Annotating the Semantic Web Using Natural Language

Boris Katz Jimmy Lin
MIT Artificial Intelligence Laboratory
200 Technology Square
Cambridge, MA 02139, USA
{boris,jimmylin}@ai.mit.edu

Abstract

Because the ultimate purpose of the Semantic Web is to help users better locate, organize, and process content, we believe that it should be grounded in the information access method humans are most comfortable with—natural language. However, the Resource Description Framework (RDF), the foundation of the Semantic Web, was designed to be easily processed by computers, not humans. To render RDF more friendly to humans, we propose to augment it with natural language annotations, or metadata written in everyday language. We argue that natural language annotations, parsed into computer-readable representations, are not only intuitive and effective, but can also accelerate the pace with which the Semantic Web is being adopted. We believe that our technology can facilitate a happy marriage between natural language technology and the Semantic Web vision.

1 Introduction

The vision of the Semantic Web (Berners-Lee et al., 2001) is to convert existing Web information into a more machine-readable form, with the goal of making the Web more effective for users. This goal grew out of the recognition that a wealth of information readily exists today in electronic form; however, since this information lacks any machine-understandable semantics, it cannot be easily processed by computer systems.

Fundamentally, Semantic Web research is attempting to address the problem of information access: building systems that help users locate, collate, compare, and cross-reference content. As such, we believe that the Semantic Web should be motivated by and grounded in the method of information access most comfortable to users—natural language. Natural language is intuitive, easy to use and rapidly deployable, and requires no specialized training. In our vision, the Semantic Web will be equally accessible by computers using specialized languages and interchange formats, and humans using natural language. The scenario of being able to ask a computer "when was the president of Taiwan born," or "Find me the cheapest vacation package in the Bahamas this month" and getting back "just the right information" is very appealing.

Because the first step to building the Semantic Web is to transform existing sources (stored as HTML pages, in legacy databases, etc.) into a machine-understandable form (i.e., XML/RDF), it is sometimes at odds with a human-based natural language view of the world. Although the general framework of the Semantic Web includes provisions for natural language technology, the actual deployment of such technology remains largely unexplored. In an effort to exploit the synergistic opportunities between the Semantic Web and natural language techniques, we propose three mechanisms for seamlessly integrating natural language technology into the Resource Description Framework. The first involves augmenting RDF property definitions. The second involves creating information access schemata to bridge the gap between language and RDF. The third mechanism proposes further extensions that attempt to mirror human question answering behavior in the form of natural language "query plans." All these mechanisms are based on the concept of natural language annotations, a technique which we have pioneered in the last decade. This technology has already been successfully used in START, the first question answering system available on the World Wide Web, and we believe that it provides a simple mechanism to marry natural language technology and the Semantic Web.

2 Natural Language Annotations

Use of metadata is a common technique for rendering information fragments more tenable to processing by computer systems. Using natural language itself as metadata presents several additional advantages: it preserves human readability, allows for easy querying, and encourages non-expert users to engage in metadata creation. To this end, we have developed natural language annotations (Katz, 1997), which are machine-parsable sentences and phrases that describe the content of various information segments. These annotations serve as metadata to de-

scribe the kinds of questions that a particular piece of knowledge is capable of answering.

To illustrate how this technology works, consider the following paragraph about Joseph Brodsky:

> "For an all-embracing authorship, imbued with clarity of thought and poetic intensity," Joseph Brodsky was awarded the 1987 Nobel Prize in Literature.

This paragraph may be annotated with the following:

> Joseph Brodsky was awarded the Nobel Prize for Literature in 1987.
> 1987 Nobel Prize for Literature

A question answering system would parse these two annotations and store the parsed structures (e.g., ternary expressions (Katz, 1988)) with pointers back to the original information segment. To answer a question, the user query, parsed into the same type of structures, would be compared against the annotations stored in the knowledge base. Because this match would occur at the level of parsed representations, linguistically sophisticated machinery such as synonymy/hyponymy relations, ontologies, and structural transformation rules (e.g., "S-Rules" (Katz, 1997; Katz and Levin, 1988)) could be brought to bear on the matching process. If a match were found, the segment corresponding to the annotation would be returned to the user as the answer. Because sophisticated natural language processing could be invoked in matching questions with annotations, precision far beyond that of standard keyword-based information retrieval techniques could be achieved. In addition, a linguistically-based system allows for variations in user queries, e.g., alternate formulations, active/passive voice, nominalizations, etc. To give a more concrete example, the annotations above would allow a question answering system to answer the following questions (see Figure 1 for an example):

> What prize did Brodsky receive in 1987?
> Who was awarded the Nobel Prize for Literature in 1987?
> Tell me about the winner of the 1987 Nobel Prize for Literature.
> Who was the Nobel Prize for Literature given to in 1987?

An important feature of the annotation concept is that any information segment can be annotated: not only text, but also images, multimedia, database queries, and even procedures.

We have implemented the above technology in START[1] (Katz, 1988; Katz, 1997), the first question answering system available on the World Wide Web. Since it came online in December 1993, START has engaged in exchanges with hundreds of thousands of users all over the world, supplying them with useful knowledge. Currently, our system can answer millions of natural language questions about places (e.g., cities, countries, lakes; coordinates, weather, maps, demographics, political and economic systems), movies (e.g., titles, actors, directors), people (e.g., birth dates, biographies), dictionary definitions, and much, much more.

In order to give START uniform access to semistructured resources on the Web, we have created Omnibase (Katz et al., 2002), a virtual database that integrates numerous heterogeneous Web sources under a single query interface. To actually answer user questions, however, the gap between natural language questions and structured Omnibase queries must be bridged. Natural language annotations serve as the enabling technology that allows the integration of START and Omni-

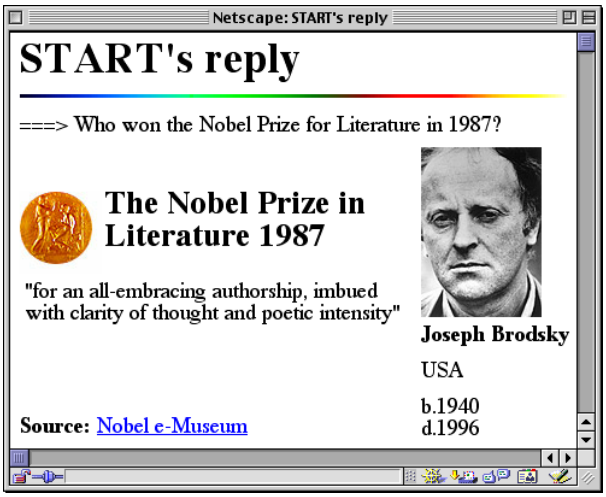

Figure 1: START answering the question "Who won the Nobel Prize for Literature in 1987?"

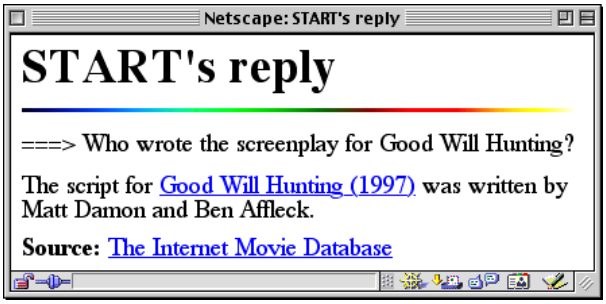

Figure 2: START answering the question "Who wrote the screenplay for Good Will Hunting?" by extracting information from the Internet Movie Database and generating an appropriate response.

[1] http://www.ai.mit.edu/projects/infolab

```
<rdfs:Class ID="Country">
  <rdfs:comment>A Country in the
  CIA Factbook</rdfs:comment>
</rdfs:Class>

<rdf:Property ID="population">
  <rdfs:domain rdf:resource="#Country"/>
  <rdfs:range rdf:resource="xsd:string"/>
  <nl:ann text="Many people live in ?s"/>
  <nl:ann text="population of ?s"/>
  <nl:gen text="The population of ?s is ?o"/>
</rdf:Property>
```

Figure 3: Augmenting an ontology about the CIA World Factbook with natural language annotations.

base. Since annotations can describe arbitrary fragments of knowledge, there is no reason why they can't be employed to describe Omnibase queries. In fact, annotations can be parameterized, that is, they can contain symbols representative of an entire class of objects. For example, the annotation "a person wrote the screenplay for `imdb-movie`" can be attached to an Omnibase query that retrieves the writers for various movies from the Internet Movie Database (IMDb).[2] The symbol `imdb-movie` serves as a placeholder for any one of the hundreds of thousands of movies that IMDb contains information about; when the annotation matches the user question, the actual movie name is instantiated and passed along to Omnibase. After Omnibase fetches the correct answer, START performs additional postprocessing, e.g., natural language generation, to present the answer (see Figure 2).

3 Towards Human-friendly RDF

The Resource Description Framework (RDF) (Lassila and Swick, 1999; Brickley and Guha, 2002), the standardized Semantic Web language for describing metadata, was meant for consumption by computers, not humans. Given this philosophy, how can we be sure that we're creating useful metadata? How can we be sure that our ontologies mirror the way users organize and think about content? Since the final beneficiary of the Semantic Web should be the average user, we advocate a human-centered organization of metadata[3] grounded in natural language.

[2] `http://www.imdb.com/`

[3] It is true that many parts of the Semantic Web will never have any contact with humans, and may be created only for the benefit of software agents, e.g., inventory management systems communicating with warehouses. For these applications, natural language may not be necessary. Nevertheless, a large fraction of the Semantic Web involves end users, where we believe natural language forms the best information access medium.

In 1997, we proposed to attach natural language annotations to everything available on the Web (Katz, 1997). Furthermore, we described a distributed mechanism for knowledge gathering:

> By allowing thousands of people to build up knowledge about knowledge, we will create a knowledge base of an interesting form. The Web will continue to be built out of "opaque" information segments: text, maps, charts, audio, video, *etc.*; but attached to each of these will be natural language annotations that facilitate retrieval. By giving humans access to relevant information that humans can further interpret and understand, we will transform the Web into an intelligent, high performance knowledge base.

The Semantic Web provides many of the mechanisms required to realize this dream. In this paper, we describe three concrete proposals for leveraging Semantic Web research to accomplish our vision: First, we propose to embed natural language annotations directly in RDF property definitions to facilitate language-based querying. Second, we propose the use of information access schemata, an extension of the schemata currently being used by START, to capture patterns of user requests. Third, we propose even more "natural" (but somewhat less powerful) information access schemata that would allow ordinary users to become skillful knowledge engineers.

3.1 Simple Properties

The foundation of the Semantic Web rests on RDF statements, which are essentially triples denoting objects, properties, and values. An alternative and often used description of RDF statements is in terms of "subject," "relation," and "object," revealing a grammatical basis for RDF constructs. In fact, the RDF triples are very similar, both in spirit and in form, to our ternary expression representation of natural language (Katz and Winston, 1982; Katz, 1988). We propose to make this connection more explicit by augmenting `rdf:Property` definitions with natural language annotations.

Figure 3 illustrates our proposal, using a fragment of an ontology representing the CIA World Factbook.[4] Intuitively, the `population` property is a relation connecting a country to its population value. Natural language annotations express this connection concretely in natural language sentences and phrases, via the `nl:ann` property. For example, the phrase "population of `?s`" is linked to every RDF statement involving the `population` property; `?s` is shorthand for indicating the subject (domain) of the

[4] `http://www.cia.gov/cia/publications/factbook/`

relation. From this, a natural language-aware software agent could answer the following English questions without forcing the user to learn and use precisely defined ontological terms:

> How many people live in Kiribati?
> What is the population of the Bahamas?
> Tell me Guam's population.

In addition, the `nl:gen` property specifies a natural language rendition of the knowledge, allowing software agents to present meaningful, natural sounding responses to users.

By "hooking" natural language annotations directly into RDF property definitions, we can not only ensure that our ontologies "make sense" to a user, but also provide natural language question answering and generation capabilities with minimal additional knowledge engineering overhead.

3.2 Information Access Schemata

Despite the simplicity of adding natural language annotations to RDF properties directly, there is a significant restriction to the types of questions that this technique can answer, namely, only one RDF statement can be queried at once. We propose to overcome this limitation by creating schemata that capture similar patterns of information access. For example, consider this "family" of questions:

> What is the country in Africa with the largest area?
> Tell me what Asian country has the highest population density.
> What country in Europe has the lowest infant mortality rate?
> What is the most populated South American country?

We propose to capture this "pattern" of information requests in an information access schema, shown in Figure 4. Here, natural language annotations are employed to describe a pattern of RDF statements.[5] More formally, an information access schema is a quadruple:

- **Annotations:** natural language sentences (either declarative or interrogative) or phrases that describe the types of user questions this schema can answer. These sentences and phrases can contain special symbols that stand in for whole classes of lexical items, e.g., `$country` might stand in for any country in the CIA World Factbook. This allows annotations to be parameterized for greater knowledge coverage.

[5] Because annotations would be processed by linguistically-sophisticated systems, different adjectives such as "highest" and "largest" could be uniformly mapped onto the `maximum` operation.

```
<nl:InformationAccessSchema>

  <nl:ann>what country in $region has the
          largest $attribute</nl:ann>

  <nl:pattern>?x a :Country</nl:pattern>
  <nl:pattern>?x map($attribute) ?val</nl:pattern>
  <nl:pattern>?x :location $region</nl:pattern>

  <nl:action>display(boundto(?x, max(?val)))
  </nl:action>

  <nl:mapping>
    <nl:hash variable="$attribute">
      <nl:map value="population">
      :population
      </nl:map>
      <nl:map value="area">
      :area
      </nl:map>
      ...
    </nl:hash>
  </nl:mapping>

</nl:InformationAccessSchema>
```

Figure 4: An information access schema for superlatives (for simplicity, only one annotation is shown.)

- **Pattern:** a declarative pattern of RDF triples (expressed in N3 (Berners-Lee, 2000) for sake of brevity) that references a pre-existing ontology. The patterns can contain the same special symbols used in the annotations, as well as introduce new unbound variables. Usually, the pattern is written in such a way that when it is satisfied, particular variables would be bound to the answer.

- **Action:** a set of operators to further process variables bound during the pattern matching process. The actions could be as simple as displaying the output, but allows for more complex operations, e.g., aggregation (`count`, `average`, `max`, `min`, etc.), comparison ($<$, $>$, etc.). In order to present users with natural sounding answers, natural language generation may be involved.

- **Mapping:** an optional hash for specifying the bindings between special symbols used in the natural language annotations and RDF resources. For example, the RDF property `:area` might be lexicalized as "land area," "area," or "size"; The `mapping` provides the mechanism for handling this disjunction between lexical and ontological terms.

The schema in Figure 4 could provide answers to questions that involve region-specific superlative comparison of countries in the world. The pattern binds to the value of the particular attribute for countries within the queried geographic region, and the action specifies an aggregate operation (maximum) over the values bound within the pattern. The country corresponding to that maximum value is returned as the answer.[6] The mapping provides a translation from (lexicalized) language attributes to RDF properties. Note that information access schemata are written with respect to a particular pre-existing ontology; for this example, we assume than an appropriate ontology has been established (i.e., `:Country` is defined as a class, and `:location` is defined as a property).

In our vision of the Semantic Web, information access schemata grounded in natural language would co-exist alongside RDF metadata. These schemata could be distributed (e.g., embedded directly into Web pages) or centralized; either way, a software agent would compile these schemata into a question answering system capable of providing natural language information access to users.

Figure 5 provides another example of an information access schema that could allow a linguistically aware software agent to answer the following questions:

> Is Canada's coastline longer than Russia's coastline?
> Which country has the larger population, Germany or Japan?
> Is Nigeria's population bigger than that of South Africa?

Once again, a stylized language annotation describes an RDF knowledge fragment that contains the answer. A specific sequence of commands extracts the information and returns it to the user.

3.3 Going Further

Consider the question "what is the distance from Japan to South Korea?" A reasonable answer would be to compute the distance between their respective capitals. A person faced with this question might first find the capitals of the two countries, and then compute (or look up) the distance between those cities. People generally have no difficulty describing in natural language a "plan" for answering questions that require multiple operations from different sources. Could humans "teach" such plans to a computer directly? Currently, the answer is no, because existing mechanisms of knowledge acquisition require familiarity with precise ontologies,

[6] Although the answer in this schema is simply the country name, it could be couched in a natural language generated sentence for better presentation to the user.

```
<nl:InformationAccessSchema>

  <nl:ann>$country-1's $att is larger
          than $country-2's $att</nl:ann>

  <nl:pattern>?x a :Country</nl:pattern>
  <nl:pattern>?x map($att) ?val-1</nl:pattern>
  <nl:pattern>?y a :Country</nl:pattern>
  <nl:pattern>?y map($att) ?val-2</nl:pattern>

  <nl:action>display(gt(?val-1,?val-2))
  </nl:action>

  <nl:mapping>
    <nl:hash variable="$attribute">
      <nl:map value="population">
      :population
      </nl:map>
      <nl:map value="area">
      :area
      </nl:map>
      ...
    </nl:hash>
  </nl:mapping>

</nl:InformationAccessSchema>
```

Figure 5: An information access schema for comparisons

```
<nl:InformationPlanningSchema>

  <nl:ann>distance between $country1
          and $country2</ann>

  <nl:plan>
    <rdf:Seq>
      <rdf:li>what is the capital of $country1
              := ?capital1</rdf:li>
      <rdf:li>what is the capital of $country2
              := ?capital2</rdf:li>
      <rdf:li>what is the distance between
              ?capital1 and ?capital2
              := ?distance</rdf:li>
    </rdf:Seq>
  </nl:plan>

  <nl:action>display(?distance)</nl:action>

</nl:InformationPlanningSchema>
```

Figure 6: An information planning schema

something that cannot be realistically expected for all users. Despite having plenty of common sense, most users cannot become effective knowledge engineers. We propose to utilize natural language annotations to address this difficulty in imparting knowledge to computers.

We think that "information planning schemata," an extension of the information access schemata technology described in the previous section, can dramatically simplify the task of knowledge engineering. An example of our proposal is shown in Figure 6. Instead of writing RDF patterns, which would require knowledge of domain-specific ontologies, we could use natural language itself to describe the process of answering a question. The answer plan (`nl:plan`) reflects the user's thought process expressed in natural language: first find the capitals of the countries, and then find the distance between those cities.

The above example could answer the following questions:

> How far is the United States from Russia?
> What's the distance between Germany and England?

This method of specifying schemata essentially serves to capture the intuitive thought patterns of a human, and allows ordinary users to "teach" a computer knowledge using natural language.

3.4 Integrating the Three Methods

The three proposed methods for integrating natural language and RDF can be used together to afford greater flexibility. Annotating RDF properties is a low-cost (from a knowledge engineering perspective) way of providing natural language access to RDF statements. Information access schemata, while being more complex and requiring knowledge of domain-specific ontologies, give experienced knowledge engineers fine-grained tools for manipulating RDF and controlling the output. Finally, information planning schemata allow users to describe, in natural language itself, how they would go about answering a particular class of questions. These three methods can combine to provide the foundation for question answering on the Semantic Web.

Ultimately, the actual details of natural language annotations should be hidden from the user behind GUI authoring tools, so that he or she need not come into direct contact with XML or RDF. Additionally, an authoring tool could pre-parse the natural language annotations and store those representations (essentially triples themselves) alongside the annotations.[7] With both natural language and parsed representations at their disposal, software agents would have even greater flexibility in manipulating metadata.

4 Deploying the Semantic Web

We believe that natural language annotations are not only an intuitive and helpful extension to the Semantic Web, but will assist in the deployment and adoption of the Semantic Web itself. The primary barrier to its creation is a classic chicken-and-egg problem: people will not spend extra time marking up their data unless they perceive a value for their efforts, and metadata will not be useful until a "critical mass" has been achieved. Although researchers have been focusing on technology to reduce barriers to entry (via authoring tools, for example), such initiative may not be sufficient to overcome the hurdles. As Hendler (Hendler, 2001) remarks, lowering markup cost isn't enough; for many users, the benefits of the Semantic Web should come for free. Semantic markup should be a by-product of normal computer use and there is no process of metadata creation that is easier and more intuitive than the use of natural language. By divorcing the majority of users from the need to understand formal ontologies and a precisely defined vocabulary, we can dramatically lower barrier-of-entry, easing the transition into the Semantic Web vision.

Our technology of information access schemata provides an annotation system suitable for different levels of user experience. Novices to the Semantic Web merely have to rationally elucidate the process by which they come up with the answer to a particular set of questions, and then describe that plan in a form of stylized language. For more advanced users, the ability to access RDF directly allows finer-tuned control, greater flexibility, and more concise descriptions.[8]

Ultimately, let us not forget that the purpose of the Semantic Web is to benefit humans, not computers. The original idea was that instead of waiting for computers to become smart enough to solve all the problems of understanding human language, we should focus on the slightly less difficult problem of making human data more understandable to computers. To this end, the foundations of the Semantic Web are grounded in language. However, to achieve interoperability and to facilitate interaction between software agents, we've had to sacrifice a lot of human understandability—precise ontologies and formally defined semantics are foreign concepts to the average user. By reintroducing natural language

[7] Without an authoring tool, such a scheme would not be feasible because we cannot expect humans to manually generate parse structures. Note also that keeping natural language annotations makes it possible for them to be re-analyzed later as more powerful parsers become available.

[8] For example, expert users may be able to write schemata that parameterize across RDF properties (the equivalent of verbs in natural language). This may be unnatural for novice users, because it corresponds to higher order logics.

annotations and rendering the connection to human language explicit, we can achieve a satisfying middle ground between computer and human needs.

5 Patterns of Information Requests

We have argued that natural language annotations are a useful and flexible extension to the Semantic Web. But "is it enough?" Specifically, can information access schemata achieve broad enough knowledge coverage to be useful? We believe the answer is yes.

Natural language annotations can serve as more than metadata; they can capture generalized patterns of information access. As shown in the previous sections, our annotations can be parameterized to encompass entire classes of questions. A schema about the CIA Factbook in which the properties and countries are parameterized can answer tens of thousands of potential questions. The cost of writing schemata is not proportional to the number of class instances, but rather the complexity of the class itself. A single schema for the Internet Movie Database, for example, could grant the user natural language access to over three hundred thousand titles!

It is our experience that people ask the same types of questions frequently. Analysis of the questions from the TREC-9 (Voorhees and Tice, 2000) and TREC-2001 (Voorhees, 2001) QA Track, a standardized test set for question answering, reveals a question distribution that qualitatively obeys Zipf's law: a few high frequency query types account for a large portion of all questions (Lin, 2002). Furthermore, questions can often be modeled as database queries (Katz et al., 2002) to simplify access. From these observations, we can conclude that although information access schemata require human effort, they are nevertheless an effective way of achieving broad knowledge coverage at reasonable costs.

6 Future Work

We have described three concrete proposals for making the Semantic Web friendly to computers and humans alike. All the pieces for the implementation of our ideas already exist. START, our natural language question answering system, has demonstrated for nearly a decade that natural language annotations are both useful and effective. Currently, schemata in START can contain actual content (e.g., text and images), procedures (e.g., to access the system clock to tell time), or Omnibase queries (to access heterogeneous data from the Web). Generalizing this technology to the information access schemata we've proposed in Section 3.1 is a relatively straightforward task.

Furthermore, all the machinery necessary to perform complex queries over RDF stores has already been developed by other researchers. For example, the RDF Query Language (RQL) (Karvounarakis et al., 2002) provides a SQL-like query language for accessing large amounts of RDF triples. More concretely, the types of questions handled by annotations in Figure 3 would ultimately translated in a declarative query like:

```
SELECT Y
FROM {X}population{y}
WHERE X = 'Taiwan'
```

In fact, RQL provides a rich set of comparison and aggregation operators necessary to perform complex queries, just like a standard RDBMS.

We are currently collaborating with the Haystack Project (Adar et al., 1999; Huynh et al., 2002), part of MIT's Project Oxygen,[9] to implement the ideas proposed here. Haystack is a personalized information repository built on RDF, and we plan to use it as a testbed for exploring the interactions between natural language technology and the Semantic Web.

In addition to deploying these proposed technologies, we are also researching advanced methods for decomposing complex natural language queries into a series of simpler ones. For example, a question like "When was the president of Russia born," could be broken down into two information seeking steps: first, find out who the president of Russia is, and then find out his or her birthdate. We could write annotations that capture this reasoning directly, but this would require far too many annotations to accommodate every possible combination of properties. Instead, we would like to be able to perform this question decomposition automatically, guided by whatever ontologies are available; for example, we would know that the president of a country is a person, and that a person has a birthdate. With a little bit of search, perhaps we could draw this connection without manual intervention. Note, however, that there are limitations to this approach; the example given in Figure 6 cannot be broken down automatically in this manner because it implicitly captures the heuristic "since one cannot directly calculate the distance between two countries, a reasonable answer is to calculate the distance between their capitals."

7 Conclusion

Just like the development of the Semantic Web itself, early efforts to integrate natural language technology with the Semantic Web will no doubt be slow and incremental. However, we believe that the three mechanisms we've proposed are a step in the right direction. The final goal is truly alluring: an enormous network of knowledge easily accessible by machines and humans alike.

[9] http://oxygen.lcs.mit.edu/

8 Acknowledgements

Thanks to Sue Felshin, David Huynh, Greg Marton, Dennis Quan, and Vineet Sinha for their comments on earlier drafts of this paper. We would also like to thank two anonymous reviewers for their helpful comments. This research is funded by DARPA under contract number F30602-00-1-0545 and administered by the Air Force Research Laboratory. Additional funding is provided by the Oxygen Project.

References

Eytan Adar, David Karger, and Lynn Andrea Stein. 1999. Haystack: Per-user information environments. In *Proceedings of the 1999 Conference on Information and Knowledge Management*.

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American*, 284(5):34–43.

Tim Berners-Lee. 2000. Primer: Getting into RDF and Semantic Web using N3.

Dan Brickley and R.V. Guha. 2002. RDF vocabulary description language 1.0: RDF Schema. W3C Working Draft, World Wide Web Consortium, April.

James Hendler. 2001. Agents and the Semantic Web. *IEEE Intelligent Systems*, 16(2):30–37.

David Huynh, David Karger, and Dennis Quan. 2002. Haystack: A platform for creating, organizing and visualizing information using RDF. In *Proceedings of the Eleventh World Wide Web Conference Semantic Web Workshop*.

Greg Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. 2002. RQL: A declarative query language for RDF. In *Proceedings of the Eleventh International World Wide Web Conference (WWW2002)*.

Boris Katz and Beth Levin. 1988. Exploiting lexical regularities in designing natural language systems. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING '88)*.

Boris Katz and Patrick H. Winston. 1982. Parsing and generating English using commutative transformations. AI Memo 677, MIT Artificial Intelligence Laboratory.

Boris Katz, Jimmy Lin, and Sue Felshin. 2001. Gathering knowledge for a question answering system from heterogeneous information sources. In *Proceedings of the ACL 2000 Workshop on Human Language Technology and Knowledge Management*.

Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran. 2002. Omnibase: Uniform access to heterogeneous data as a component of a natural language question answering system. In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002)*.

Boris Katz. 1988. Using English for indexing and retrieving. In *Proceedings of the 1st RIAO Conference on User-Oriented Content-Based Text and Image Handling (RIAO '88)*.

Boris Katz. 1997. Annotating the World Wide Web using natural language. In *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97)*.

Ora Lassila and Ralph R. Swick. 1999. Resource Description Framework (RDF) model and syntax specification. W3C Recommendation, World Wide Web Consortium, February.

Jimmy J. Lin. 2002. The Web as a resource for question answering: Perspectives and challenges. In *Proceedings of the Third International Conference on Language Resources and Evaluation*.

Ellen M. Voorhees and Dawn M. Tice. 2000. Overview of the TREC-9 question answering track. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

Ellen M. Voorhees. 2001. Overview of the TREC 2001 question answering track. In *Proceedings of the 2001 Text REtrieval Conference (TREC 200)*.