Facilitating Semantic Web Search with Embedded Grammar Tags

Gautham K.Dorai and Yaser Yacoob
Department of Computer Science
University of Maryland
College Park, MD 20742
gauthamt@glue.umd.edu,yaser@umiacs.umd.edu

Abstract

We propose a new framework for intelligent information access. The backbone of this framework consists of embedded grammar tags (EGT's) that capture natural language queries. These embedded grammar tags reflect information content in web pages by anticipating the queries that may be launched by users to retrieve a particular content. These grammars provide a unifying component for speech recognition engines, semantic web page representation and speech output generation. We demonstrate the new EGT representation to enable a software agent to respond to natural speech input from users in narrow domains such as weather, stock market and news queries.

1 Introduction

1.1 Motivation

The explosive expansion in web content has not been accompanied by correspondingly powerful search and content analysis engines. Search engines are hindered by the fact that markup languages were designed for representation of information for human-users and thus this information lacks semantic content that can be interpreted intelligently by search engines, software agents and robots. As a result, extraction of contextual semantic information remains a formidable challenge. Recent efforts to expand markup languages such as DAML, RDF, OIL and XML aim to enhance the effectiveness of content-recovery from web pages by embedding *Tags* that may guide a search engine in uncovering the "meaning" of information [Heflin and Hendler , 2000][DAML 2000][Brickley and Guha, 1999][XML 1998].

Our research focus is on speech-based query of information from the web. Specifically, we envision users employing somewhat constrained natural language sentences to initiate web-queries. This constrained speech can be captured by speech grammars commonly used to enhance the performance of off-the-shelf speech recognition engines. A user's query generates a web access to a page that contains the wanted information. The discovery of relevant content is done by matching the user's query with Embedded Grammar Tags (EGTs) instead of "scalar" tags. Once the desired information has been detected in a web page, a generative grammar that may also correspond to the user's parsed query is used for text-to-speech synthesis.

This model of access to the web is completely hands-free and our ability to achieve it depends greatly, at this point, on narrowing the domain of access so that natural language input can be modeled a priori by grammar. The appeal of this model is that a generative grammar can be employed simultaneously in three tasks (1) constraining the speech recognition engine (2) recognizing the semantic information in a web page (3) determining the sentence composition that the system provides as an output.

While recent research efforts seek to add markup relevant to the content of the web page [Heflin and Hendler , 2000][Abasolo and Gomez, 2000][Trends 2000], we go a step further in embedding the generative grammars in the markup documents. We show that this contributes to enhanced automatic semantic-based recovery of information content while eliminating the increasing reliance on search engines that characterizes the on-going efforts for the semantic web.

1.2 Background

The fact that accurate Natural Language Processing (NLP) [Soderland , 1997][Freitag , 1998] is not yet achievable in general domains has led to numerous efforts to create standardized semantic languages for the web. The semantic web aims to create content which both humans and machines can understand [Heflin and Hendler, 2000]. This is to be achieved by explicitly adding markup to describe the contents of a web-page. A general overview of the markup language layer model for the web is shown in Figure 1.

The HyperText Markup Language (HTML) was the initial language for presentation of documents on the World Wide Web. The main drawback of HTML is its inability to represent content semantically. This led to the Extensible Markup Language (XML) [XML 1998], which allows defining content-specific XML tags. To fully exploit such domain specific tags, there is a need to create semantics that could be understood by all the search engines. This mandated the development of the Resource Description Framework (RDF) [Brickley and Guha, 1999], a standard proposed by the World Wide Web Consortium, to define the web using such domain specific XML tags. The RDF uses *metadata* to explicitly describe document content on the Web. Metadata is basically

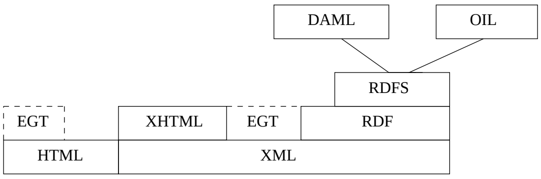


Figure 1: Layer Language Model of the Web

data that is used to describe the information on a web-page.

< RDF:RDF >
< RDF:Description >
<DC:Creator>Eric Miller</DC:Creator>
<DC:Subject>Dublin Core element description</DC:Subject>
<DC:Description>This document is a reference description of the Dublin Core Metadata Element Set designed to facilitate resource discovery.</DC:Description>
<DC:Publisher>Online Computer Library,Inc.</DC:Publisher>
<DC:Format>text/html</DC:Format>
<DC:Type>Technical Report</DC:Type>
</RDF:Description>
</RDF:RDF>

From the above sample we see that RDF uses special tags such as <Description>,<Subject>,<Type> to describe the context of a web documents. Based on the interpretation of such tags, the search engines are able to make queries for information more efficient and accurate. The SHOE language [Heflin, Hendler and Luke, 1999][Heflin and Hendler , 2000] also adds markup to the web-pages by choosing the relevant ontology's vocabulary to describe the concepts on the page. The most recent work in the area of semantic markup is the Darpa Agent Markup Language (DAML) [DAML 2000]. It is being developed as a comprehensive semantic markup language standard to describe web documents. We wish to conclude this brief background overview by providing an example to clearly demonstrate our niche area of semantics for intelligent web information extraction. Consider the following DAML extract generated for a particular user profile (*http://www.cs.umbc.edu/fperic1/damlprofile/*).

< Profile rdf:parseType="Resource" >
< FirstName > Gautham < /FirstName >
< Organization > University of Maryland < /Organization >
< Email > gauthamt@glue.umd.edu < /Email >
< BioSketch > I am presently a Research Assistant at
the University of Maryland doing my Masters degree in the area
of Computer Engineering. I am presently pursuing my thesis
under Professor X. My most prized possession is my black Nissan
Sentra. < /BioSketch >

The above detailed semantics make the document more machine understandable. However, if suppose the software agent wants to find an answer to a question such as 'What car does Gautham have', we would still have to rely on NLP techniques to extract an answer. This is where we propose a semantic language, where we are able to embed 'conversational grammar' as tags into the document. The EGT syntax is defined both as HTML [ISO 1986][Ragget, 1995] and XML [XML 1998] extensions. As such it has its own Document Type Definitions (DTDs) which specify valid tags that can be used. A simple implementation of the above technique would be as shown below. The <ROBOTGRAM-IN> is our own custom designed tag to embed grammar into the content. As mentioned in Section 2, a significant advantage of EGTs over other content description techniques such as RDF, is that it obviates the need for constant revision of schemas to satisfy the rapidly changing web content.

< BioSketch > I am presently a Research Assistant at the University of Maryland doing my Masters degree in the area of Computer Engineering. I am presently pursuing my thesis under Professor X. My most prized possession is my black <ROBOTGRAM-IN> What car does Gautham have </ROBOTGRAM-IN> Nissan Sentra.
< /BioSketch >

2 Proposed Semantics Based Model

2.1 Framework Characteristics

Our proposal for intelligent semantic-web access can be characterized by the following:

- User input is assumed to be via spoken language to the software agent. Therefore, a level of linguistic competence must be achieved using grammars that constrain the interpretation of natural speech. Existing speech recognition engines utilize such grammars to improve performance. A more detailed description of the grammar structure is given in a later section.
- Parsing and generative grammars are used in three stages (1) constraining the speech recognition engine (2) uncovering the semantic information in a web page, and (3) determining the sentence composition that the system provides as an output.
- While the tags currently employed in markup languages such as RDF and XML describe attributes of the information being tagged, the proposed embedded grammars describe queries to which the attached information can be used directly as an answer. This a fundamental departure from existing approaches since it requires the builder of a web page to make it explicit why a particular information item is included on a web page. For example, in the case of a weather page, while a tag such as "temperature" describes the attribute of "47 F," the designer of a web page has to commit to a class of queries that seek to know about "what the temperature is". This

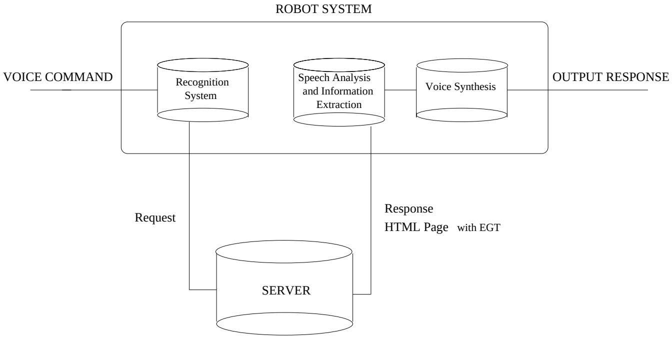


Figure 2: General Structure of Proposed Model

aspect of our model is most readily suitable to domain-specific web pages but may become more challenging as web pages become less structured and more multi-dimensional in content.

- The mediating role of a web search engine for interpreting the embedded semantic information is significantly reduced. Since the embedded grammars already encode the meaning of the user query and the web page was designed with anticipation for an answer search, it is straightforward for the search engine (actually more appropriately, the agent) to recover the desired content. This contrasts with existing approaches that increasingly demand more sophisticated performance from the search-engine in matching the user's query to the embedded tags in a semantically accurate process. It is true that it is not possible to embed all possible ways in which a particular information can be queried. But even in the case of a complicated query, we can restructure the query using existent NLP techniques and then search for a matching grammar. This paradigm shift in the domain for applying NLP is a significant contribution, since applying NLP techniques to the query is much simpler than to the content on the web.
- Another significant feature of the EGT representation is that it is naturally extensible. A web-page creator can independently create new embedded grammar to describe unique information that would be impossible to design as part of a 'closed markup' language. For example, one can embed a grammar that describes a particular novel object , and all that is needed is to anticipate the way user's are likely enquire about it.

2.2 Framework Description

In this section we present a general description of the working of our proposed system. A general overview of our system is shown in Figure 2.

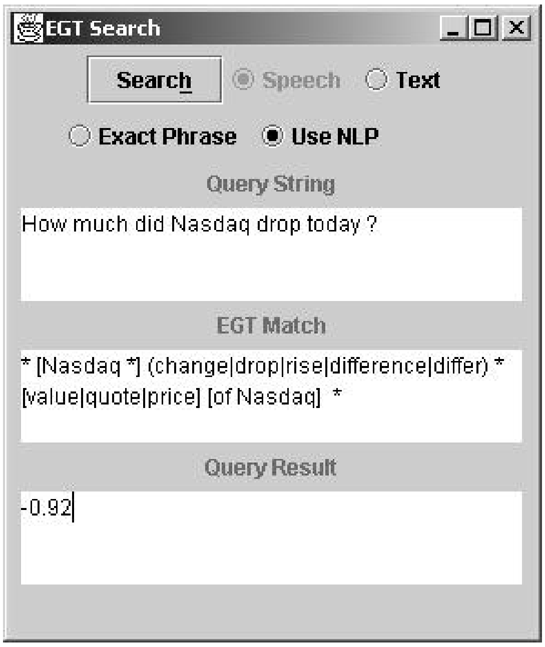
- **Voice Recognition System:** The voice recognition system recognizes the query that the user initiates by employing pre-defined grammars. The system then sends a request to the web server to get the relevant web-pages to search for matching EGT's. We use the Via-Voice engine for speech recognition, along with the Java Speech API (JSAPI).
- **Semantic Analysis and Information Extraction:** Based on the request from the agent, the Server sends back the corresponding document which has been annotated with our <ROBOTGRAM-IN> tags. It then parses the file to find a match for the query that has been requested. If it finds a match, it extracts the corresponding information, and sends it to the speech synthesis system.
- **Voice Synthesis System:** The voice synthesis system employs a text-to-speech generator that speaks out the information extracted from the web-page using a grammar that is dependent on the input query and the embedded grammar.

2.3 Grammar Format

Our proposed framework involves the addition of 'Embedded Grammar Tags' explicitly into the web-content in as general a structure as possible. Such an annotation enables an Agent to utilize the knowledge on the web in a conversational manner. The grammar framework that we employ is derived from the rule-based grammar used by the speech recognition engine. We use the general structure used in the BNF syntax. This is a plain text representation which is similar to traditional BNF grammar [W3C Working Draft2001] used in speech recognition, like the Java Speech Grammar Format (JSGF).

<query> = [please] tell me who [is] the President of USA [is]
*<query> = * [is] the author of the (book|novel|best-seller) Melissa [is]*

In the above specification, words enclosed in square brackets are optional in the query framework, while those in round braces imply that one of the choices must be spoken. A '*' character implies that it can be replaced by any word or combination of words. Thus in the first example, questions such as 'Please tell me who is the President of USA' or 'Tell me who the President of USA is' are both considered as tag matches. Similarly queries such as 'Who is the author of Melissa' or 'Can you please tell me who the author of the best-seller Melissa is', are both EGT matches for the second example. The above format, though general to a certain extent, constrains the grammar that the query language can contain. This can be extended by providing a more hierarchical architecture to the grammar structure as suggested in the



● Speech ○ Text

○ Exact Phrase ● Use NLP

Query String

How much did Nasdaq drop today ?

EGT Match

* [Nasdaq *] (change|drop|rise|difference|differ) * [value|quote|price] [of Nasdaq] *

Query Result

-0.92

Figure 3: EGT Search

W3C voice grammar specification. In the above example, the [please] rule, forces the query parser to recognize only one format of a query. This can be augmented by introducing a package that contains a list of rules, containing the various general formats of requests. The [please] rule can be replaced by such parseable statements, to make it encompass a wide range of typical user requests. An example for the above implementation is given below.

<GRAM-RULE> REQUEST = [please | Can you please tell me | I request you | I seek your help ......] </GRAM-RULE>
<query> = “REQUEST” tell me who [is] the President of USA [is]

2.4 Adding EGT to Web Pages

The information that we actually require is only a very small portion of the web-page. Consider the case of a web-page that tells us the weather of a particular place. In the case of interaction with an agent, the only useful information that is required is the temperature of that particular place and the forecast. All the unnecessary frills that usually accompany the html page for graphics and advertisement can be ignored. The challenge actually lies in identifying and extracting this portion. Consider the following extract from a html page telling us about the weather at a particular location.

<td valign=”bottom”> <span class=”redtemps”> mostly sunny</span> </td>

<span class=”FSnPSmDk”> Temp: </span> <span class=”FSnBSmDk12”> 32 F

We add explicit annotations to the html pages to indicate which portion to extract for a particular query. When the agent gets the response html page based on its query to the server, it tries to match the question to the best extent possible, with the grammar that is embedded in the html page. A simple illustration of adding EGT to the content in the above example is shown below :

<td valign=”bottom”> <span class=”redtemps”> <ROBOTGRAM-IN> [is] the weather [is] at *</ROBOTGRAM-IN> mostly sunny*

We are at present designing a crawler that would parse through web-pages looking for EGT’s and storing it along with the url link in the form of a local database. In this way the EGT search engine can refer this table to find the link to the corresponding response html page.

3 Experimental Section

In this section we illustrate the use of Embedded Grammar Tags to identify and extract responses to queries in natural language, from the world wide web. The client end agent has speech recognition, semantics analysis and speech synthesis software modules integrated in it. An EGT search tool written in Java, is launched to parse the incoming web-page and extract the required response. A screen shot of the EGT search tool is shown in Figure 3.

Speech recognition and synthesis are implemented using the Via Voice engine along with the Java Speech API (JSAPI). The request by the user in the form of conversational grammar, is the query for which the EGT search tool tries to find a match, by parsing the annotated web-page. In order to provide a working demonstration of the model, we download the required web-pages from the corresponding web-sites, annotate them with the EGTs and simulate the system on a local server.

3.1 Annotation with EGT

The annotation of content with EGT is illustrated in the domain of weather and stock market pages. The three steps involved in annotating a particular web-page with EGT are outlined below.

- Identifying the content that a person is likely to query from the web-page.
- Analyzing the various formats in which a particular user can query this information using natural language grammar.
- Embedding grammar tags modeling this query in as general a structure as possible.

Example 1 : Weather

In this example, we consider the web-pages download from the weather site at *www.cnn.com*. We carry out the steps enumerated above. A careful analysis, shows that the information of interest in the web-page are 1) Temperature 2) Humidity 3) Wind Speed 4) Sunrise time 5) Sunset time. EGT's representing the above mentioned information are illustrated below.

<span class=”FSnPSmDk”>Temp:</span> <span class=”FSnBSmDk12”><ROBOTGRAM-IN> * [is] the temperature [is] at College Park</ROBOTGRAM-IN>45 F

<td valign=”bottom”><span class=”redtemps”><ROBOTGRAM-IN> * [is] the weather [is] at College Park</ROBOTGRAM-IN>mostly clear</span></td>

<span class=”FSnPSmDk”>Rel. Humidity:</span> <span class=”FSnBSmDk12”><ROBOTGRAM-IN>* [is] the humidity [is] at College Park</ROBOTGRAM-IN>61

<span class=”FSnPSmDk”>Sunrise:</span> <span class=”FSnBSmDk12”><ROBOTGRAM-IN>* [time] [does|did] [is] the sun [will|would] (rise|rises|rose) * at College Park</ROBOTGRAM-IN>>06:19 am

<span class=”FSnPSmDk”>Sunrise:</span> <span class=”FSnBSmDk12”><ROBOTGRAM-IN>* [time] [does|did] [is] the sun [will|would] (set|sets) * at College Park *</ROBOTGRAM-IN>>06:19 am

<span class=”FSnPSmDk”>Wind:</span> <span class=”FSnBSmDk12”><ROBOTGRAM-IN>* [is] the [wind] (speed|velocity) [of the] [wind] [is] at College Park [is]</ROBOTGRAM-IN>3 mph

Example 2 : Stock Market Queries

Again in this example, we follow the steps previously outlined for annotation with EGT's. The content that a user is likely to query in this case is more limited than in the weather page - 1) Current stock quote 2) Change in the stock quote. EGT annotated samples are shown below.

<td align=”RIGHT” bgcolor=”#DDDDDD”><font face=”arial,helvetica,sans-serif” size=”2” class=”mkcharttxt”><ROBOTGRAM-IN>* [is] [Nasdaq *] [the] (value|quote|price) [of Nasdaq] *</ROBOTGRAM-IN>22.42</font></td>

<td align=”RIGHT” bgcolor=”#DDDDDD”><font face=”arial,helvetica,sans-serif” size=”2” class=”mkcharttxt” ><ROBOTGRAM-IN> * [Nasdaq *] (change|drop|rise|difference|differ) * [value|quote|price] [of Nasdaq] *</ROBOTGRAM-IN>-0.92</font></td>

3.2 Semantic Analysis using EGT

Semantic analysis involves parsing the annotated web-pages to find a match for the query, and extracting the relevant response. This response is fed to the speech recognition system which reads it out using a TTS (Text to Speech) engine. The general grammar format is aimed at encompassing a wide variety of the query language structure used by humans, in normal conversational mode. In the case of a query on the weather at a particular place, the user can ask questions like *“What is the weather at .place.”, “Can you please tell me how the weather is at .place.” or say “Tell me the weather at College Park”* and the parser would be able to find a matching EGT and extract the response with perfect accuracy. Similarly, in our stock value example the user would be able issue queries in formats such as *“How is Nasdaq's price today”,”What is the value of Nasdaq today”,”Tell me the quote of Nasdaq today”*. It is impossible to be able to include all possible query formats that could be used. When we are not able to find a matching EGT, we perform limited NLP on the query. The parser engine then generates different possible query structures with the same meaning, and tries to find a matching EGT. Thus even in the worst case of a highly complicated query, we have transferred the technique of NLP to the query content, rather than to the information on the web-page. Successful experiments on the basis of a large number people who conversed with our system, revealed that almost all the users queried the required information in speech formats that we had included. Even in the case of the few unconventional query formats, our engine successfully extracted the response, by restructuring the query and then finding an EGT match.

4 Future Work

Our current research is focused on creating, evaluating and expanding the utility of EGTs for the semantic web. Specifically, we are exploring

– *Expandable Grammar* We are looking into making the grammar used to model query formats dynamically expandable and capable of learning from a wide variety of user queries. This would also enable authors of web-pages to extend from universally available query packages.

- *Automatic generation of EGTs*. We are developing automatic methods for creation of EGT annotated web-pages. This is an important step of reducing the effort that is, so far, expected from the designer of the web page to exert in spelling-out and expressing the EGTs that are appropriate for the web page content.
- *Different metrics for recognition of EGTs*. We are evaluating the use of different distance metrics to determine the degree of matching between a given query and EGT. Anticipating users to employ unaccounted for query formats, it becomes necessary to find the best match for the query. Classic tools currently employed in NLP such as Hidden Markov Models and Bayesian statistical analysis are being considered.
- *Searching for EGT enabled pages* We are designing a crawler that will parse through web-pages and store a database of EGT queries along with the corresponding links. This would assist the development of a more powerful search engine for user queries.
- *Public use*. We are building a public tool that will provide users the opportunity to employ the EGT technology for speech-based information retrieval from web-sites. This will allow a broad testing of the power of the EGT representation and determine the directions in which it needs to be improved.

5 Conclusion

In this paper a new semantic tagging representation (i.e., EGT) was proposed and developed. The tagging approach is a departure from existing definition and use of tags in XML, RDF and DAML. Employing BNF grammar to represent the queries which users may employ to recover information changes the current view of semantic content of web pages since we reach beyond meaning into anticipation of query syntax and semantics. There are far reaching impacts to this proposal. First, the designer of the web is given the role of anticipating the queries that are matched to particular content items. Second, the web-search engine is relieved from the load of performing NLP since the mapping between queries and content has been already programmed into the page. Third, users can creatively expand the semantic reach of the content of web-pages by simply creating new EGTs that reflect potential queries.

We provided an implementation of our EGTs in the domain of weather and stock market search. We are currently expanding the power of EGTs and refining the means of matching queries to EGTs.

References

[Freitag , 1998] D.Freitag Information Extraction from HTML : Application of a General Machine Learning Approach *American Association for Artificial Intelligence Conference(AAAI-98)*

[Soderland , 1997] S.Soderland Learning to Extract Text-based Information from the World Wide Web. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining(KDD-97)*

[Ciravegna et al , 2000] F.Ciravegna,R.Basili, R.Gaizauskas *Learning to Tag for Information Extraction from Text*. ECAI Workshop on Machine Learning for Information Extraction ECAI2000, Berlin, August 2000.

[Heflin and Hendler , 2000] J.Heflin and J.Hendler. Searching the Web with SHOE. Artificial Intelligence for Web Search. Papers from the AAAI Workshop. WS-00-01. AAAI Press, Menlo Park, CA, 2000. pp. 35-40.

[Heflin and Hendler , 2000] J.Heflin and J.Hendler. Semantic Interoperability on the Web. *Proceedings of Extreme Markup Languages 2000. Graphic Communications Association, 2000. pp. 111-120.*

[Heflin, Hendler and Luke, 1999] J.Heflin, J.Hendler and S.Luke SHOE: *A Knowledge Representation Language for Internet Applications, Technical Report, CS-TR-4078 (UMIACS TR-99-71).* Dept. of Computer Science, University of Maryland (1999)

[Ragget, 1995] D.Ragget HyperText Markup Language Specification Version 3.0. W3C (World-Wide Web Consortium)

[ISO 1986] International Organization for Standardization ISO 8879:1986(E) Information processing – Text and Office Systems – Standard Generalized Markup Language (SGML). First Edition – 1986- 10- 15. [Geneva]

[DAML 2000] www.daml.org.

[Brickley and Guha, 1999] D.Brickley and R.Guha Resource Description Framework Model and Syntax Specification - W3C Recommendation.

[XML 1998] Extensible Markup Language(XML) 1.0- W3C Recommendation

[Trends 2000] The Semantic Web and its Languages Trends and Controversies November/December 2000

[Abasolo and Gomez, 2000] J.M.Abasolo, M.Gomez Melisa, An ontology-based agent for information retrieval in medicine ECDL 2000 Workshop on the Semantic Web

[W3C Working Draft2001] Speech Recognition Grammar Specification for the W3C Speech Interface Framework. W3C Working Draft 2001

[Heflin and Hendler, 2000] J.Heflin and J.Hendler *Dynamic Ontologies on the Web.* Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000). AAAI/MIT Press, Menlo Park, CA, 2000. pp. 443-449.