Learning to Classify Identity Web References using RDF Graphs

Matthew Rowe and José Iria
The OAK Group
Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield S14DP UK
{m.rowe,j.iria}@dcs.she.ac.uk

ABSTRACT

The need to monitor a person's web presence has risen in recent years due to identity theft and lateral surveillance becoming prevalent web actions. In this paper we present a machine learning-inspired bootstrapping approach to monitor identity web references that only requires as input an initial small seed set of data modelled as an RDF graph. We vary the combination of different RDF graph matching paradigms with different machine learning classifiers and observe the effects on the classification of identity web references. We present preliminary results of an evaluation in order to show the variation in accuracy of these different permutations.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*

Keywords

Semantic Web, Social Web, Machine Learning, Identity, RDF

1. INTRODUCTION

The modern web user has a presence on the web which is accessible through search engines and web gateways. This increased presence has lead to unwanted by-products such as disseminating personal information without a person's knowledge. As a result the online privacy of a person is reduced and has lead to a rise in activities such as lateral surveillance [1] and identity theft. In order to address these issues personal information describing a given person must be found so that the correct actions can be taken (i.e. Removing the information). Such is the scale of the web, automatic methods are required to monitoring identity web references (web pages which contain a reference to a given person).

In this paper we present an application of self-training in order to learn to classify identity web references of a given person. Self-training is a type of semi-supervised learning which iteratively learns, and improves, a classifier from labeled training data. We address one of the hard problems in machine learning, the lack of training data, by adopting a bootstrapping technique to build a classifier from very little seed data. We model web resources (ontologies, web pages, XML feeds) as RDF graphs describing the underlying knowledge in the resource. This enables features of the learning instances to be modeled as RDF instances from the RDF graph and permit the variation of the feature similarity measure used. Figure 1 presents an overview of the approach which is divided into two areas: Seed data generation and learning to classify.

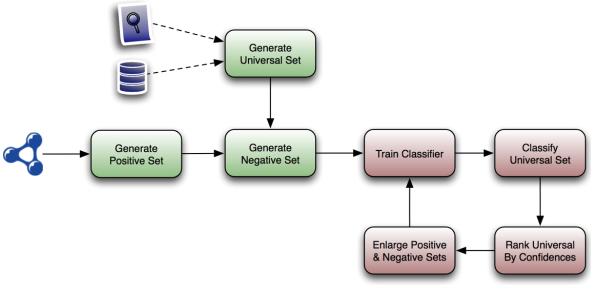

Figure 1: An approach of learning to classify identity web references

2. GENERATING SEED DATA

Our approach uses three sets of data; the positive set, the negative set and the universal set which we refer to as P, N and U respectively. The positive set contains correct identity web references, the negative set contains web pages which do not contain an identity reference and the universal set contains web pages which are unlabeled and yet to the classified.

It is now common for the majority of web users to have more than one account or profile on the social web, such profiles contain useful information describing a person's identity which can then be utilised as seed data for our approach. Therefore, we compile seed data by extracting information from platforms on the social web, returning XML, and lifting this to RDF thereby producing a social graph. The social graph contains social network and biographical information modeled using the FOAF [1] and GeoNames[2] ontologies. We link together several social graphs from different social web platforms thereby forming a complete identity representation of a given person [4], which is added to P. We then generate RDF models for resources linked to the person within the social graph (i.e. Homepage, blog, work page) and add them to P.

[1] http://xmlns.com/foaf/spec/
[2] http://www.geonames.org/ontology/

Using the person's name as the query, we populate U by searching the web through Google and Yahoo and the semantic web through Watson[3] and Sindice7[4]. RDF models are generated for each web resource in the results using GRDDL [3] should the resource contain lowercase semantics such as RDFa or Microformats. If the resource contains no such semantics, a person name gazetteer is employed to find the occurrence of a name within the page and use a window function of 50 tokens either side of the name to find contextually relevant information; email, webpage, location, etc. At this stage in the approach P is relatively small wheras U is large (as it contains many web pages to be classified).

Negative examples are selected from the universal set using a similar strategy to the one described by Yu et al in [5]: A list of positive features is compiled from instances in the positive set. All instances from the universal set are then filtered out that contain any strong positive features from the list, leaving instances that contain no positive features and are therefore more likely to be strong negatives. We constrain the size of the negative set forcing it to be roughly the same size as the positive set, both in the number of instances and the number of features within those instances.

3. LEARNING TO CLASSIFY

Knowledge within each instance in P, N and U is represented semantically as an RDF model. As we follow the intuition that a given person will co-occur on the web with members of his/her social network, we use instances of *foaf:Person* found within the model as features. We are able to vary the similarity measured when the *foaf:Person* instances between three methods: Jaccard Similarity (graph edit distance) [2], Inverse Functional Property matching (detecting equivalent unique resources in either graph) and RDF Entailment[5] (detecting triplesets from one graph subsuming the tripleset of another graph). We map the instances in P, N and U into a dataset that is compatible with the input of any of the learning algorithms (Naive Bayes, Support Vector Machines and Perceptron). Naive Bayes extracts the conditional probabilities from the information in the dataset, and SVM and Perceptron build their separating hyperplanes by optimization also based on the information in the dataset.

Our self-training strategy begins by first using the seed data to train a classifier, at this stage the size of the sets are relatively small. We classify instances from U using the trained classifier and rank the instances based on their classification confidence. We then choose the strongest positive instances and the strongest negative instances from U according to rankings, and enlarge P and N respectively whilst removing those instances from U (as shown in figure 2). The process is repeated by retraining the classifier and reapplying it to U until U is empty.

4. PRELIMINARY RESULTS

Evaluation of the approach has been completed for 20 members of the semantic web and web 2.0 communities. For each participant 9 different permutations of classifier (Naive Bayes, Perceptron and SVM) and RDF graph similarity measure were tested and the derived classifications were compared with a manually created gold standard for each test participant using the information retrieval measures; precision, recall and f-measure.

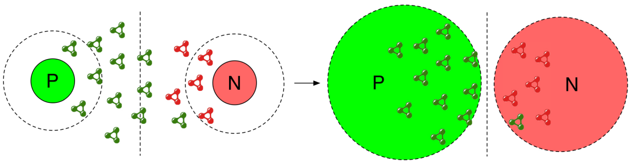

Figure 2: Initial training sets iteratively enlarged using resources with the strongest confidence scores

Table 1: Classification results of a combination of machine learning classifier and RDF graph similarity measure

Classifier	Sim' Measure	Prec'	Reca'	F-Meas'
Perceptron	Jaccard	0.943	0.319	0.427
	IFP	0.817	0..386	0.453
	Entailment	0.516	0.652	0.459
Naive Bayes	Jaccard	0.887	0.433	0.521
	IFP	0.743	0.433	0.477
	Entailment	0.621	0.669	0.577
SVM	Jaccard	0.891	0.429	0.517
	IFP	0.795	0.435	0.477
	Entailment	0.680	0.704	0.634

The preliminary results (shown in table 1) indicate that Perceptron and Jaccard similarity produce the most permutation results whereas SVM combined with Entailment yields the highest recall and f-measure score. In terms of the intentions of our work we wish achieve accurate classifications whilst maximizing data coverage, therefore based on the preliminary results our chosen permutation of classifier and similarity measure would be SVM combined with Entailment.

5. REFERENCES

[1] Andrejevic, M.: The Discipline of Watching: Detection, Risk, and Lateral Surveillance. *Critical Studies in Media Communication.* vol. 23, pp392-107 (2006)

[2] Bunke, H., Dickinson, P., Kraetzl, M., and Wallis, W.: A Graph-Theoretic Approach to Enterprise Network Dynamics. *Progress in Computer Science and Applied Logic (PCS)*, vol. 24, pp. 110 (2006)

[3] Connolly, D.: Gleaning Resource Descriptions from Dialects of Language. *World Wide Web Consortium* `http://www.w3.org/TR/grddl/` (2007)

[4] Rowe, M.: Interlinking Distributed Social Graphs. *In: Proc. Linked Data on the Web Workshop, WWW09.* Madrid, Spain. (2009)

[5] Yu, H., Han, J., and Chang, K.: PEBL: Web Page Classification without Negative Examples. *IEEE Transactions on Knowledge and Data Engineering.* IEEE Educational Activities Department, vol. 16.1, pp. 70-81 (2004)

[3] http://watson.kmi.open.ac.uk/WatsonWUI/

[4] http://www.sindice.com

[5] http://www.w3.org/TR/2004/REC-rdf-mt-20040210/#entail