantic web

[10] http://www.thephonebook.bt.com
[11] http://www.192.com
[12] http://www.yahoo.com

search engines Watson[13] and Swoogle[14] to gather a list of resources containing semantic metadata using the person's name. Following this we then have at least 100 resources ready for graph generation for one identity.

4.2 Extracting Resource Graphs

A semantic graph structure is generated for each web resource using available identity information. Different techniques are used to perform the semantic graph generation depending on the available metadata within each resource. If the web resource is a web page containing lightweight semantics such as Microformats [15], eRDF [22], and RDFa [1], then the page is parsed using an XSL transformation [9] with GRDDL [10]. The advantage of this method is its adaptability to allow new lightweight semantic specifications to be adopted by simply including the accompanying XSL transformation. The extracted information in XML is converted to RDF using the FOAF, and Geo ontologies in a similar manner to the social graph generation obtaining a URI for the geographical location using the GeoNames service. We overcame the problem of differing usages of lightweight semantics (figure 4) by converting identity information into the same semantic format.

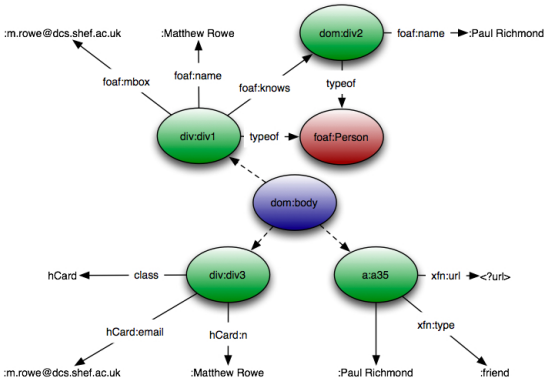

Fig. 4. Differing usage of lightweight semantics in web resources

Ontologies returned when searching the semantic web already contain machine-readable metadata therefore we parse the ontologies using Jena [17]. Any content described using the FOAF specification found within the ontology is extracted into a graph describing an instance of the `foaf:Person` class, attributed identity properties and social connections using the `foaf:knows` relation. We retrieve URIs described using the `foaf:seeAlso` property within the ontology and add these to the crawler, assuming that this linked data is likely to contain a reference to the

[13] http://watson.kmi.open.ac.uk/WatsonWUI
[14] http://swoogle.umbc.edu

person we are searching for. The derived resource graph can then be integrated into the graph space.

From our work we found that the majority of web resources contained no machine-readable semantics, instead the content is standard HTML. In such cases we use regular expressions formed from a person name gazetteer to match person names within the web resources. When a match occurs we create an instance of the `foaf:Person` class and assign the `foaf:name` property to express the found person name in the resource graph. When a person name match occurs, we match identity information such as email, homepage and date of birth using regular expressions surrounding the matched name through a window function of 50 words either side of the name. We use this window function on the assumption that any identity information relating to the given name would occur within the near proximity of the matched name.

5 Identity Disambiguation

5.1 Integration into the Graph Space

The graph space is initially populated with the social graph of the user, and iteratively with each web resource graph using an indexing paradigm. At a low-level this involves the comparison of semantic concepts and string literals in the case of semantic properties such as `foaf:name` to provide node deduplication. We use the Smith-Waterman-Gotoh string metric to compare string literal nodes, if the derived measure is above our assigned threshold we integrate the web resource graph into the graph space by replacing the internal node with the comparable node in the global graph. We also do the same for geographical semantic concepts by comparing location URIs. We perform this process until all web resource graphs have been integrated into the global graph space. Semantics play an important role in the merging of nodes within the graph space by limiting node comparisons to only nodes of the same semantic type. As more nodes are merged within the graph space the knowledge structure becomes enriched, providing more paths through the graph space.

In order to derive distance measures between nodes within the graph space we apply an edge weighting function based on the edge label. As each edge in the graph is described using a semantic property or relation it is possible to adjust the weighting model based on the preference of identity features. For example, the `foaf:homepage` property from the social graph acts as a URI therefore the edges labelled with the `foaf:homepage` property are given less weight to reduce the traversal cost. We normalise the weightings for semantic edge types in the graph using a standard weighting distribution.

5.2 Graph Traversal

Once the graph space has been fully integrated we derive the strongly connected component within the graph starting from the identity node, and prune the graph space of web resources containing no identity references. We define the

identity node as the `foaf:Person` concept in the social graph representing the social graph owner. We use Tarjan's algorithm [26] to derive the component by performing a depth first search and returning all connected nodes and edges from the identity node and therefore the strongly connected component.

Using this pruned graph we now traverse the graph space using random walks [25] starting from the identity node. We begin by deriving the adjacency matrix A to compute the local similarity of adjacent nodes in the graph as described in [13] by deriving the cost of moving from each node i to each adjacent node j. Let $L = (l_1, \cdots, l_n)$ denote the set of distinct semantic edge types (properties and relations) in the graph space, such that the function $w(l_k)$ returns a weight based on the semantic edge type, therefore we populate A using:

$$a_{ij} = \begin{cases} \sum l_k \in L \frac{w(l_k)}{|(i,.) \in E : l(i,.) = l_k|}, & (i,j) \in E \\ 0, & otherwise \end{cases}$$

We compute the diagonal degree matrix using $D = \sum {}_k A_{ik}$. A random walk through the graph is derived from a Markov chain [18] describing the probability of moving to node j given that the random walker is currently in node i given that t steps have been traversed through the graph space. Transition probabilities are defined as $P = D^{-1}A$ for one step, and $P^t = \left(D^{-1}A\right)^t$ for t steps. We now have a matrix which when consulted returns the probability of traversing from one node to another in t steps. At this stage we use two different techniques to derive distance measures between the identity node and the web resources in the graph space:

Commute Time. This measure derives the commute time taken to set off from the identity node, reach a web resource and then return again. We set the transition probability matrix for 2 steps due to the graph structure being relatively short in length between nodes. We use the formula described in [25] to derive the cost of moving to node k and returning back given that we are currently in state i:

$$m\left(k|i\right) = \begin{cases} 1 + \sum_{j=1}^{n} p_{ij} m(k|j), & i \neq k \\ 0 \end{cases}$$

This iterates through the graph cumulatively recording the cost of the path from the identity node to each web resource. Agglomerative clustering [27] is then used to cluster the most similar web resource nodes with the identity node based on the bottom-up clustering principle being well suited to comparison against a single social graph. The end product is 2 cluster sets: Correct identity references and ambiguous identity references (those resources that cannot be confirmed as referring to the identity we are interested in or not).

Optimum Transitions. The second measure derives the optimum number of steps needed to traverse from the identity node to web resource nodes within the graph space by incremening t until the transition probability has peaked. Given

that we wish to traverse from node i to node j, we derive the optimum value for t using $max(P_{ij}^t)$. We then use agglomerative clustering to cluster the most similar web resource nodes with the identity node using the same method as the Commute Time clusters, we similarly derive 2 sets of clusters; correct identity references and ambiguous identity references.

6 Evaluation

6.1 Data Set

We evaluate our approach using a data set compiled from members of the Department of Computer Science[15] at the University of Sheffield. We selected members by analysing results from the search engines Google and Yahoo using each person's name, and manually identifying different entities referenced by the same name. If a large number of namesakes appeared in the results then the person was included for the evaluation. We compiled the data set by listing social web accounts, and referent web resources for each participant as described in section 4. This data set was then used to generate the complete social graph and web resource graphs, and compile the graph space.

6.2 Selection of a Baseline Technique

We compared our approach against a standard baseline latent semantic analysis method using hierarchical clustering as presented in [16]. In this paper we wish to demonstrate the effectiveness of implementing a semantic graph-based approach in comparison with standard feature comparison techniques that do not use graph structures. Therefore the selection of latent semantic analysis provides a comparable implementation due to its feature based methodology, clustering resources by calculating similarity measures based on features in each resource [16]. We perform the comparison between the baseline method and our approach by comparing the derived clusters. Our approach clusters the social graph with the most similar web resource graphs, whereas the baseline method produces a number of clusters each relating to a different namesake. Therefore we set the cluster containing the person we are searching for as the correct identity referents and merge all other clusters to form the ambiguous identity referents. This allows comparison of the cluster sets between our approach and the baseline using the following measures:

6.3 Accuracy Measures

We evaluate both our approach and the baseline approach using the information retrieval metric f-measure to assess the minimisation of the number of false positives and false negatives. We evaluate the accuracy of the correct identity referents cluster where f-measure is well suited to deciding the accuracy of information retrieved. We define p as the identity we are looking for, s_p as the set

[15] http://www.shef.ac.uk/dcs

of web resources that correctly reference the identity p, c_c as the cluster containing correct identity references, and c_a as the cluster containing ambiguous identity references. We define precision as $P(c_c) = |s_p \cap c_c| / |c_c|$ and recall as $R(c_c) = |s_p \cap c_c| / |s_p|$, and therefore f-measure as the harmonic mean of the precision and recall:

$$F(c_c) = \frac{2P(c_c)R(c_c)}{P(c_c) + R(c_c)}$$

6.4 Results and Discussion

Table 1 shows the results from the evaluation. We have included the column "Namesake Count" to highlight the ambiguity of the people we are searching for by counting how many different namesakes appeared in the search results. We achieve an average F-Measure of 0.76 for commute time and an average F-Measure of 0.78 for optimum transitions. In each instance both semantic graph-based methods outperform the baseline. The f-measures indicate that we achieve a healthy combination of true positives and true negatives when disambiguating web resources in comparison to the baseline technique.

Table 1. Evaluation results

Name	# of Namesakes	Commute Time	Optimum Transitions	Baseline
Matthew Rowe	22	0.88	**0.9**	0.85
Jonathan Butters	12	**0.72**	**0.72**	0.6
Rodrigo Carvalho	16	**0.76**	**0.76**	0.39
Sam Chapman	23	**0.6**	0.52	0.33
Neil Ireson	10	**0.88**	**0.88**	0.75
Paul Richmond	18	0.47	**0.87**	0.83
Christopher Brewster	16	**0.97**	0.92	0.83
Mark Greenwood	19	**0.81**	0.7	0.23
Joao Magalhaes	16	**0.81**	0.78	0.47
Diana Maynard	11	0.71	**0.72**	0.45
Overall		0.76	**0.78**	0.573

The results shows that there is little difference between each distance measure used to cluster resources, in certain cases the commute time outperforms optimum transitions and in others the converse. Overall optimum transitions produce a more consistent measure with less fluctuation than commute time. For instance, when searching for web resources relating to the person "Paul Richmond", commute time is outperformed by both the baseline and optimum transitions. In this instance this was due to the large number of paths available through the graph space from the social graph to each of the web resource graphs, optimum transitions works better when dealing with a large selection of possible paths. With the other participants the paths varied less through the graph space, and therefore commute time functioned better.

7 Related Work

7.1 Social Graph Generation

Generating and 'semantifying' content from Web 2.0 services has been addressed in [21] by extracting information from the photo sharing social networking site Flickr[16] and converting the API response into RDF annotated using both the FOAF and SIOC [5] ontologies. A similar approach[17] provides an exporter of RDF according to the FOAF ontology from the microblogging site Twitter[18]. Our previous work in [23] provides an approach to export RDF according to the FOAF ontology from the social networking site Facebook that we used within this paper. The problem faced following the extraction of individual semantic social graphs from dispersed Web 2.0 services is data linkage where the use of a person name is too ambiguous, therefore a URI to link the graphs is a necessity. The inclusion of the `foaf:openid` property in the latest FOAF specification was useful for our work in two ways: Firstly, Web 2.0 services never allow the extraction of email addresses as they are deemed private, thus rendering the `foaf:mbox` and `foaf:mbox_sha1sum` properties unusable. And secondly, this provides a single URI for retrieving the social graph from a triple store, and linking additional triples.

7.2 Resource Graph Generation

Lightweight specifications such as Microformats, RDFa (Resource Description Framework in Attributes) and eRDF (Embedded RDF) allow "lowercase" semantic metadata to be included within web pages using XHTML. Using an XSL transformation (XSLT) with GRDDL to extract available metadata allows semantic graphs to be generated from web pages. Work by [2] provides a mechanism to extract RDF from XML based content (including HTML) by merging XQuery and SPARQL to create a query language called XSPARQL therefore allowing semantic metadata to be extracted from HTML without the need for XSLT. Our approach complements the state of the art by using an XSLT transformation where possible to extract lightweight semantics with GRDDL and convert this into an RDF graph. The most commonly used specification for describing identity information is FOAF, such information available on the semantic web has been mined in [19] using a two-part approach to find ontologies containing FOAF content, extracting relationships from within the ontologies and deriving bond strengths using name co-occurrence from web queries. The semantic web is crawled for ontologies annotated with FOAF in work by [11], FOAF content is extracted and aggregated with other FOAF files. This allows assertions to be made about the people that were discovered during the crawl using the extracted semantic information. We have implemented both searching and crawling mechanisms for ontology gathering from the semantic web.

[16] http://www.flickr.com
[17] http://tools.opiumfield.com/twitter/mattroweshow/rdf
[18] http://www.twitter.com

7.3 Identity Disambiguation

Work by [16] uses an unsupervised method to perform person disambiguation by searching for web pages using a person name, and clustering web pages accordring to the community struture in each page. A similar approach is presented in [27] where named entities are extracted from within web pages, along with biographical information such as date of birth and email. These features are then used to cluster similar pages as belonging to different namesakes. Person name ambiguity is addressed in [12] by aligning a namesake with a relevant concept using a maximum entropy model, and clustering similar web resources based on the concept.

A graph-based approach is described in [4] utilising the link structure of web pages to cluster similar pages as belonging to different people working under the intuition that "Web pages of a group of acquaintances are likely to be interconnected". Link structures are modelled in a graph space, and web pages are clustered based on the distances between them. A similar approach is described in [27] by extracting names and hyperlinks from within web pages as features. Unlike [4] the graph model is extended, similar to our work, by setting nodes in the graph space as web pages, and named entities and hyperlinks found within those pages. Correlation clustering is then used to cluster similar web resources based on the strengths of the connections between them. A model of the graph space is populated in [13] by extracting various web page features, such as named entities, hyperlinks, and lexical tokens. Each feature is modelled as a node in the graph space, and edges are labelled between the nodes. Random walks are then performed through the graph space to derive distance measures according to the commute time between nodes. We address the disambiguation problem with respect to current state of the art methods by providing a semantic interpretation of the graph space using available ontologies to describe the nodes and edges. Our approach to identity disambiguation differs to the state of the art by clustering similar web resources with respect to the social graph rather than creating individual clusters for each namesake.

8 Conclusions

Our approach produced a semantic social graph from semi-structured XML describing the identity of the account holder and their social network, therefore fulfilling the requirement for producing machine-readable semantic information by generating RDF. The majority of web resources that were analysed for identity information contained no lightweight semantics. We believe that this is due to the slow uptake by web developers in annotating web content using lightweight semantics. Nevertheless, our approach was able to extract identity information from web resources using a gazetteer with regular expressions and convert this information into a semantic format.

Our work bootstraps an existing identity disambiguation technique minimising the user input by only involving the user in the social graph generation stage.

The bootstrapping focuses the derivation of distances within the graph space between certain nodes, by only measuring from the social graph. This requires less computations than the current state of the art by focussing the approach on one identity, rather than trying to produce a cluster containing web resources for each namesake, thus allowing more data to be handled.

We extended person disambiguation to perform identity disambiguation by modeling identity features captured throughout the social graph generation and resource graph generation stages of our approach. Preference is given to certain identity properties by weighting the semantic edge types in the graph space. For example, disambiguation is heavily influenced if the `foaf:mbox` (email address) property from the social graph is present in a web resource due to an email address being a URI. The results from our evaluation demonstrate the effectiveness of our approach of using a graph-based disambiguation in comparison with the baseline latent semantic analysis technique. The results also show that our optimum transitions measure for deriving distances between nodes in the graph space outperforms the commute time measure. We also experienced a more efficient process requiring less time to compute the distances, we plan to document this improved efficiency in comparison to commute time in our future work.

References

1. Adida, B., Birbeck, M.: RDFa Primer: Bridging the Human and Data Webs. World Wide Web Consortium (2008), http://www.w3.org/TR/xhtml-rdfa-primer/
2. Akhtar, W., Kopecky, J., Krennwallner, T., Polleres, A.: XSPARQL: Travelling between the XML and RDF Worlds – and Avoiding the XSLT Pilgrimage. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 432–447. Springer, Heidelberg (2008)
3. Andrejevic, M.: The Discipline of Watching: Detection, Risk, and Lateral Surveillance. Critical Studies in Media Communication 23(5), 392–407 (2006)
4. Bekkerman, R., McCallum, A.: Disambiguating Web Appearances of People in a Social Network. In: Proc. 14th international conference on World Wide Web, Chiba, Japan, pp. 463–470 (2005)
5. Breslin, J., Harth, A., Bojars, U., Decker, S.: Towards Semantically Interlinked Online Communities. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 500–514. Springer, Heidelberg (2005)
6. Brickley, D., Miller, L.: FOAF Vocabulary Specification (OpenID Edition). Creative Commons (2007), http://xmlns.com/foaf/spec/
7. Brickley, D.: Basic Geo Vocabulary. World Wide Web Consortium (2006), http://www.w3.org/2003/01/geo
8. Chapman, S., Norton, B., Ciravegna, F.: Armadillo: Integrating Knowledge for the Semantic Web. In: Proc. Dagstuhl Seminar in Machine Learning for the Semantic Web, Dagstuhl, Germany (2005)
9. Clark, J.: XSL Transformations. World Wide Web Consortium (1999), http://www.w3.org/TR/xslt
10. Connolly, D.: Gleaning Resource Descriptions from Dialects of Language. World Wide Web Consortium (2007), http://www.w3.org/TR/grddl/
11. Finin, T., Ding, L., Zhou, L., Joshi, A.: Social Networking on the Semantic Web. The Learning Organisation 1(5), 418–435 (2005)

12. Fleischman, M., Hovy, E.: Multi-Document Person Name Resolution. In: Proc. Workshop on Reference Resolution and its Applications: ACL 2004, Barcelona (2004)
13. Iria, J., Xia, L., Zhang, Z.: WIT: Web People Search Disambiguation using Random Walks. In: Proc. of the 4th International Workshop on Semantic Evaluations (Semeval 2007), Prague, Czech Republic (2007)
14. Kalashnikov, D., Chen, Z., Mehrotra, S., Nuray, R.: Web People Search via Connection Analysis. IEEE Transactions on Knowledge and Data Engineering 20(11), 1550–1565 (2008)
15. Khare, K.: Microformats: the next (small) thing on the Semantic Web? IEEE Internet Computing 10, 68–75 (2006)
16. Malin, B.: Unsupervised Name Disambiguation via Social Network Similarity. In: Proc. Workshop on Link Analysis, Counterterrorism, and Security, SIAM International Conference on Data Mining, Newport Beach, CA (2005)
17. McBride, B.: Jena: a Semantic Web toolkit. IEEE Internet Computing 6, 55–59 (2002)
18. Meyn, S., Tweedie, R.: Markov chains and stochastic stability. Springer, London (1993)
19. Mika, P.: Bootstrapping the FOAF-Web: An Experiment in Social Network Mining. In: Proc. Workshop on Friend of a Friend, Social Networking and the Semantic Web, Galway, Ireland (2004)
20. Nottingham, M.: Atom Syndication Specification. The Internet Society (2005), `http://www.atomenabled.org/developers/syndication/atom-format-spec.php`
21. Passant, A.: RDF Export of Flickr Profiles with FOAF and SIOC (2007), `http://apassant.net/blog/2007/12/18/`
22. RDF in HTML. Talis (2006), `http://research.talis.com/2005/erdf/`
23. Rowe, M., Ciravegna, F.: Getting to Me - Exporting Semantic Social Network Information from Facebook. In: Proc. Social Data on the Web Workshop, ISWC 2008, Karlsruhe, Germany (2008)
24. Rowe, M.: Interlinking distributed Social Graphs. In: Proc. Linked Data on the Web Workshop, WWW 2009, Madrid Spain (2009)
25. Saerens, M., Fouss, F., Yen, L., Dupont, P.: The principal components analysis of a graph, and its relationships to spectral clustering. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS, vol. 3201, pp. 371–383. Springer, Heidelberg (2004)
26. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM Journal on Computing 1(2), 146–160 (1972)
27. Wan, X., Gao, J., Li, M., Ding, B.: Person resolution in Person Search Results: WebHawk. In: Proc. of the 14th ACM international conference on Information and knowledge management, pp. 163–170 (2005)
28. Winer, D.: RSS 2.0 Specification. Creative Commons (2007), `http://www.rssboard.org/rss-specification`