Computing Information Value from RDF Graph Properties

Sinan al-Saffar
Pacific Northwest National Laboratory
Richland, WA 99354
USA
sinan.al-saffar@pnl.gov

Gregory Heileman
University of New Mexico
Albuquerque, NM 87131
USA
heileman@ece.unm.edu

ABSTRACT

Information value has been implicitly utilized and mostly non-subjectively computed in information retrieval (IR) systems. We explicitly define and compute the value of an information piece as a function of two parameters, the first is the potential semantic impact the target information can subjectively have on its recipient's world-knowledge, and the second parameter is trust in the information source. We model these two parameters as properties of RDF graphs. Two graphs are constructed, a target graph representing the semantics of the target body of information and a context graph representing the context of the consumer of that information. We compute information value subjectively as a function of both potential change to the context graph (impact) and the overlap between the two graphs (trust). Graph change is computed as a graph edit distance measuring the dissimilarity between the context graph before and after the learning of the target graph. A particular application of this subjective information valuation is in the construction of a personalized ranking component in Web search engines. Based on our method, we construct a Web re-ranking system that personalizes the information experience for the information-consumer.

Categories and Subject Descriptors

H.1.1 [**Information Systems**]: Models and PrinciplesSystems and Information Theory[Value of Information]

Keywords

Semantic Web, Information Valuation, Semantic Search

1. INTRODUCTION

Information retrieval is a vital component of any information system, indeed the retrieval functionality is really the purpose of an information system. Retrieving information can be thought of as two-phased comprising searching and ranking. In searching several candidate solutions are discovered. For example all the documents indexed with a sought keyword are internally determined in the first phase of a Web search engine. In the second phase, all those candidate documents are examined and ranked before being returned in a prioritized list to the information consumer [7, 10]. The Semantic Web [9] extends the regular Web through enriching textual documents with Semantic markup that makes it easier for software agents to process the intended meaning of natural language. The Semantic Web can be viewed as a bottom up alternative to the earlier AI top-down attempts to extract meaning from natural language [21]. In the Semantic Web some of the burden is shifted to the information producer to clarify their intended meaning by semantically annotating their content which still happens with some automation from assisting software. The difference between AI and the Semantic Web is thus really that of how sophisticated the meaning-producing software needs to be. Regardless of the approach used to create the semantic model, the Semantic Web is comprised of documents whose meaning is processable by software agents. Machine-processable semantic annotations are typically expressed in families of logic-based languages that need to balance a trade-off between expressiveness for efficiency or even decidability. RDF [8] is one widely used such language comprised of binary predicates connecting two uniquely identifiable entities, a subject and an object. The connected entities and connecting predicate can be thought of as two vertices in a graph along with a connecting edge hence a set of these RDF binary predicates can be thought of as a directed graph, an RDF graph.

Web Ranking has thus far been carried out in a system and application-dependent manner whereas at its core, ranking is information valuation plus sorting. We need to formalize a definition and computation for the value of an information piece and simply employ such a formula in an information retrieval system be it a Web, Semantic Web, recommender system, or a closed-world knowledge base. Whether a system retrieves entire documents or answers specific questions is irrelevant to valuating information. These should simply be considered representational matters. Information value is subjective and should be calculated as a function of trust and impact on the information consumer's context [2].

2. BACKGROUND AND RELATED WORK

While the searching stage of Web retrieval has employed term-based similarity methods, ranking has mostly relied on utilizing content relevance through measures such as term frequency-inverse document frequency (tf-idf) [7] and approximating global popularity through link analysis [17, 19]. Enhancements to the rank quality of Web information retrieval have been attempted through capturing more (in addition to the search keywords) of the consumer's context such as preferences of certain subjects and bookmarked pages [22, 15]. Personalized rankings biased to such extended consumer contexts may naturally demand more resources but while this represents a challenge [16, 14], the usefulness of such limited optimization is questionable as they do not provide rank results that are much different than a global document rank [1].

Ranking in the Semantic Web has adapted techniques employed in the "regular" Web such as computing global popularity by analyzing link structure but including labeled RDF links in addition to regular anchor links [13, 20]. Moving beyond textual term matching in RDF meta data, researchers have performed semantic content analysis favoring concepts that are more central or more richly defined in an ontology or properties that are least likely to occur in a knowledge base [5, 4, 18] (these infrequent properties are assumed of provide higher informational content). Some of these methods require the information-consumer to explicitly specify a contextual parameter or two such as the unpredictability of the expected answer or the RDFS or OWL [12] classes to which the sought entities belong [4, 18]. Providing a consumer context is essential for making the ranking subjective and hence increasing information quality as we argued in [2]. An example of one way to map a consumer context using a special kind of graph was given in [3] where graphs were employed to compute a value for an information piece based on the trust in the provenance of the information and the causal impact it may have on it's consumer's context. We expand and refine those ideas in this work using standard RDF graphs only.

3. CREATION OF RDF TARGET AND CONTEXT GRAPHS

Given a target piece of information in natural language, a semantic RDF graph of the information of this document can be created either by the human author semantically annotating the document, or by a software agent or program. We created such a prototype as a Web browser plug-in. (more on this on the later sections). The target information piece is mapped to an RDF graph that represents the meaning of the document semantically. We denote this target RDF graph with G_t. Our goal is to determine the value of this graph subjectively (to a certain information consumer).

The information consumer's preferences and world-view is represented with a set of assertions in RDF triples that also constitutes a graph. We call this the context graph, G_c. The context graph can be created through a software agent from monitoring and interacting with the information consumer and recording preferences. For example a simple context graph can be created from the semantic annotations from all the web pages in the consumers browser bookmark file or through a browser plug-in that extends its functionality with the ability to approve or disprove statements obtained from the semantic annotations on visited pages.
At the present time automated natural language processing abilities are somewhat limited but this is orthogonal to the fact that given the capability of the day to create semantic annotations from natural language, we should base information valuation on solid theoretic foundations assuming a certain mapping accuracy [2]. Information value is subjective. This is why we need a consumer's context. This is not an implementation artifact. This is required in a theoretical definition of information value. The simplest consumer context is a query containing keywords. Such contexts are not differentiating enough to allow personalization of the results and understanding of the consumer needs. A more elaborate context is needed and this has been recognized in attempts to personalize search for example. Ideally all of a consumer's context should be captured. This is difficult and may not be possible but we can assume that as much as possible, a world-view of the information consumer is captured in G_c, the more the technology enables us to make an accurate mapping, the better. But we submit that information value calculation depends on this context and can only improve as the context reflects reality better.

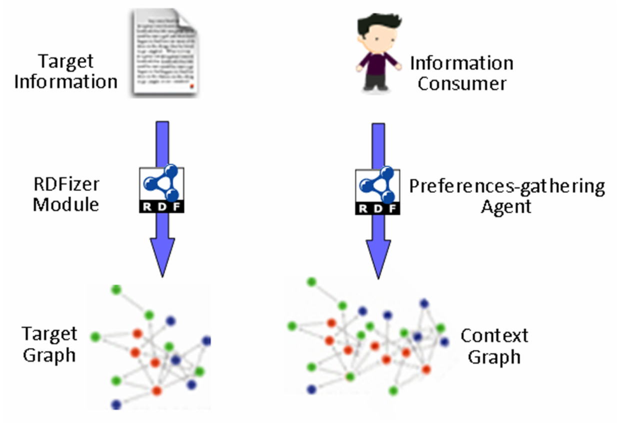

Figure 1: Creation of target and context graphs.

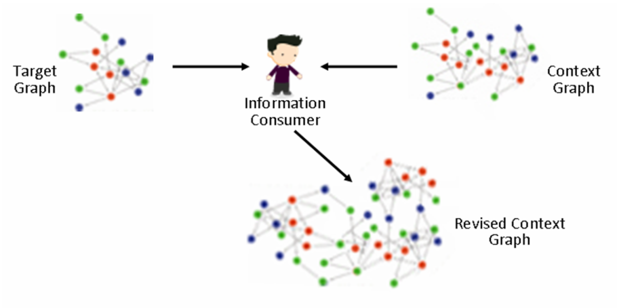

Figure 2: Learning as belief revision

4. SEMANTIC INFORMATION VALUE

Input:
G_t: a target graph represent's the information piece to be valuated.
G_c: a context graph represent's the consumer's context, or world-view.
Output:
$SIV(G_t, G_c) \in [0, 1]$, the Semantic Information Value of the content represented by G_t to the information consumer with a context graph G_c.

4.1 Actual Semantic Information Value

Definition 1. *The Actual Semantic Information Value of a target information piece, I_t is a function of two graphs and is denoted with $SIV_a(G_t, G_c)$. The target graph, G_t is a semantic representation of I_t and the context graph, G_c, is a semantic representation of the information consumer's context:*

$$SIV_a(G_t, G_c) = \Delta G_c \tag{1}$$

Where ΔG_c is the amount of change G_c undergoes as a result of its consumer learning G_t.
The actual value of G_t to a consumer may not be exactly predictable in the real world as the rules that govern the reasoning of the human consuming the information graph, G_t may not be capture-able or even knowable. However, in a closed-world knowledge system where inference rules for automated reasoning have been extracted from an expert, SIV_a may be computed by applying those inference rules to G_c to revise it into a new context graph, $G_{c'}$. Then we compute ΔG_c by employing a suitable similarity (or more accurately dissimilarity in this case) to compare between G_c before the consumer learns G_t and G_c afterwards.
Let $G_{c'}$ denote the revised context graph or the context graph, G_c, after the learning of G_t and let l denote some learning function representing the set of inference rules through which the information consumer may learn from assertions in G_t and generate new assertions to update G_c. then:

$$G_{c'} = l(G_c, G_t) \tag{2}$$

There are two cases where we can compute SIV_a: Either we know l so we can obtain $G_{c'}$ by an automated procedure of reasoning, or we are given $G_{c'}$. In either case we can cast the information value of I_t as a graph dissimilarity problem:

$$SIV_a(G_t, G_c) = \Delta G_c = disSimilarity_a(G_c, G_{c'}) \tag{3}$$

In the literature, the term similarity of two graphs is also used but these terms are really equivalent since:

$$dissimilarity(G_1, G_2) = 1 - similarity(G_1, G_2) \tag{4}$$

The dissimilarity of two graphs has been mostly computed based on either graph edit distance methods or finding the maximum common subgraph which have been shown to be equivalent [11]. For our domain we base the dissimilarity measure on edit methods where we assign costs to different graph modifying operations such as insertions, modification, and deletion and the dissimilarity between two graphs is based on the minimum cost incurred by the series of operations required to make the graphs isomorphic. Note the isomorphism in our case can be solved in linear time as the graphs are labeled so one combination needs to be tried for the matching as opposed to many ($N!$) in the general graph isomorphism problem.

We employ the following distance measure that gives uniform weight to operations thus degenerating into using a count to compute the edit distance for calculating SIV_a:

$$disSimilarity_a(G_c, G_{c'}) = \frac{\mid E_{c'} \mid - \mid E_c \mid}{\mid E_{c'} \mid} \tag{5}$$

Where:
$disSimilarity(G_c, G_{c'}) \in [0, 1]$
$\mid E_c \mid=$ number of edges in the original context graph, G_c, and,
$\mid E_{c'} \mid=$ number of edges in the revised context graph, $G_{c'}$ + the number of edges in G_c that got their labels changed. We assume knowledge does not decrease so relationships between entities change but do not disappear. This is reflected by changes in the graph edge labels instead of deletions. We only use the number of edges as an indicator of graph growth as graphs grow by RDF triple units which is equal to an edge and two vertices so the edges suffice as an indicator. We do not consider it of different importance for a consumer to learn of a new relationship between two already known resources (vertices) or learn of an entirely new resource as we assume such importance would already (by definition) be reflected in the revision of the context graph.

We are using graph size, not structure as a measure of dissimilarity in the case of computing the actual semantic information value as the structure of the old context graph dictated this resulting size change (we do need to use structural properties of G_c when we want to *predict* $G_{c'}$ as we do in the next section). Sensitivity is uniform in computing the actual SIV where $G_{c'}$ is known and the context graph contains entities and properties whose importance is uniform. For example; even if some vertex is more connected than another then we do not assume that changed edges directed into such a vertex will have a higher impact on graph dissimilarity than changes in graph components near a less connected vertices. The reason behind this is that $G_{c'}$ is the graph resulting from applying all the inference to be made and thus all changes that did occur are already reflected in the revised graph, $G_{c'}$. We can not however make this same assumption when we consider sensitivity for SIV_p in the next section as we do not know $G_{c'}$ but predict what it could look like through such properties as connectedness. We considered SIV_a as a reference starting point to explain our idea of information value but in most systems we need to compute SIV_p.

4.2 Predicted Semantic Information Value

In most systems including the Semantic Web where the actual rules governing the reasoning of the information consumer are not known, we can only compute SIV_p , a prediction of the change in G_c. As in the previous section SIV_p is cast as a graph dissimilarity problem except we adjust the sensitivity in assigning costs of the distance by adding weights to new edges according to the ideas of impact and trust. We compute SIV_p based on the potential impact G_t can have on G_c and the consumer's trust in G_t inferred from degree of overlap (agreement) between the two graphs; G_c and G_t.

We are given the target graph G_t representing the semantics of the target information to be evaluated, (I_t). This could be a single statement, a document, or a semantic association. We also know G_c, the semantic RDF graph representing the consumer's context before learning G_t. We first compute $G_{c'}(G_t, G_c)$, as a union of the two graphs. This represents all the possibilities the new knowledge is added to the existing one.

$$G_{c'} = G_t \cup G_c \tag{6}$$

We use G_c and $G_{c'}$ to compute SIV_p similar to computing SIV_a in equation (3) *but* assign weights non-uniformly to the resulting $G_{c'}$ so higher weights get assigned to edges that may increase the size of G_c significantly through un-

known inferences. The structure of the context and target graphs and where they overlap gives us some idea of such possible revisions. One property we use is the degree of connectedness in the overlapping vertices. The intuition is that learning of new unknown (disconnected) concepts is not as important as learning related to already known concepts which in turn is less important than learning concepts that are centrally connected.
$SIV_p =$

$$disSimilarity_p(G_c, G_{c\prime}) = \frac{\sum_{i=1}^{n} w(e\prime_i) - \sum_{i=1}^{m} w(e_i)}{\sum_{i=1}^{n} w(e\prime_i)} \quad (7)$$

One way to assign the weights to reflect the fact that connectedness is likely to result in higher graph growth due to inference can be:
$G_{c\prime} = G_t \cup G_c$
$e\prime_i \in E_{c\prime}$, $e_i \in E_c$
$w(e\prime_{uv}) = t \times degree(v) \Rightarrow u \in G_t, v \in V_c \cap V_t$
$w(e\prime_{uv}) = t \times degree(u) \Rightarrow v \in G_t, u \in V_c \cap V_t$
$w(e\prime_i) = 1 \Rightarrow$ otherwise
$t = Trust(G_t, G_c)$ from equation(8) and is explained next. The justification for the weight assignments in $G_{c\prime}$ above reflects impact and trust. We assign non-uniform costs to edges so that the edit distance will be higher for more valuable graph components. We base the sensitivity in assigning these costs on properties in the RDF graph that reflect the impact G_t can have on its consumer's world-view, G_c and on the trust that consumer has in G_t. We discuss these components:

4.2.1 Trust:

The graph, G_t changes G_c by an amount proportional to the amount of trust the information consumer has in G_t. In our previous work we required the consumer to explicitly (maybe with the help of an assisting application) specify degrees of trust they have in the provenance of the information source of G_t[3]. In this work we employ the idea that trust can be inferred from the piece of information to be evaluated itself represented by G_t. Such an approach is becoming more relevant with the proliferation of the implementation of linked data where additional meta data including provenance is becoming an inferable property and hence eventually becomes part of G_t by definition.

Definition 2. *Trust is the degree of agreement between the target graph, G_t and the consumer's context graph, G_c. We use the following equation to estimate this degree of agreement:*

$$Trust(G_t, G_c) = \frac{\mid E_t \cap E_c \mid}{\mid E_t \mid} \quad (8)$$

Where E_t are the edges in G_t and E_c are edges in G_c. We normalize using $\mid E_t \mid$ to account for the number of assertions made by the target document or information piece to be evaluated. A document stating all the facts in the world is surely to have some that agrees with our context. Thus trust is asymmetric for typically $SIV_{tc} \neq SIV_{ct}$.
We are using the target information piece or document, I_t to be evaluated to deduce trust from the number of assertions already agreed upon between the document being evaluated and the consumer's context. We divide the intersection by the size of G_t as G_t is expected to be smaller than G_c and narrower in domain. If an author made ten statements in a document and we agree with eight of them then our trust in this document is 80 percent. Trust is necessary but not sufficient in deciding the value of a target piece of information. We also need to measure impact.

4.2.2 Impact:

Even if an information piece represented by G_t is surely to be trusted, it is of little value if it is not relevant to its consumer's context. If a consumer's context is domain-focused on, say sculpting, a piece of information about basketball will be of less value to this consumer than a piece of information about an art gallery (the art gallery assertions are more likely to be connected with predicates to the sculpting components G_c). If G_t had absolutely no overlapping vertices (entities) with G_c, then G_c will be unchanged in becoming $G_{c\prime}$ except for the new disconnected component gained from equation (6). We assign the cost of 1 to the added edges of disconnected components in the new context graph, G_c. in other words if a new RDF triple is added to an existing set od RDF triples and there is no overlap between the resources of the two sets, then the cost of the new RDF triple in computing the editing distance is equal to one. In such cases the information consumer learns of a new fact or set of facts about entities they didn't even know existed before learning of G_t. Such new information typically relates to domains of no interest to the information consumer. For example a computer scientists being told that a new green butterfly is discovered in Africa (assuming G_c is accurate so no such concepts were in the context graph to begin with). When a new RDF triple does overlap with a concept known to its consumer, as represented in G_c, it may end up creating other assertions to be added to G_c through an unknown consumer reasoning process or set of rules. A computer scientist learning a new algorithmic bound doesn't just gain that fact but his world-view, G_c get updated through some unknown-to-us set of rules that may add and modify other assertions in G_c to create $G_{c\prime}$. The more new knowledge is connected to existing knowledge the more valuable it is. We assign a cost value to new knowledge proportional to the degree of connectedness of the concepts (vertices) in G_c with which it overlaps. Learning concepts about entities that are highly connected in the original context graph, G_c is most valuable as this has high impact on the consumer's world-view.

Impact is a function of connectedness of a vertex. If many vertices in G_c have directed edges into a certain vertex, v_c overlapping with a vertex, v_t in G_t, this indicates a high likelihood of change in G_c as the document being evaluated may be discussing a central component in the consumer's world-view.

$$Impact(G_t, G_c) = \frac{\sum_{i=1}^{n} d_i}{\sum_{i=1}^{m} d\prime_i} \quad (9)$$

where:
$d_i =$ degree of v_i in G_c and $v_i \in V_c \cap V_t$
$d\prime_i =$ degree of v_i in G_c and $v_i \in V_c$

We have:

$$SIV_p = f(Impact(G_t, G_c), Trust(G_t, G_c)) \quad (10)$$

where f may be different functions to combine trust and impact in different ways. We use multiplication. In equation (7) we computed this SIV_P based on a particular weights assignment that models trust and impact.

Scalability: Since the similarity computation will need to be carried out frequently and the context graph representing the consumer's preferences and world-view gets more useful as it grows in size, a scalable similarity function is critical. We only consider algorithms with an asymptotically linear time complexity in relation to the size of G_c. The summations and set operations above are linear in the size of the input graphs.

When a consumer is presented with G_t, and he believes all of its content, then maybe we are tempted to state that the new context is simply a $G_{c\prime} = G_t \cup G_c$ not biased by new weights. This is not correct. The reason being is that the RDF graphs only deal with assertions in the knowledge base at question. What is missing are the rules. We do not know how a single assertion from G_t will affect G_a. It may not be simply added to $G_{c\prime}$ even if it were fully believed as there may be for example an inference rule that causes the addition of several new assertions or modifications of existing ones. These rules , unlike the case when calculating the actual SIV are, as we discussed, not known and may not be knowable. They may even depend on the mood of the consumer. However we can still estimate the potential impact G_t can have on G_c by assigning more connected vertices higher weights as we did in calculating impact above. The reasoning is that vertices with more connections are likely to propagate any affect on them to the connected objects through the unknown reasoning process. We started with the union operation of G_t and G_c to construct $G_{c\prime}$ but made the sensitivity of the dissimilarity nonuniform and dependent on the connectivity of the vertices by the way we assigned weights to the new components in $G_{c\prime}$. Added graph components on vertices that are more connected increase dissimilarity more than the same components intersect with a less connected area of G_c. We further multiply this calculation by $Trust(G_t, G_c)$ so a $G_{c\prime}$ that would have been very dissimilar to the original G_c due to the many added graph components would be less dissimilar because there is little trust in the assertions as represented by the components being added. We calculate trust as an overlap which can be thought of as a similarity of the two graphs, G_t and G_c hence there is a sweet point or trade-off between similarity (trust) and dissimilarity (impact) to contribute to the calculation of information value but overall after we predict $G_{c\prime}$, we are using its dissimilarity with G_c as the final measure of information value.

5. APPLICATION

We built a system to re-rank Web pages based on their predicted semantic information value, SIV_p. We describe our experiments in this section.

5.1 Front-End Module

The front-end is built as a Web Browser Plug-in: This is a software add-on that extends the user interface and functionality of a Web browser. Our plug-in works with Mozilla Firefox. The plug-in is simply a front-end that sends the URL of the Web page presently displayed in the browser to our back-end server for semantic analysis. The plug-in also has a "semantic bookmark" button to allow the user to add the current page to their context graph, G_c which is maintained along with other user account details on the back-end server. The plug-in also has a log-in button as the system cannot function without identifying the consumer's context graph to be used. The plug-in is implemented with AJAX and XUL. Figure 4 is a screen shot of our plugin shown on the Firefox browser on Linux. After installing the plugin it can be used when you login by clicking the "Biased for button" which displays a window for logging in or registration. Hitting the "Evaluate" button will cause the current page to get analyzed on the back-end server. The result is a semantic value for the page and an optional explanation in the sidebar to the left.

The system enhances current browsing and search experience by providing a semantic value for the page being browse and displaying it in the browser plug-in UI on the tool bar. The plug-in also displays the information value on each link inside the browsed Web page. This serves as a guide for navigation so the consumer can pick the highest rank link if they wish. An explain button opens a sidebar to explain the calculation of the semantic value of the along with semantically identified entities (presently these are entities we could discover in dbpedia).

5.2 Back-End Module

The back-end consists or our application server which we wrote as a Java servlet which gets called from an Apache Tomcat server. Upon the server completing the semantic value computation of the page, it sends it back to the plug-in for display. The message sent back contains three parts, a semantic value for the present page, a semantic value for every page linked to from the present page, and an explanation of the semantic value which is a report that describes the identified entities on the page, G_t along with their RDF class types ad the ones that intersect with the context graph. The servlet receives URL's from front-end plug-in, downloads the text of the page, and creates the target RDF graph, G_t by scanning for terms appearing as literal objects in the rdf:label property for a locally running dbpedia database[6]. We loaded dbpedia in Virtuoso RDF store. We have extracted the labels string URI pairs in a MySQL table for speeding up the lookup process. The back-end server also receives bookmark requests from the plug-in in which case the browsed page is semantically marked but added to the user's context, G_c. The back-end maintains users' accounts and login passwords as well.

5.3 RDF Store and Database

We installed a full version of dbpedia, the semantically annotated Wikipedia on a ducal core server running 64-bit Ubuntu Linux on eight Gigs of RAM. We used MySQL database to cache the text labels to URI mappings, an optimization to speedup the URI lookups. Installing Virtuoso and loading it with dbpedia was not a trivial experience especially that the dbpedia dataset contains more than 100 million triples! The implementation of the system presented many technical challenges. The main shortcoming is on the natural language processing front: we are able to map all the concepts from text to their respective URI's but we usually have difficulty in mapping relationships which end up being the edges in the context graphs. We also implemented a version that uses a Reuters Web service that extracts URI's from text but that too is not very good at extracting properties (a whole RDF triple from text where both the object and subject are in the text itself in addition to the property). This affects how accurately RDF graphs represent the original text, but this is an NLP problem.

6. CONCLUSIONS

In this paper we built on our previous work in defining and computing the semantic value of a target piece of information subjectively as a function of the impact this target information may have on its consumer's context and of the trust that consumer has in the provenance of the target information. Though we had used special graphs as models for implementing these ideas[3], we take our work a step further by simply using RDF graphs as such models.

In this framework we use an RDF graph as a model for the target information to be evaluated and another for the context of its consumer. Since we measure information value as the change in the context graph as a result of the consumer learning the target graph, we compute this value as a dissimilarity between two graphs; the context graph before the consumer learned the target information and the revised context graph afterwards. We predict the revised context graph (and then the potential dissimilarity to the original context) based on the connectedness of the overlapping graph components.

This work points to several directions for improvements. The NLP problem of creating more accurate target and context graphs is still not solved though we argue that it should not stop us from proceeding with the information valuation aspect of this research given a certain capability of mapping natural language to a semantic model. We built a re-ranking systems based on the presented model and method.

Traditional evaluation techniques used in global ranking do not apply to subjective ranking and valuation methods as there is no absolute correct answer for comparison thus human-based survey evaluation techniques may be required.

7. REFERENCES

[1] S. al-Saffar and G. Heileman. Experimental bounds on the usefulness of personalized and topic-sensitive pagerank. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, Washington, DC, USA, 2007. IEEE.

[2] S. al-Saffar and G. Heileman. Semantic-based information valuation. In *IS '08: 4th International IEEE Conference on Intelligent Systems, 2008.*, Varna, Bulgaria, sept. 2008. IEEE.

[3] S. al-Saffar and G. Heileman. Semantic impact graphs for information valuation. In *DocEng '08: Proceeding of the eighth ACM symposium on Document engineering*, New York, NY, USA, 2008. ACM.

[4] B. Aleman-Meza, C. Halaschek-Weiner, I. Arpinar, C. Ramakrishnan, and A. Sheth. Ranking complex relationships on the semantic web. *IEEE Internet Computing*, 9(3):37–44, 2005.

[5] K. Anyanwu, A. Maduko, and A. Sheth. Semrank: ranking complex relationship search results on the semantic web. In *WWW '05: Proceedings of the 14th int. conf. on WWW*, NY, USA, 2005. ACM.

[6] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of 6th Int. Semantic Web Conf.*, November 2008.

[7] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Info. Retrieval.* Addison Wesley, 1999.

[8] D. Beckett. Rdf/xml specification. *http://www.w3.org/TR/rdf-syntax-grammar/.*

[9] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 2001.

[10] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[11] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recogn. Lett.*, 18(9):689–694, 1997.

[12] M. Dean and G. Schreiber. Owl web ontology language reference. *http://www.w3.org/TR/owl-ref/.*

[13] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 652–659, New York, NY, USA, 2004. ACM.

[14] D. Fogaras and B. Racz. Towards scaling fully personalized pagerank. *In WAW*, 2004.

[15] T. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *In Proceedings of the 11th Int. WWW Conf.*, May 2002.

[16] G. Jeh and J. Widom. Scaling personalized web search. In *WWW '03: Proceedings of the 12th international conference on WWW*, pages 271–279, New York, NY, USA, 2003. ACM.

[17] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5), 1999.

[18] Y. Lei, V. Uren, and E. Motta. Semsearch: A search engine for the semantic web. In *Proc. of the 15th Int. Conf. on Knowledge Eng. and Management*, 2006.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *In Technical Report, Stanford University*, 1998.

[20] C. Patel, K. Supekar, Y. Lee, and E. K. Park. Ontokhoj: a semantic web portal for ontology searching, ranking and classification. In *Proceedings of the 5th ACM Int. workshop on Web info. and data management*, pages 58–61. ACM, 2003.

[21] R. H. Richens. Preprogramming for mechanical translation. *Mechanical Translation*, 3(1):20–25, 1956.

[22] J. Teevan, S. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. *In Proceedings of the 28th annual int. ACM SIGIR conf. on Research and development in info. retrieval*, pages 449 – 456, August 2005.

Example

Listing 1: Text (source of the triples in listing 2)

```
A good study of ethics can be found in "
    Beyond Good and Evil" by Frederick
    Nietzsche who influenced many
    philosophers and writers including Jean
    -Paul Sartre. Sartre discusses the
    human condition, ethics, and
    metaphysics in his novels and short
```

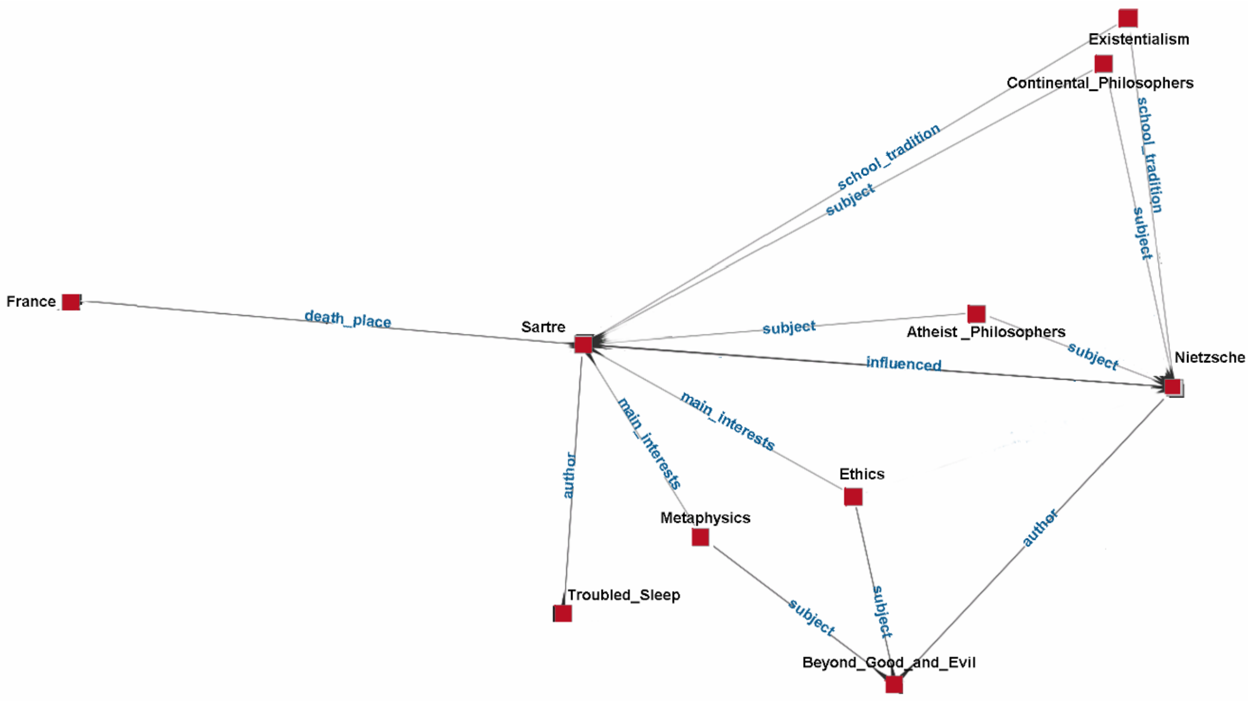

Figure 3: The target RDF graph, G_t, representing the semantics of listing 1

```
    plays such as '"Troubled Sleep". Sartre
     later died in France after
    establishing the school of
    existentialism to which Nietzsche and
    other atheist and continental
    philosophers belonged.
```

Listing 2: RDF Triples of target graph G_t in fig. 3

```
<rdf:RDF>

<rdf:Description rdf:about=d1:Jean-Paul_Sartre>
  <d0:mainInterests rdf:resource=d1:Metaphysics>
  <d0:mainInterests rdf:resource=d1:Ethics>
  <d2:subject rdf:resource=d1:Category:
       Continental_philosophers>
  <d0:schoolTradition rdf:resource=d1:
       Existentialism>
  <d3:deathplace rdf:resource=d1:France>
  <d2:subject rdf:resource=d1:Category:
       Atheist_philosophers>
</rdf:Description>

<rdf:Description rdf:about=d1:Friedrich_Nietzsche>
  <d2:subject rdf:resource=d1:Category:
       Continental_philosophers>
  <d0:influenced rdf:resource=d1:Jean-Paul_Sartre>
  <d0:schoolTradition rdf:resource=d1:
       Existentialism>
  <d2:subject rdf:resource=d1:Category:
       Atheist_philosophers>
</rdf:Description>

<rdf:Description rdf:about=d1:Beyond_Good_and_Evil
    >
  <d3:subject rdf:resource=d1:Ethics>
  <d3:author rdf:resource=d1:Friedrich_Nietzsche>
  <d3:subject rdf:resource=d1:Metaphysics>
</rdf:Description>

<rdf:Description rdf:about=d1:Troubled_Sleep>
  <d3:author rdf:resource=d1:Jean-Paul_Sartre>
</rdf:Description>

</rdf:RDF>
```

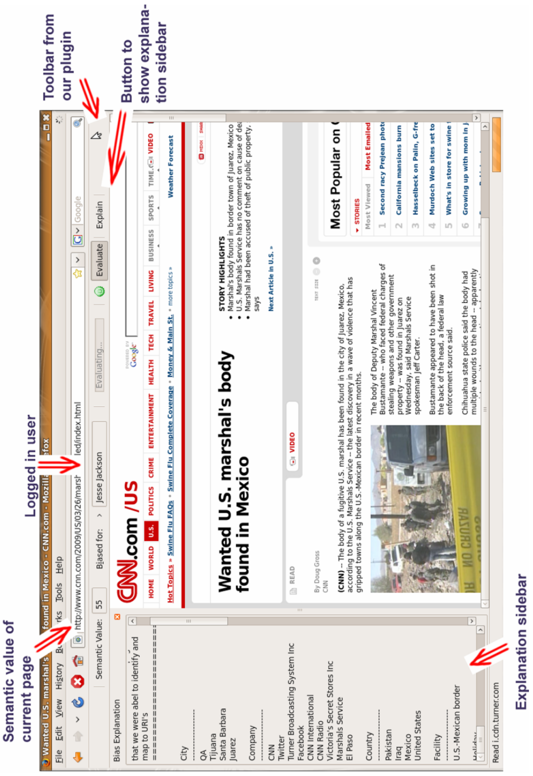
Toolbar from our plugin
Button to show explanation sidebar
Logged in user
Semantic value of current page
Explanation sidebar

Figure 4: A screen shot of our semantic plugin

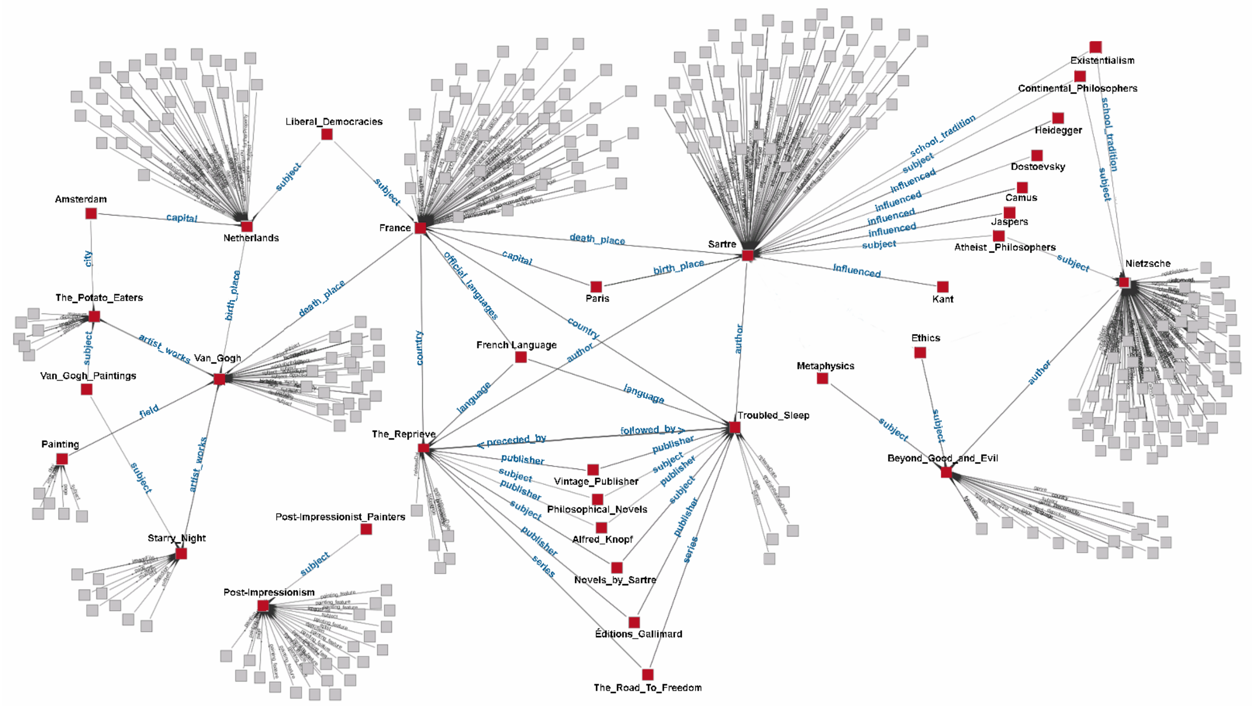

Figure 5: The context graph, G_c, before the consumer learning of G_t

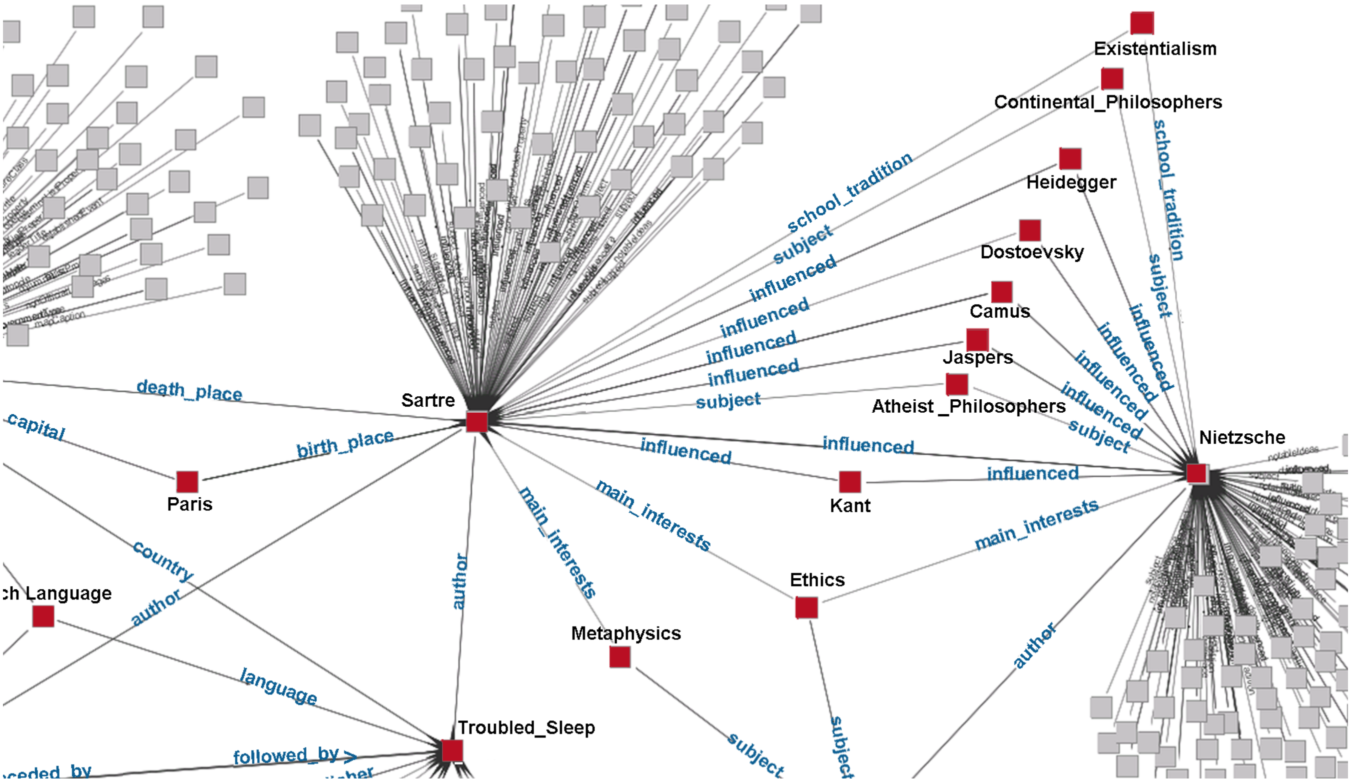

Figure 6: A magnified portion of $G_{c'}$, which is G_c (fig. 5) after learning of G_t (fig. 3)