Semantic Annotation for the Web of Data: An Ontology and RDF based Automated Approach

Che-Yu Yang\textsuperscript{1}, Hua-Yi Lin\textsuperscript{2}

\textsuperscript{1,2}Dept of Information Management, China University of Technology, Taipei, Taiwan, Email: cyyang@cute.edu.tw, calvan.lin@msa.hinet.net, \textsuperscript{3}Corresponding author doi:10.4156/jcit.vol6. issue4.34

Abstract

The construction of metadata which annotates the documents is one of the major tasks for making data understandable to the machine on the Semantic Web. Most of the exiting Semantic Web authoring tools allow user with limited knowledge Semantic Web technologies such as RDF, OWL to markup documents with semantics. These tools help to create the semantic annotations to the documents either during or after composing the documents. The common is these approaches all need human annotator to manually determine the semantic information of the content and then create the metadata which describes this information via tools. This research proposes an approach to bridge the gap between today’s Web and the Semantic Web. The word sense disambiguation technique is utilized to automatically discover the underlying semantic information of the text document, based on the popular open-domain vocabulary ontology - the Wordnet. This semantic information is then annotated to the documents in RDF that conforms to the Semantic Web standards for future reuse. We believe this is one of the feasible ways to bring a large-scale of documents to the Semantic Web.

Keywords: Semantic Web, RDF, Word Sense Disambiguation, Wordnet.

1. Introduction

The Semantic Web, as described by the W3C, is a Web of data — the collection of Semantic Web technologies, e.g. RDF, OWL, SKOS, SPARQL, provides an environment where software application can query the data, make inferences using vocabularies (ontology). To realize the Web of Data, it is essential to have the huge amount of data on the current Web available in a standard format accessible and manageable by the Semantic Web tools. Furthermore, not only does the Semantic Web need access to data, but the relationships among data should be made available too, to create a Web of Data. This collection of interrelated datasets on the Web can also be referred to as Linked Data.

The realization of the Semantic Web requires the widespread availability of semantic annotation for existing and new documents on the Web. The semantic annotation here refers to indicating the relationship between two or more resources by using ontology to express semantic meaning of the relationship. To achieve and create this linked data, technologies should be available for a common format, such as the Resource Definition Framework (RDF), to make either conversion or on-the-fly access to existing data. The RDF is a framework for representing semantic information in the Web. It allows anyone to make statements about any resource using an XML-based syntax.

The creation of the linked data out from the current data in the Web is not an easy task. The fully automated creation of semantic annotations is still an opening issue. Especially for the text data composed of natural language words, an approach that can automatically, effectively and correctly identify and describe the semantic meanings to the text data is definitely needed. This work aims at automating the process of identifying and annotating semantic information to the existing general-domain text data, such as web pages.

To make text data semantically available in RDF, resource that represents vocabulary concepts is needed, precisely in the form of individual URI for each concept. If text data is linked to the vocabulary resource, not only the data can be understood by machine, but further reasoning over data can be achieved through the relationships defined between each vocabulary. This will benefit the management of text data, e.g. indexing, retrieval, extraction, classification, summarization, and so on.

To realize this goal, the following resource and technologies are required:
A resource that contains the conceptual-semantics and lexical relations on general-domain vocabulary which pretty much like a dictionary or vocabulary ontology.

A technology that identifies conceptual-semantics of words in the context of text data. It can be an easy task, however is not really that simple, when there are lots of words that may stand for different meanings when appear in different contexts. A word carrying multiple meanings is called a polysynym, which needs to be disambiguated with its meanings.

A framework to describe/annotate the conceptual-semantics of text data with vocabulary ontology.

To make text data linked data, text analytics must be performed first. Important keywords are retrieved and their semantics are identified corresponding to the concepts in the vocabulary ontology. Word sense disambiguation technology is a feasible way to deal with polysynym, when there is a need to identify one from many possible meanings defined in vocabulary ontology. This relationship between text and its conceptual semantics is needed to be encoded in a standard format accessible and manageable by Semantic Web tools.

2. Related work

To annotate data with semantics in RDF, there are some RDF authoring tools designed to ease the pain of creating RDF instance. Examples are SHOE Knowledge Annotator [1], Annotea [2], SMORE [3], SemanticWord [4], Yawas [5] and GATE [6]. These tools are designed to enable users with limited knowledge of RDF/OWL to markup Web pages using ontology in a Semantic Web compliant fashion. With these tools, authoring linked data is mainly a matter of dragging in data and binding it together through ontology using a graphical interface.

SHOE [1] was one of the earliest systems for adding semantic annotations to web pages. It allows users to mark up pages in SHOE guided by ontologies available locally or via a URL. These marked-up pages can be reasoned about by SHOE-aware tools. The Annotea [2], which uses an RDF schema for describing annotations as metadata, associates text strings to a web document or selected parts of a web document without actually needing to modify the original document. Annotea offers RDF based markup but it does not support information extraction and is not linked to ontology. SMORE [3] is designed to enable users to markup HTML documents in OWL using Web Ontologies. It allows the user to markup web documents with limited knowledge of OWL terms and syntax, and provides a flexible environment to create simple web page simultaneously with markup. SemanticWord [4] is a Microsoft Word-based environment which adds several toolbars to the standard interface which support the creation of semantic annotations in documents and templates according to selected ontologies. The GATE [6] supports discovery and annotation, but there is limited use, reuse and maintenance of the discovered knowledge. OntoAnnotate [7] is an annotation framework for the Semantic Web which includes tools for both manual and semi-automatic pages annotation.

The major tasks of these tools are to compose the data (documents) and add annotations to it. Meanwhile, people still have to manually determine the semantics of the data (according to some ontology), and then add/edit this semantic information to the data with these tools. The semantic information of the data still needs to be extracted or determined by human before annotated by these tools. However, only if there is enough data annotated (linked data), so the full power of a Semantic Web can be received. One of the major approaches for improving the semantic annotation task is information extraction and much can be learned from the semantic tagging of linguistic corpora. In particular, information extraction is, at least to some extent, gaining prominence for automating the formerly purely manual annotation task. There have been efforts to automate some of these tasks using machine learning [8, 9, 10]. These systems try to extract detailed structural data from web pages and require significant training in advance before they can be productive. Furthermore, such systems usually don’t work with a common shared ontology, which plays the key role of the Semantic Web.

PANKOW [11] proposed a method which employs an unsupervised, pattern-based approach to categorize instances with regard to an ontology. The approach is implemented in OntoMat [12], an annotation tool for the Semantic Web. Later in [13] C-PANKOW is proposed which alleviates several shortcomings of PANKOW from various aspects. MnM [14] streamlines the automatic production of semantic annotations by using an information extraction system (IES). It supports: manually annotating web pages (for training the IES), training the IES using the annotated pages, and running the IES to
automatically annotate a set of pages. AKTive Media [15] is an ontology based cross-media annotation (images and text) system. The goal is to automate the process of annotation by suggesting knowledge to the user in an interactive way while the user is annotating and hence minimizing user effort.

Manual annotation is difficult, slow, time-consuming, tedious and costly. These approaches have shown the possibilities to automate the generation of semantic annotations. Unfortunately, the scope of these annotations tends to be restricted to only filling in one information template per document, or need training on large sample in advance. This research tries to bridge the gap between today’s Web and the Semantic Web. An approach to automatically discover the underlying semantic information of the text data is proposed. This semantic information is identified based on a popular open-domain vocabulary ontology – the Wordnet. And then how to annotate this semantic information to the data in the way conforms to the Semantic Web standards is demonstrated. The content this article is organized as follows: the introduction section introduces the issues this research tries to deal with; the related work section reviews the approaches that have been proposed to deal with the issue; then followed by three sections respectively talking about the utilized ontology (Wordnet), a method for discovering conceptual semantics from text data, and how to annotate the data with Wordnet ontology in RDF; and finally a conclusion.

3. Wordnet in RDF and OWL representation

Because meaningful sentences are composed of meaningful words, any computer system that hopes to process natural languages as human do must have the information about words and their meanings. This information is traditionally provided through dictionaries, and digitalized dictionaries are now widely available. But most dictionaries are designed and constructed for the convenience of human readers, not for machines. Fortunately, there are a few machine-readable dictionaries emerged and are developing continuously. One most widely utilized is the Wordnet [16], developed by George A. Miller et al. at Princeton University.

In Wordnet, synonymous words are grouped together into “synonym sets”, or called “synsets” [17]. Each synset represents a single distinct concept. For example, in Wordnet the synset consists of the word forms {car, auto, automobile, machine, motorcar} represents the concept of “4-wheeled motor vehicle; usually propelled by an internal combustion engine”. A synset contains one or more word senses and each word sense belongs to exactly one synset. In turn, each word sense has exactly one word that represents it lexically, and one word can be related to one or more word senses. Synsets are in turn linked by means of conceptual-semantic and lexical relations.

The success of the Semantic Web depends on the availability of ontologies as well as on the proliferation of web pages annotated with metadata conforming to these ontologies. Thus, a crucial problem is the construction of ontologies. The knowledge of vocabulary defined in Wordnet can act as the basis for coding the semantics of data to a relatively detailed degree. The WordNet Task Force [18] is under the Semantic Web Best Practices and Deployment Working Group (SWBPD WG). It aims at providing a standard conversion of WordNet for direct use by Semantic Web application developers [19]. In this RDF/OWL representation of WordNet, the WordNet schema has three main classes: Synset, WordSense and Word, and their subclasses, as shown in table 1.

<table>
<thead>
<tr>
<th>Class</th>
<th>Subclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synset</td>
<td>AdjectiveSynset, AdjectiveSatelliteSynset, AdverbSynset, NounSynset, VerbSynset</td>
</tr>
<tr>
<td>Word</td>
<td>Collocation</td>
</tr>
</tbody>
</table>

Each instance of Synset, WordSense and Word class is assigned with one distinct URI. These URIs are designed in a pattern so that (i) it is easy to determine from the URI the class to which the instance belongs; and (ii) the URI provides some information on the meaning of the entity it represents; and (iii) the pattern produces a unique URI [19]. The pattern for the instance of a Synset is: wn20 instances: + synset- + %lexform%- + %type%- + %sensenr%, where
is the lexical form of the first WordSense of the Synset.

- %type% is one of noun, verb, adjective, adjective satellite and adverb.

- %sensenr% is the number of the WordSense that is contained in the Synset.

For example, the following URI represents a NounSynset that contains a WordSense which is the second sense of the word "bank":

http://www.w3.org/2006/03/wn/wn20/instances/synset-bank-noun-2

The pattern for URIs of WordSenses is similar as for Synsets, except that "synset" is replaced by "wordsense". One example is:

http://www.w3.org/2006/03/wn/wn20/instances/wordsense-bank-noun-1

The pattern for Words is:

wn20 instances: + word- + %lexform%, where

- %lexform% is the lexical form of the Word.

For example:

http://www.w3.org/2006/03/wn/wn20/instances/word-bank

The classes and properties of the schema also have a pattern:

wn20 schema: + %ID%, where

- where the %ID% is the name of the property or class.

For example, the URI for the participleOf property is:

http://www.w3.org/2006/03/wn/wn20/schema/participleOf

Table 2 lists the properties in the WordNet schema [19]. The properties are divided into four categories according to their functions: (i) properties that connect the main classes; (ii) properties that provide data in the form of XML Schema Datatypes; (iii) properties that represent WordNet relations between synsets; and (iv) properties that represent relations between word senses, and finally two superproperties that were introduced for relationship properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Domain</th>
<th>Range</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>containsWordSense</td>
<td>Synset</td>
<td>WordSense</td>
<td>Connect the main classes</td>
</tr>
<tr>
<td>word</td>
<td>WordSense</td>
<td>Word</td>
<td>Provide data in the form of XML Schema Datatypes</td>
</tr>
<tr>
<td>lexicalForm</td>
<td>Word</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>synsetId</td>
<td>Synset</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>tagCount</td>
<td>Synset</td>
<td>xsd:integer</td>
<td></td>
</tr>
<tr>
<td>gloss</td>
<td>Synset</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>frame</td>
<td>VerbWordSense</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>hyponymOf</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>entails</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>similarTo</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>memberMeronymOf</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>substanceMeronymOf</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>partMeronymOf</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>classifiedByTopic</td>
<td>Synset</td>
<td>Synset</td>
<td>Represent WordNet relations between synsets</td>
</tr>
<tr>
<td>classifiedByUsage</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>classifiedByRegion</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>causes</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>sameVerbGroupAs</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>attribute</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>adjectivePertainsTo</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>adverbPertainsTo</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
<tr>
<td>derivationallyRelated</td>
<td>WordSense</td>
<td>WordSense</td>
<td>Represent relations between word senses</td>
</tr>
<tr>
<td>antonymOf</td>
<td>WordSense</td>
<td>WordSense</td>
<td></td>
</tr>
<tr>
<td>seeAlso</td>
<td>WordSense</td>
<td>WordSense</td>
<td></td>
</tr>
<tr>
<td>participleOf</td>
<td>WordSense</td>
<td>WordSense</td>
<td></td>
</tr>
<tr>
<td>classifiedBy</td>
<td>Synset</td>
<td>Synset</td>
<td>Superproperties for relationship properties</td>
</tr>
<tr>
<td>meronymOf</td>
<td>Synset</td>
<td>Synset</td>
<td></td>
</tr>
</tbody>
</table>

With this RDF/OWL representation, Wordnet is now an available ontology to the Semantic Web.
tools, and the URIs for synsets or word senses can be made associated with text data.

4. Use word sense disambiguation to determine meanings of the data

Before we can manipulate the text data with vocabulary ontology at the semantic level, the conceptual semantics of the text data has to be understood first, e.g. what are the major issues in a context or what meaning does a certain word stand for in a sentence. The potential problem of this task is: most of human languages have words that can mean different things in different contexts, such words with multiple meanings are potentially “ambiguous”. For almost all applications of language technology, word sense ambiguity is a potential source of error. “Polysemy” - a single word having more than one meaning; “synonymy” - multiple words having the same meaning, are both important issues in natural language processing or artificial intelligence related fields.

Word Sense Disambiguation (WSD) is the task of figuring out the intended meaning of a word when used in a sentence. In many ways, WSD is similar to part-of-speech tagging. It involves labeling every word in a text with a tag from a pre-specified set of tag possibilities for each word by using features of the context and other information. Human beings are especially sophisticated at WSD. For example, given the sentence “The bank holds the mortgage on my home”, we effortlessly know that the bank here refers to a financial institution that accepts deposits and channels the money into lending activities. Whereas in another context – “He sat on the bank of the river and watched the currents”, the bank here means the sloping land beside a body of water. But unfortunately, it is very difficult for machine to do the same job easily. The machine only recognizes it as an alphabetical string consists of four letters ‘b’, ‘a’, ‘n’ and ‘k’.

The WSD task based on Wordnet ontology can be illustrated as fig. 1. The left side portion in fig. 1 represents a given text data as an input demanding for WSD task. The right side portion in fig. 1 represents the Wordnet Ontology, where a node represents a conceptual semantics (synset), and the links between nodes represents the semantic relations between concepts. The WSD task tries to figure out the intended meaning of a word (as many of them are possibly polysemy) from various possible candidates defined in Wordnet.

![Figure 1. The WSD task against Wordnet – to generate mappings between word forms and synsets.](image)

The research on WSD has been one of the most difficult issues in computational linguistics for a long while. Recent advances benefit from machine learning techniques, sophisticated sense inventories, e.g. WordNet [20], and large corpora to find relevant linguistic features. Generally the WSD can be divided into two categories: supervised approach and unsupervised approach.

Supervised approaches [21, 22, 23, 24, 25], which learn from correctly sense-annotated corpora, achieve better performance; however, they are highly coordinated to the training corpora, and need large amount of high quality annotated data to achieve reliable results. Moreover, to create large sense annotated corpora is a time consuming and labor-intensive job, and sometimes the judgment is subjective and thus may be different by individuals. On the other hand, unsupervised approaches [26, 27, 28, 29, 30, 31] have the advantage of making judgment without the need for training and thus can be adopted immediately. However, presently they tend to perform less well than the former. Some recent works [32, 33, 34] also report positive results on WSD based on concept similarity measurement.
The WSD task has been an important field in natural language processing (NLP) for a long time [35], and has been conceived as a fundamental task of other NLP tasks such as machine translation, information retrieval, text summarization, question answering, … et al. And now in this research, the WSD is utilized to help the construction of linked data for the Semantic Web, in the way —

— use WSD to figure out the conceptual semantics of the text data, that is to create the mapping between a word and a Wordnet synset. And;
— use RDF to annotate this semantic information (the mapping) in the format that accessible to the Semantic Web tools. The annotation approach will be demonstrated in the later in this article.

5. Annotate the meanings of data with RDF

The semantic information of text data can be extracted by the task of WSD so the “interests” of the content can be known now. The next issue is: how to annotate these interests to the text data in a form available to the machine, namely the Semantic Web tools, for further utilization such as query or inference.

The Resource Description Framework (RDF) is a language for representing the information about resources in the World Wide Web [36]; RDF is recommended by the W3C and is designed to represent information in a minimally constraining and maximally flexible way, thus offers greater value from sharing. The value of information increases as it becomes accessible to more applications across the entire Internet. To facilitate data operation at Internet scale, RDF is an open-world framework that allows anyone to make statements about any resource. In RDF, the structure of a statement is a collection of “triples” consisting of a “subject”, a “predicate” and an “object” [37]. The assertion of an RDF triple indicates that some relationship (represented as the “predicate”) holds between the things denoted by “subject” and “object” of the triple. Such a RDF statement can be illustrated by a node and directed-arc graph, where each triple is represented as a node-arc-node link, as shown in fig. 2. Such a graph is called an RDF graph, where:

— Subject: the resource the statement describes
— Predicate: a specific property of the resource the statement describes
— Object: the value of this property for the resource the statement describes

![Figure 2. RDF graph data model to make statements about any resource.](image)

A node may be (i) a Uniform Resource Identifier (URI) with optional fragment identifier (URI reference, or URIref); (ii) constant values (called literals) represented by character strings. A URI is a compact sequence of characters that identifies an abstract or physical resource. Properties are URI references. A URI reference or literal used as a node identifies what that node represents. A URI reference used as a predicate identifies a relationship between the things represented by the nodes it connects. A predicate URI reference may also be a node in the graph.

The RDF graph data model can be used to annotate text data with their conceptual semantics. This annotation can be made on a “document basis” if the operation to the text data is on a document-basis, such as a PDF file, a MS Word file, a Web page, a text file, et al. Thus a document can correspond to a “subject”; a word sense (synset) in Wordnet can correspond to an “object”. As to the “predicate” which indicates the relationship between “a document” and “a synset”, should be qualified to reflect the idea of “talks about the concept in its context” or “mentions to the concept in its context”. The RDF graph of conceptual semantics annotation to the text data is defined as fig. 3.

![Figure 3. Annotate a (text) document with semantic information in RDF.](image)
As to the feasible choice for the “predicate” in the above model, needs to hold the idea of “talks about/mentions to the concept”. Since literals may not be used as subjects or predicates in RDF statements [37], some kind of metadata collection that is popular and forms to the Semantic Web standard must be chosen. The Dublin Core Metadata standard is a simple yet effective element set for describing a wide range of networked resources. It has become one of most popular vocabularies for use with RDF during the past years, more recently in the context of the linked data movement [38]. The idea of Dublin Core is the "core metadata" for simple and generic resource descriptions [39], provides a simple standard to facilitate the finding, sharing and management of information.

The Dublin Core Metadata Element Set is a vocabulary of fifteen properties for use in resource description. It has been formally endorsed in the standards of ISO Standard 15836:2009, ANSI/NISO Standard Z39.85 and IETF RFC 5013 [40]. The fifteen properties are namely: contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title and type. Among the fifteen metadata elements, the term “subject” is the most proper here to represent the idea of “talks about/mentions to the concept.” Table 3 shows the definition of the metadata term “subject”.

<table>
<thead>
<tr>
<th>Table 3. Term “subject” in Dublin Core Metadata Element Set</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URI:</strong> <a href="http://purl.org/dc/elements/1.1/subject">http://purl.org/dc/elements/1.1/subject</a></td>
</tr>
<tr>
<td><strong>Label:</strong> Subject</td>
</tr>
<tr>
<td><strong>Definition:</strong> The topic of the content of the resource.</td>
</tr>
<tr>
<td><strong>Comment:</strong> Typically, a Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary.</td>
</tr>
</tbody>
</table>

With the RDF/OWL representation of WordNet, each Synset that represents a concept becomes a resource in the notions of URI available to the Semantic Web applications. Together with the semantic information extracted from the text data file through WSD task and the term “subject” in Dublin Core, it is now possible to encode the semantic information into RDF graph data model. For example, to assert the statement: “http://about.com/loan.html” mentions to the concepts “bank” (the 2nd sense of the word bank - financial institution), “money” (the 1st sense of the word money - medium of exchange) and “check” (the 1st sense of the word check - a written order directing a bank to pay money), the RDF graph for the assertion would look like fig. 4.

Figure 4. An example RDF graph indicating a document refers to three concepts
In this way a document (text) is just like a bag filled with “concepts” in it. Furthermore machines can know what a document is about and can access to the document in a way conforming to the Semantic Web standards.

Furthermore, by given the XML qualified name prefixes, defined as follows:

- prefix “rdf:”, namespace URI: “http://www.w3.org/1999/02/22-rdf-syntax-ns#”
- prefix “dc:”, namespace URI: "http://purl.org/dc/elements/1.1/"
- prefix “wn:”, namespace URI: "www.w3.org/2006/03/wn/wn20/instance/"
- prefix “ab:”, namespace URI: "http://about.com/"

The RDF graph in fig. 4 can be written in another form of RDF statements, as follows:

<table>
<thead>
<tr>
<th>ab:loan.html</th>
<th>dc:subject</th>
<th>_:z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rdf:type</td>
<td>rdf:Bag</td>
</tr>
<tr>
<td></td>
<td>_:z</td>
<td>rdf:_1</td>
</tr>
<tr>
<td></td>
<td>_:z</td>
<td>rdf:_2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wn: synset-money-noun-1.</td>
</tr>
<tr>
<td></td>
<td>_:z</td>
<td>rdf:_3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wn: synset-check-noun-1.</td>
</tr>
</tbody>
</table>

The above form is called “triples notion”, which is intended as a shorthand notation.

RDF also provides an XML syntax for writing down and exchanging RDF graphs, called RDF/XML. Unlike triples, RDF/XML is the normative syntax for writing RDF. The RDF graph in fig. 4 can be written in RDF/XML as follows:

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
<rdf:Description rdf:about="http://about.com/loan.html">  
  <dc:subject>  
    <rdf:Bag>  
      <rdf:li rdf:resource="www.w3.org/2006/03/wn/wn20/instance/synset-bank-noun-2"/>  
      <rdf:li rdf:resource="www.w3.org/2006/03/wn/wn20/instance/synset-money-noun-1"/>  
      <rdf:li rdf:resource="www.w3.org/2006/03/wn/wn20/instance/synset-check-noun-1"/>  
    </rdf:Bag>  
  </dc:subject>  
</rdf:Description>
</rdf:RDF>
```

Finally, the process flow of the construction of linked data from text data is shown in fig. 5.

---

Figure 5. The construction flow of linked data from text data.
6. Conclusion and future work

The current status of the Semantic Web is that there is not much of a Semantic Web data (linked) due to the lack of annotated web pages. There is such a lack because when annotating documents, the information providers have to determine how to annotate the data, including which part of data should be annotated, and what kind of vocabulary (ontology) should be used, or even create their own ontologies.

In this work an automated approach to semantics extraction and annotation to text data is proposed. Word sense disambiguation technique is used to identify the concepts in content against Wordnet ontology. RDF is used to annotate the semantics. In this fashion, much of the existing data on the Web can be processed and brought to the Semantic Web.

However, how to query and search for data based on these semantic annotations would be a great topic for future research. Furthermore, to make inference from the ontology to help the searching is also an important issue, which involves the share of semantic annotations between different ontologies.

7. References


