Spamming in Linked Data

Ali Hasnain, Mustafa Al-Bakri, Luca Costabello, Zijie Cong, Ian Davis, Tom Heath

Spamming in Linked Data

Ali Hasnain[1], Mustafa Al-Bakri[2], Luca Costabello[3], Zijie Cong[4], Ian Davis[5], and Tom Heath[6]

[1] Digital Enterprise Research Institute, National University of Ireland, Galway
ali.hasnain@deri.org
[2] University of Grenoble, France
Mustafa.Al-Bakri@imag.fr
[3] INRIA, France
luca.costabello@inria.fr
[4] Universidad Rey Juan Carlos, Móstoles, Spain
zijie.cong@urjc.es
[5] Independent, United Kingdom
me@iandavis.com
[6] Talis Education Ltd, United Kingdom
tom.heath@talis.com

Abstract. The rapidly growing commercial interest in Linked Data raises the prospect of "Linked Data spam", which we define as "deliberately misleading information (data and links) published as Linked Data, with the goal of creating financial gain for the publisher". Compared to conventional technologies affected by spamming, e.g. email and blogs, spammers targeting Linked Data may not be able to push information directly towards consumers, but rather may seek to exploit a lack of human involvement in automated data integration processes performed by applications consuming Linked Data. This paper aims to lay a foundation for future work addressing the issue of Linked Data spam, by providing the following contributions: i) a formal definition of spamming in Linked Data; ii) a classification of potential spamming techniques; iii) a sample dataset demonstrating these techniques, for use in evaluating anti-spamming mechanisms; iv) preliminary recommendations for anti-spamming strategies.

Keywords: Linked Data, Spam, Spam Vectors

1 Introduction

Adoption by organisations such as the BBC[7], New York Times[8] and BestBuy[9] demonstrates a rapidly growing commercial interest in Linked Data. Initiatives such as schema.org[10] further increase the financial incentives to participate in

[7] http://www.w3.org/2001/sw/sweo/public/UseCases/BBC/
[8] http://data.nytimes.com/
[9] http://bit.ly/bestbuy-rdfa
[10] http://www.schema.org/

a Web of Data by boosting the publisher's prominence in search engine results. These factors increase the likelihood of rogue users attempting to pollute the Web with malicious data that brings them some financial gain, creating a form of "Linked Data spam". In the Linked Data context we define spam as *"deliberately misleading information (data and links) published as Linked Data, with the goal of creating financial gain for the publisher"*.

The emphasis of Linked Data, on publishing and interconnecting machine-readable data on the Web, means spam in this context differs from that found in traditional channels such as wikis and blog comments. Those environments provide convenient mechanisms for publishers to push information towards consumers, a process which does not have direct analogues in the Linked Data paradigm. However, as an open environment, the Web of Data allows any party to publish their own data, therefore seek to subvert the mechanisms of Linked Data consumption to exploit end users.

In tandem with ongoing challenges such as performance, reliability and security [14], applications consuming Linked Data may be severely impacted by the discovery, integration and usage of spam data from the Web. Spamming may take the form of undermining the trust assumption between datasets, introducing malicious triples or modifying existing content, determining undesirable behaviour in end-user applications.

To date, the Linked Data and Semantic Web research community has not systematically addressed the issue of Linked Data spam. As applications consuming Linked Data become more widespread this need must be addressed. This is the goal of the work presented here. While harmful attacks can be performed on other layers of the Internet infrastructure (e.g. DNS spoofing, DoS attacks), the scope of this work is limited to spam that exploits Linked Data specifications. The contributions of this work include: i) definition of spamming in Linked Data; ii) classification of possible spamming techniques; iii) sample dataset for demonstrating spamming techniques and evaluating anti-spamming mechanisms; iv) preliminary recommendations for anti-spamming strategies in Linked Data.

The remainder of this paper is organized as follows: Section 2 presents related work at the intersection of research into spam and Linked Data. Section 3 presents and classifies spammer goals in Linked Data before outlining a number of spamming techniques. In Section 4 we describe a novel sample dataset that embodies the attack techniques presented, while Section 5 proposes preliminary anti-spamming strategies in Linked Data.

2 Related Work

Efforts have been done for assessing the quality of the data available on the Web. We classify the related work in two categories namely *spamming in traditional Web* and *spamming in Linked Data*. For Spamming in LD there are two further relevant types of studies, *Data Provenance for Quality Assessment* and *Data Quality Assessment using Heuristics*.

In the area of spamming in the traditional web, Zoltan et al. [8] systematically study spamming techniques targeting the web and organize different web spamming techniques into a taxonomy with two major types, namely 1) *Boosting Techniques* relying on text fields in which spamming occurs and 2) *Link Spamming Techniques*, based on incoming and outgoing links. The authors also claim that their work can provide a framework for fighting spamming on the Web and propose a spam taxonomy used to define countermeasures. WebID[11], an early initiative by Brickley and Berners-Lee, aims at uniquely identify a person, a company, or any agent or organisation on the Web relying on the FOAF+SSL authentication scheme[12]. This unique ID might help identifying malicious attacks or spamming activity. Machine learning techniques, such as Bayesian classifiers [21,6] and neural networks [17,20,18] are employed with users' feedback and potential spam content as inputs. Taudgian [19] discusses email spam classification methods such as neural networks, Support Vector Machine (SVM)classifiers, Nave Bayesian (NB) classifiers and J48 classifiers. Spam blocking in blogs is discussed by Mishne et al. [13].

Spamming in Linked Data faces additional challenges compared to the traditional Web. In a blog post titled "*Linked Data Spam* Vectors"[13], Davis discusses a number of attack vectors for spammers in Linked Data (see Section 3). In [9] Hartig and Zhao propose a strategy for using provenance of Web data for assessing its quality. Their three-staged approach for assessment consists of 1) *provenance element collection*, 2) *decision of influence of elements on the assessment* and 3) *function application for quality calculation*. Provenance elements collected at first stage and their relationships amongst each other are useful for finding the provenance information for example "whether specific provenance element is created by the actual creator of specific data item or not" [9]. In their work they consider information quality (IQ) as collective value of multiple information quality criteria, e.g accuracy, completeness and timeliness as proposed by [16]. In the domain of information quality criteria, Lee et al. [12] propose a quality assessment methodology that measures IQ whereas Bobrowski et al [4] propose an assessment strategy using questionnaires. Detecting malicious data can be difficult when a large amount of triples from a spammer seem to be useful. Motro and Rakov propose an automated method to access the evaluation of reliability and completeness of different data sources [15]. A prediction algorithm used to find and calculate the response times of Web data sources is proposed by Gruser et al. [7]. Ballou et al. introduce a quantitative method for assessing the measurement and calculation for the timeliness of data which is based on provenance-related knowledge [2]. The WIQA[14] Information Quality Assessment Framework proposed by Bizer et al., is a collection of software components that deliver quality assessment guidelines for filtering information available on the Web. The WIQA framework is developed to fulfill three ba-

[11] `http://www.w3.org/wiki/WebID`
[12] `http://www.w3.org/wiki/Foaf\%2Bssl`
[13] `http://blog.iandavis.com/2009/09/21/linked-data-spam-vectors/`
[14] `http://www4.wiwiss.fu-berlin.de/bizer/wiqa/`

sic requirements namely 1) *flexible representation of information together with quality-related meta-information*, 2) *support for different information filtering policies*, and 3) *explaining filtering decisions*. W3C established the Provenance Incubator Group[15] to create an up-to-date roadmap in the area of provenance in Semantic Web for standardization efforts. The group discusses the importance of provenance and defines the requirements for provenance in the Web architecture.

Bizer et al. [3] classify *Data Quality Assessment using Heuristics* into three main categories based on the type of information used as quality indicators, namely 1) *Content-based Heuristics* which use information to be assessed itself, 2) *Context-based Heuristics* which rely on meta-information along with the circumstances and conditions in which information was originated and 3) *Rating-based Heuristics* which rely on number of factors including ratings about information, sources of information, and/or the credibility of data providers. Heath and Bizer [10] discuss the problem of accessing and improving quality on the client side and suggest that, after data quality on the web has been assessed based on any of the aforementioned heuristic, it can be ranked, fused or filtered according to the requirements.

3 Type of Attacks

In this section we analyse hypothetical spam attacks to the Web of Data. Although we do not provide a comprehensive assessment of threats, we raise awareness on spamming, as the issue might play a relevant role in the uptake of Linked Data. We start by listing spammer goals in the context of Linked Data. We then provide a list of "spam vectors"[13], i.e. techniques to introduce spam at RDF-level. Finally, we describe how spammers might introduce these vectors in the Web of Data infrastructure.

3.1 Spammer Goals in Linked Data

The ongoing evolution of the Web of documents into the the Web of Data preserves two fundamental features that might ease the task of Linked Data spammers: first, the open world assumption still holds and second, the cost of the infrastructure to publish and consume triples has not significantly raised. The ultimate goal does not change either, as spammers still aim at providing unsolicited content to users. Our assumption is that, being Linked Data part of the Web, spammer goals replicate well-known observed and studied patterns. We thus adapt and extend the taxonomy proposed in [8] for the context of spam in Linked Data:

Application pollution. Linked Data applications use the Web of Data as information source. If spammers infect this dataspace, application results might be polluted with malicious content, thus delivering spam to end users *directly* inside applications.

[15] `http://www.w3.org/2005/Incubator/prov/`

Improve ranking. Spammers focus on improving the ranking of malicious resources on search engines, by creating triples containing malicious literal values (to influence term-based metrics), or by creating fake external links to the resource, thus trying to influence algorithms that compute scores according to link information.

Hiding. Spammers must hide malicious content, trying to outsmart spam protection techniques.

3.2 Spam Vectors

To achieve the objectives described above, Semantic Web spammers necessarily need to deal with the RDF datamodel and Linked Data patterns. Triple-level and HTTP-related operations are therefore required to introduce pollution and achieve the desired results. A number of these Linked Data *Spam Vectors* have already been described[13]. We categorize these spam vectors in three groups and discuss their implications on Linked Data actors. Besides, we add four new attacks, *misleading dataset description*, *inverse-functional property cloning*, *presentation knowledge pollution* and *malicious subclassing*.

1. **Content contamination vectors.** Spammers might pollute Linked Data by introducing malicious information at triple level, typically exploiting popular properties of well-known vocabularies. The most relevant threats might origin from the following attacks:

   False labelling. It consists in adding `rdfs:label` triples containing spam in literals. The `rdfs:label` property provides a user-friendly label meant to be displayed by applications. Spammers exploit this common practice, to take control of prominent areas of Linked Data application UIs, thus increasing spam effectiveness. A similar attack could target `rdfs:comment` property, thus leading to "False commenting".

```
example:Motorola rdfs:label "BUY CHEAP GRAVIA"@en.
```

Listing 1.1: False labeling example

   Misattribution. The `dc:creator` property might be used to associate malicious statements to unaware authors (that could be people, organizations or services), as in Example 1.2.

```
:q1 a bibo:Quote;
    bibo:content "I  buy REPLICA WATCHES on replicaking.com";
    dc:creator "Tim Berners-Lee".
```

Listing 1.2: Misattribution example

   Schema pollution In this case instances are not polluted and contain therefore clean data. Spammers add spurious content inside the schema declaration. For instance, a spammer might publish a polluted copy of a well known vocabulary, publish it with a deceiving URI and pollute with

the latter services such as `prefix.cc`[16]. The attack directly targets application UIs, as spammers use it to pollute triple representation. The example in Figure 1.3 contains a portion of a polluted FOAF schema. The spammer created a semantically identical schema, and added spam inside `rdfs:label`s.

```
foaf:Person a owl:Class;
   rdfs:label "Buy GRAVIA at graviamaster.com";
```

Listing 1.3: Schema pollution example

Misleading dataset description. VoID descriptions[17] associated to datasets favour data discovery and help users and applications identify the right data. Deceiving VoID descriptions might be associated to malicious datasets, thus cheating Linked Data consumers. The attack undermines data discovery, if this procedure relies on dataset metadata. Link traversal is affected, as techniques might rely on choosing target datasets on the fly, according to VoID descriptions. The example below states that the `:graviaMaster` dataset contains triples about computer science publications.

```
:graviaMaster a void:Dataset ;
      foaf:homepage <http://www4.wiwiss.fu-berlin.de/dblp/all>;
      dcterms:subject <http://dbpedia.org/resource/Computer_science>;
      dcterms:subject <http://dbpedia.org/resource/Proceedings>.
```

Listing 1.4: Misleading dataset description example.

Malicious subclassing. Simple reasoning processes might generate undesired results due to RDFS/OWL-based attacks. Attackers explicitly targeting reasoning services might design polluted inference chains, e.g. associating to malicious classes a number of regular classes, thus determining reasoning pollution (Example 1.5).

```
gm:GraviaBuyer a owl:Class;
        foaf:homepage <http://graviamaster.com/>;
        rdfs:label "A class for premium GRAVIA customers".

foaf:Person rdfs:subClassOf gm:GraviaBuyer.
```

Listing 1.5: Malicious subclassing example.

Presentation knowledge pollution. RDF resources might be associated to presentation-level information. Fresnel[18] rendering engines are backed by a vocabulary including presentation-level concepts for RDF. In Fresnel, *Lens* components select and filter information while *Formats* define how to present data. Spammers might create malicious Fresnel declarations and associate this data to meaningful triples, since Fresnel specification does not explicitly say *who* must provide this data - it could

[16] `http://prefix.cc`

[17] `http://semanticweb.org/wiki/VoID`

[18] `http://www.w3.org/2005/04/fresnel-info/manual/`

be the dataset provider, application developers or third-party providers. (Example 1.6). Applications adopting Fresnel might end up with polluted results, even if instances and schemas are not, as the attack is performed at presentation level.

```
:nameFormat rdf:type fresnel:Format;
    fresnel:propertyFormatDomain foaf:name;
    fresnel:label "This person buys GRAVIA on graviamaster.com".

:knowsFormat rdf:type fresnel:Format ;
            fresnel:propertyFormatDomain foaf:knows ;
            fresnel:propertyFormat
              [ fresnel:contentAfter "loves graviamaster.com!"].
```

Listing 1.6: The Fresnel format `:nameFormat` overrrides `foaf:name` label with spam while `:knowsFormat` appends a spam string after each contact.

2. **Link poisoning vectors.** A certain number of spam techniques in Linked Data rely on links to external resources.
   Identity assumption consists in associating a malicious resource to misleading `owl:sameAs` properties, thus associating spam to informative entities. Applications that heavily rely on equivalence links might be polluted during the identity resolution phase. Heuristics must be introduced to mitigate the issue on the client side.

```
example:Amazon owl:sameAs <http://85.122.5.65/gravia>.
```

Listing 1.7: Identity assumption example.

Inverse-functional property cloning Using `owl:sameAs` is not the only way to determine that two URIs identify the same real-world resource. Inverse functional properties, coupled with simple reasoning determine the same result (e.g. ISBN code for a book, social security ID for people, etc.). Spam might be introduced in datasets by associating copies of inverse-functional properties to malicious resources. These resources will be considered identical to the original entity by reasoners and entity smushing heuristics (Example 1.8).

```
timbl:i a foaf:Person ;
      foaf:homepage <http://www.w3.org/People/Berners-Lee> ;
      foaf:name "Tim Berners-Lee".

ex:person_id3423 a foaf:Person ;
      foaf:homepage <http://www.w3.org/People/Berners-Lee> ;
      foaf:name "Buy GRAVIA at graviamaster.com".
```

Listing 1.8: Inverse-functional property cloning.

Misdirection. This type of attack is one of the simplest: spammers change URIs in triples, pointing to malicious content. As seen for other attacks, popular properties might be affected (see Example 1.9). Resource interlinking via the `rdfs:seeAlso` property can be polluted, too, thus perturbing link traversal operations such as follow-your-nose navigation or introducing undesired triples while crawling the Web of Data.

```
timbl:i foaf:depiction <http://i.imgur.com/15dr2.jpg> .
```

Listing 1.9: Misdirection example.

Data URI Embedding Spammers may include malicious data in triples exploiting the `data URI` scheme[19] in literals. At rendering time, applications will decode the base64-encoded content and display malicious messages. The example in Figure 1.10 includes a malicious link `<a href="http://graviamaster.com">gravia</a>` that will be included in the HTML rendering of the triple.

```
example:Amazon rdfs:seeAlso
  <data:text/html;charset=utf-8;base64,
           PGEgaHJlZj0iaHR0cDovL2dyYXZpYW1hc3Rlci5jb20iPmdyYXZpYTwvYT4=>
```

Listing 1.10: Data URI Embedding example.

3. **Non-triple-based attacks** Threats might origin from Linked Data features beyond RDF triples, such as resource dereference mechanism and reasoning.
   Bait and switch This attack exploits content negotiation mechanism to serve spam. An attacker might provide clean content or spam according to the value of the `Accept:` HTTP header (e.g. spam when RDF is needed and good data when HTML is requested, or vice-versa). This attack shows that application developers must never take for granted that HTML triple representation matches RDF serializations.

Table 1 summarises the impact of spam vectors on common operations performed by applications that consume Linked Data. Although all attacks are aimed at delivering spam to users (therefore all impacting on the user interface), the table includes only *direct* impacts. The generic term "link traversal" includes follow-your-nose browsing, crawling and on-the-fly resource dereferencing. Identity resolution include smushing heuristics.

3.3 Linked Data Infrastructure Aggressions

Spammers must introduce the spam vectors described in Section 3.2 in the Web of Data. Such operation might be achieved either with well-known techniques (e.g. DNS spoofing), or with aggressions specifically tailored to Web of Data actors, from dataset providers to end-user applications. The non-exhaustive list of attacks below describes the latter category of aggressions:

Hub pollution. It consists in registering fake URIs for well-known schemas on hubs such `prefix.cc`, `schemapedia.com`, LOV[20], etc. Polluting services such `sameas.org` or semantic web search engines[21,22] is another realist threat.

[19] `http://tools.ietf.org/html/rfc2397`
[20] `http://labs.mondeca.com/dataset/lov/`
[21] `http://sindice.com`
[22] `http://watson.kmi.open.ac.uk/WatsonWUI`

	Link traversal	Identity resolution	Vocabulary matching	Reasoning	Data discovery	UI
False Labeling						X
Misattribution						X
Schema pollution			X			X
Misleading VoID	X				X	
Malicious subclassing				X		
Presentation knowledge pollution						X
Identity assumption	X	X				
IFP cloning		X		X		
Misdirection	X					
Data URI Embedding						X
Bait and switch						X

Table 1: Spam vectors implications on Linked Data application tasks.

Crawler pollution. Linked Data applications that rely on locally-stored, pre-crawled data might be affected by malicious content fetched from the Linked Data Cloud. Crawlers can be deceived by fetching data from malicious seeds, as spam can be added to the cloud by adding polluted seeds to services such as Ping the Semantic Web[23]. Thus, crawlers requesting a list of recently updated resources, will end up fetching spam.

Malicious RDFa content. As seen in Section 3.1, influencing search engine ranking is a primary goal for spammers, that might stockpile malicious RDFa metadata in pages, to deceive search engine algorithms.

SPARQL injection As for SQL injection, poorly conceived Linked Data applications (or SPARQL client libraries) might suffer from this kind of attack if query sanitation is not implemented.

Gaining control of popular namespaces. Attacks let spammers gain control of popular namespaces, thus introducing malicious content during given time windows only.

4 Linked Data Spam: A Sample Dataset

We provide a sample dataset[24] containing malicious triples that could be used to evaluate the resilience of Linked Data applications or to train spam filters. The dataset is the polluted version of a fraction of the Billion Triple Challenge 2012 Dataset[25]. More specifically, we chose the 1-hop expansion "Timbl crawl", a crawl seeded with Tim Berners Lee's foaf profile, and we applied the spam vectors described in Section 3.2. The resulting dataset contains approximately 16k triples (spam triples account for 4% of dataset size). The dataset includes samples of *Content contamination vectors* and *Link poisoning vectors*. Extract of the datasets are shown in the examples included in Section 3.2.

[23] `http://pingthesemanticweb.com`

[24] `http://www-sop.inria.fr/members/Luca.Costabello/spam`

[25] `http://km.aifb.kit.edu/projects/btc-2012/`

5 Proposed Strategies

We propose three categories of anti-spam techniques for Linked Data:

Techniques that require actions by data creators. The anti-spam process starts from the creator of the data. When data creators follow some simple precautions, they protect their published data from unauthorized modifications. Examples of these precautions:

- **Ensuring data quality.** Spamming in Linked Data is highly related with the wider issue of data quality. For example, the use of "dummy" URIs (for example `http://examples.com/Paris`) can cause harmful problems for data consumers, as spammers might register similar domain names, leading users to malicious sites.
- **Signing data.** Declaring the data author can help a lot in detecting spam. This declaration can be done using the Named Graph data model [5]. Using digital signatures with this declaration prevents attackers from modifying the data or impersonating the author through the identity assumption techniques.

Techniques that require actions during data delivery. Adopted at delivery time, these strategies help detect illegal modifications on data. Some examples of strategies that can be applied at this level:

- **Rating-based techniques for calculating reputation.** Using provenance information combined with ratings from users could help providers detecting the trustworthiness of data sources. Many techniques compute trust values that benefit from provenance information. For example, the EigenTrust Algorithm [11] is a well-known distributed algorithm for calculating trust based on PageRank. Such distributed techniques also help cooperation between data providers and favour the exchange of information about malicious data sources, thus giving cleaner results to users.
- **Link content analysis techniques.** These techniques are used for separating useful web pages from spam in the traditional web by calculating the similarity between the content of the page that contains the link and the refereed page. Such techniques can be used in Linked data to detect spam that uses misdirection spamming techniques.

Techniques that require actions by data consumers. These techniques are used when consumers query or use triples and they are embedded in Linked Data applications. In addition to client-side anti-spamming techniques designed for the traditional web, the following strategies can be used:

- **Subjective Rating-based techniques for calculating trust.** Reputation calculation techniques that we suggested for data providers can only detect obvious malicious behaviour, i.e. universally recognized spam. Content might be interpreted as spam according to the user and the current context whereas in specific contexts, spam might be considered as good content by some users. Thus, mechanisms that calculate subjective and local trust values are more suitable. As an example, "Alignment-Based Trust" [1] is a mechanism for calculating subjective and local

trust values depending on provenance in addition to alignments and user feedback.
- **Context-based heuristic techniques.** These techniques, suggested in [10], rely on triple meta-information (e.g circumstances in which the data was created). This data is used to decide whether to trust the triples. Heuristics include, for example, preferring product information released by the manufacturer rather than the vendor.

6 Conclusion

As the adoption of Linked Data by end user applications becomes a reality, the Semantic Web community begins to face "real world" issues such as performance, dataset reliability, data quality, peak load, etc. Spamming might become a non-trivial problem to tackle. In this paper we analyse the risk of spam in the Web of Data. Having provided a formal definition of spam in the Web of Data, we discuss a taxonomy of Linked Data attacks, showing how spammers might profit from the Web of Data paradigm. We list techniques that spammers could use to pollute Linked Data with undesired content and we provide examples in a spam dataset published online, before proposing strategies to protect linked datasets from spamming attacks. More specifically, we underline the need for further work on trust and provenance in the Web of Data. Future work would target an assessment of the current state of spam in Linked Data, in order to establish a baseline against which any increases in spam can be measured. This may in turn reveal additional spam vectors for which mitigating strategies should be proposed. Implementing the suggested best practises for spam protection will also lead to further analysis and refinements.

References

1. Manuel Atencia, Jérôme Euzenat, Giuseppe Pirrò, and Marie-Christine Rousset. Alignment-Based Trust for Resource Finding in Semantic P2P Networks. In *International Semantic Web Conference (1)*, pages 51–66, 2011.
2. D. Ballou, R. Wang, H. Pazer, and G. Kumar. Modeling Information Manufacturing Systems to Determine Information Product Quality. *Management Science*, pages 462–484, 1998.
3. C. Bizer and R. Cyganiak. Quality-driven Information Filtering Using the WIQA Policy Framework. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(1):1–10, 2009.
4. M. Bobrowski, M. Marré, and D. Yankelevich. A Homogeneous Framework to Measure Data Quality. In *Proceedings of the International Conference on Information Quality (IQ)*, pages 115–124, 1999.
5. Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named Graphs. *Journal of Web Semantics*, 3(3), 2005.
6. J. Dong, H. Cao, P. Liu, and L. Ren. Bayesian Chinese Spam Filter Based on Crossed N-Gram. In *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*, volume 3, pages 103–108. IEEE, 2006.

7. J.R. Gruser, L. Raschid, V. Zadorozhny, and T. Zhan. Learning Response Time for Websources Using Query Feedback and Application in Query Optimization. *The VLDB JournalThe International Journal on Very Large Data Bases*, 9(1):18–37, 2000.
8. Z. Gyongyi and H. Garcia-Molina. Web Spam Taxonomy. In *First international workshop on adversarial information retrieval on the web (AIRWeb 2005)*, 2005.
9. O. Hartig and J. Zhao. Using Web Data Provenance for Quality Assessment. In *Proceedings of the International Workshop on Semantic Web and Provenance Management, Washington DC, USA*, 2009.
10. T. Heath and C. Bizer. Linked data: Evolving the Web Into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1):1–136, 2011.
11. Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.
12. Y.W. Lee, D.M. Strong, B.K. Kahn, and R.Y. Wang. AIMQ: a Methodology for Information Quality Assessment. *Information & Management*, 40(2):133–146, 2002.
13. G. Mishne, D. Carmel, and R. Lempel. Blocking Blog Spam with Language Model Disagreement. In *Proceedings of the first international workshop on adversarial information retrieval on the Web (AIRWeb)*, pages 1–6, 2005.
14. Knud Moller, Michael Hausenblas, Richard Cyganiak, and Gunnar Aastrand Grimnes. Learning from linked open data usage: Patterns & metrics. In *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*, 2010.
15. A. Motro and I. Rakov. Estimating the Quality of Databases. *Flexible Query Answering Systems*, pages 298–307, 1998.
16. F. Naumann. *Quality-Driven Query Answering for Integrated Information Systems*, volume 2261. Springer Verlag, 2002.
17. A.C. Rothwell, L.D. Jagger, W.R. Dennis, and D.R. Clarke. Intelligent SPAM Detection System Using an Updateable Neural Analysis Engine, July 27 2004. US Patent 6,769,016.
18. I. Stuart, S.H. Cha, and C. Tappert. A Neural Network Classifier for Junk e-mail. *Document Analysis Systems VI*, pages 442–450, 2004.
19. D. Trudgian. Spam Classification Using Nearest Neighbour Techniques. *Intelligent Data Engineering and Automated Learning–IDEAL 2004*, pages 578–585, 2004.
20. C.H. Wu. Behavior-based Spam Detection Using a Hybrid Method of Rule-Based Techniques and Neural Networks. *Expert Systems with Applications*, 36(3):4321–4330, 2009.
21. J.A. Zdziarski. *Ending spam: Bayesian Content Filtering and the Art of Statistical Language Classification*. No Starch Press, 2005.