KATHMANDU UNIVERSITY JOURNAL OF SCIENCE, ENGINEERING AND TECHNOLOGY
VOL. 8, No. I, FEBRUARY, 2012, 125- 133

SURVEY PAPER ON "WEB PAGE CONTENT VISUALIZATION"

Sushil Shrestha

M.Tech IT, Department of Computer Science and Engineering, Kathmandu University
Faculty, Department of Civil and Geomatics Engineering, Kathmandu University

Corresponding address: sushil@ku.edu.np
Received 20 February, 2011; Revised 15 September, 2011

ABSTRACT
The paper is a survey of a work done in the field of web page content vuisualization. The paper start with summary of the semantic graph. Then it describe the process of generation of semantic graph by natural language processing and machine learning techniques and then enriching text with RDF/OWL Encoded Sense as the enhancement of the existing enrycher text conversion and ends with the possible future direction and conclusion.

Keywords: Natural language processing , semantic graph, web page, data, triplet, visualization.

INTRODUCTION
The use of a graph for the visualization of information has the advantage that it can capture a detailed knowledge structure. Therefore graphs are suitable for conveying semantic relations between individual information items and for providing an understanding of the overall information structure. Semantic graph is a network of heterogeneous nodes and links [6]. It can also be considered as a database schema of a relational databases. Semantic graph attributes describe properties of the graph [4]. In reality, a semantic graph can contain billions of nodes and links in the graph repository for querying. This kind of graph information is usually noisy and loaded with unknown and/or incomplete information. From the beginning of the internet, the continuing progress in network technologies and data storage techniques has digitalized huge amounts of documents on the internet [7]. A tool introduced by Rastier [1], semantic graph can be used to represent any semantic structure in terms of semes and the relations between them. Semes are the nodes of semantic graph (shown in boxes or brackets) and the relations are the links (shown in ellipses or parentheses). The arrows indicate the direction of the relation between nodes. This paper describe a method of text analysis with the stated purpose of extracting valuable information from documents. The graph is based on triplets retrieved from the document sentences. Moreover, it also contains a description of an application of semantic graphs generation – text summarization – as a method for reducing the quantity of information but preserving one important characteristic i.e its quality. The accessibility of information arises mostly from the rapid development of the World Wide Web and online information services. One has to read a considerable amount of relevant content in order to stay updated but it is impossible to read everything related to a certain topic. A feasible solution to this admitted problem is condensing this vast amount of data and extracting only the essence of the message in the form of an automatically generated summary. So according to [5] w e think that a tool which could render most of the text contained in web pages which is easier to grasp would greatly

improve the user experience and decrease the time needed to get at least a first impression of the page content. Such a tool should have the following features:-

- Provide a graphical representation of text as most people prefer pictures over text.
- Emphasize and display the most important content (e.g., Text summary).
- The user should be able to adjust the amount of information s/he wants to see.
- The user should be able to see the content of the summary in the order in which s/he would see it during a normal read of the page.
- The tool should be available and easy to use for the most users on the internet and be applicable to most of the web pages.

MATERIALS AND METHODS

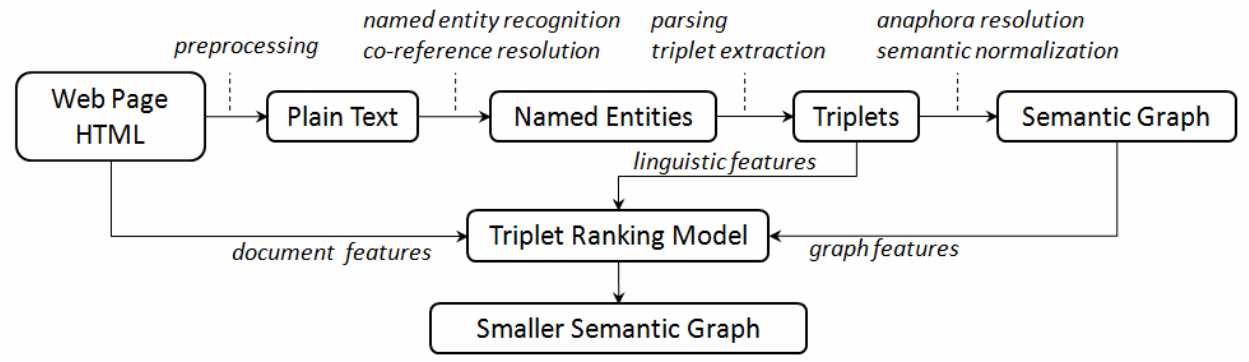


Figure 1 System overview.

According to [8], the process underlying the proposed framework consists of several phases, each depending on the output of the previous one. At first the data structures (paragraphs, titles) are split into sentences and then later into individual words. The second part of the enrichment model is the annotation section, containing information on individual objects that have been detected in the document one way or another, for instance, named entities and other semantic graph nodes. The annotations themselves contain a list of their instantiations within the document and a set of associated semantic attributes meant for describing them and linking them to an existing ontology. The third part of the enrichment model is the assertions section, containing the triplets that construct the semantic graph. This represents the individual information fragments that were extracted from the plain text and form the basis for new knowledge. Furthermore, each document contains a document metadata section storing attributes that apply to the document as a whole, such as categories, descriptive keywords and summaries.

The language-level processing step identify possible entities, so now the entity level processing consolidate the identified entities. This is done with anaphora resolution where pronoun mentions are merged with literal mentions, co-reference resolution that merges similar literal mentions and entity resolution which links the in-text entities to ontology concepts. Since entity extraction is often handled with several domain-specific extractors, the

purpose of entity level processing is to allow multiple extraction mechanisms and consolidate their output into a coherent set of entities and if possible linking them to ontology concepts. Named entities are identified which refer to names of people , locations and organizations yielding semantic information from the input text [5]. Now for the named entity recognnition GATE is considered (*General Architecture for Text Engineering*) [11] which is used as a toolkit for natural language processing. For people, gender can also be stored to minimize the amount of search space whereas for locations, names of cities and of countries can be stored which enable co reference resolution that implies identifying terms that refer to the same entity.

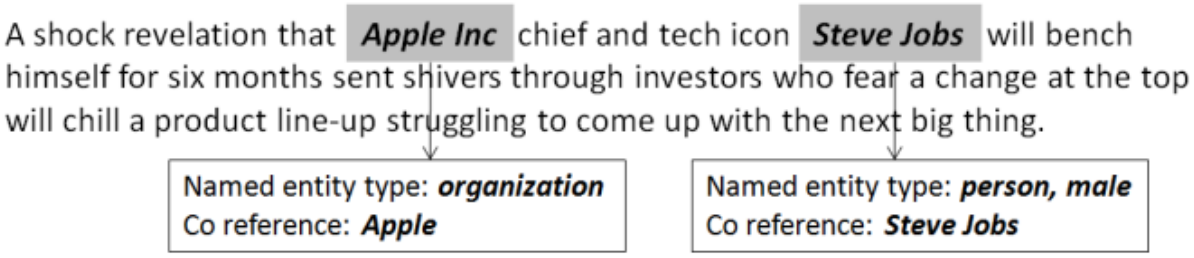


Figure 2 A document excerpts with two annotated named entities (an organization and a person).

Co-reference is defined as the identification of surface terms (words within the document) that refer to the same entity [12]. In the case of named entities composed of more than one word, the set of English stop words is eliminated (for example Ms., Inc. and so on). Heuristic methods can be applied if the two different surface forms represent the same named entity if one surface form is completely included in the other. For example, *"Clarence"*, *"Clarence Thomas"* and *"Mr. Thomas"* refer to the same named entity, that is, *"Clarence Thomas"*. Moreover, abbreviations are also co-referenced, for example *"U.S."*, *"U.S.A."*, *"United States"* and *"United States of America"* all refer to the same named entity – *"United States America"* (*"of"* will be eliminated, as it is a stop word).

After the named entities extraction, triplets are generetaed by parsing triplet extraction method. The triplet is a semantic structure composed of a subject, a verb and an object. This structure is meant to capture the meaning of a sentence. There are two approaches to triplet extraction both of which take as input a sentence with tokens with their part of speech. One of the method to extract the triplets is tree bank parser [9]. A treebank is a text corpus where each sentence belonging to the corpus has a syntactic structure added to it. Because of the common outputted parse tree of Stanford Parser and OpenNLP, the similar algorithm for triplet extraction for the two parsers is developed.

A sentence (S) is represented by the parser as a tree having three children: a noun phrase (NP), a verbal phrase (VP) and the full stop (.). The root of the tree will be S. Firstly we intend to find the subject of the sentence. In order to find it, we are going to search in the NP

subtree. The subject will be found by performing breadth first search and selecting the first descendent of NP that is a noun. Nouns are found in the following sub trees:

Sub tree	**Type of noun found**
NN	noun, common, singular or mass
NNP	noun, proper, singular
NNPS	noun, proper, plural
NNS	noun, common, plural

Secondly, for determining the predicate of the sentence, a search will be performed in the VP sub tree. The deepest verb descendent of the verb phrase will give the second element of the triplet. Verbs are found in the following sub trees:

Sub tree	**Type of verb found**
VB	verb, base form
VBD	verb, past tense
VBG	verb, present participle or gerund
VBN	verb, past participle
VBP	verb, present tense, not 3rd person singular
VBZ	verb, present tense, 3rd person singular

Thirdly, we look for objects. These can be found in three different sub trees, all siblings of the VP sub tree containing the predicate. The sub trees are: PP (prepositional phrase), NP and ADJP (adjective phrase). In NP and PP, we search for the first noun, while in ADJP, we find the first adjective. Adjectives are found in the following sub trees:

Sub tree	**Type of adjective found**
JJ	adjective or numeral, ordinal
JJR	adjective, comparative
JJS	adjective, superlative

```
function TRIPLET-EXTRACTION(sentence)  returns a solution, or failure
result← EXTRACT-SUBJECT(NP_subtree) ∪ EXTRACT-PREDICATE(VP_subtree)
        ∪ EXTRACT-OBJECT(VP_siblings)
        if result ≠ failure then return result
        else return failure

function EXTRACT-ATTRIBUTES(word) returns a solution, or failure
// search among the word's siblings
        if adjective(word)
                result← all RB siblings
        else
                if  noun(word)
                        result ← all DT, PRP, POS, JJ, CD, ADJP, QP, NP siblings
                else
```

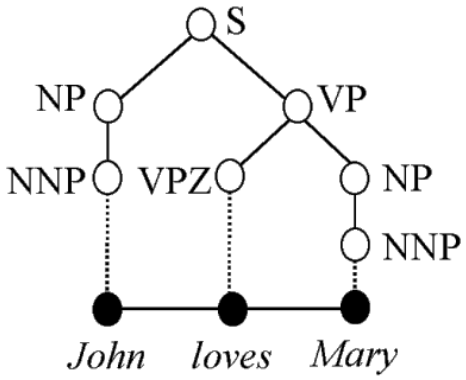
```
                        if  verb(word)
                                result ← all ADVP siblings

 // search among the word's uncles
if  noun(word) or adjective(word)
                if  uncle = PP
                        result←uncle  subtree
          else
                  if  verb(word) and (uncle = verb)
                        result←uncle subtree
if result ≠ failure then return result
else return failure

function EXTRACT-SUBJECT(NP_subtree)  returns a solution, or failure
        subject← first noun found in NP_subtree
         subjectAttributes ← EXTRACT-ATTRIBUTES(subject)
        result←subject ∪ subjectAttributes
        if result ≠ failure then return result
        else return failure

function EXTRACT-PREDICATE(VP_subtree)  returns a solution, or failure
        predicate← deepest verb found in VP_subtree
        predicateAttributes ← EXTRACT-ATTRIBUTES(predicate)
        result←predicate ∪ predicateAttributes
        if result ≠ failure then return result
        else return failure

function EXTRACT-OBJECT(VP_sbtree)  returns a solution, or failure
        siblings← find NP, PP and ADJP siblings of  VP_subtree
        for each value in siblings do
        if value = NP or PP
                object← first noun in value
        else
                object← first adjective in value
        objectAttributes ← EXTRACT-ATTRIBUTES(object)
        result←object ∪ objectAttributes
        if result ≠ failure then return result
        else return failure
```

Figure 3 Algorithm for extracting triplets in treebank output.

The another method for extracting triplets is the Stanford Parser and OpenNLP. Using Stanford Parser, Treebank parse tree can be generated for the input sentence.

Example: *John loves Mary*.

- Syntactic analysis
  (S (NP (NNP John)) (VP (VPZ loves) (NP (NNP Mary)))



After generation of triplets, anaphora resolution is performed for the creation of semantic graph. In linguistics, anaphora defines an instance of an expression that refers to another expression; pronouns are often regarded as anaphors. Anaphora resolution is performed for a subset of pronouns *{I, he, she, it, they}* and their objective, reflexive and possessive forms as well as the relative pronoun *who* [8]. A search is done throughout the document for possible candidates (named entities) to replace these pronouns.

function ANAPHORA-RESOLUTION (*pronoun, number_of_sentences*) **returns** a solution, or failure

candidates←BACKWARD-SEARCH-INSIDE-SENTENCE(*pronoun*)
∪ BACKWARD-SEARCH (*pronoun, number_of_sentences*)
if *candidates* ≠ ∅ **then**
APPLY-ANTECEDENT-INDICATORS (*candidates*)
else
candidates ← FORWARD-SEARCH-INSIDESENTENCE (*pronoun*)
∪ FORWARD-SEARCH (*pronoun, number_of_sentences*)
if *candidates* ≠ ∅ **then**
APPLY-ANTECEDENT-INDICATORS (*candidates*)
result ← MAX-SCORE-CANDIDATE (*candidates*)
if *result ≠failure* **then return** *result*
else return *failure*

function APPLY-ANTECEDENT-INDICATORS (*candidates*)
returns a solution, or failure

result ← APPLY-GIVENNESS (*candidates*) ∪ APPLY-LEXICAL-REITERATION (*candidates*) ∪ APPLY-REFERENTIAL-DISTANCE (*candidates*) ∪ APPLY-INDICATING-VERBS (*candidates*) ∪ APPLY-COLLOCATION-PATTERN-PREFERENCE (*candidates*)
if *result ≠failure* **then return** *result*
else return *failure*

Figure 4 Anaphora resolution algorithms.

Firstly, backward search is performed inside the sentence where we found the pronoun. Next possible candidates are searched in the sentences preceding the one where the pronoun is located. If there is no candidates found so far, forward search is performed within the pronoun sentence. Once the candidates have been selected then the antecedent indicators is applied to each of them and assign scores (0, 1 and 2). After assigning scores to the candidates found, the candidate with the highest overall score is selected as the best replacement for the pronoun. If two candidates have the same overall score, the one with a higher collocation pattern score is preferred. If the decision cannot be made based on the score, the candidate with a greater indicating verbs score is considered. In case of a tie, the most recent candidate (the one closest to the pronoun) is selected.

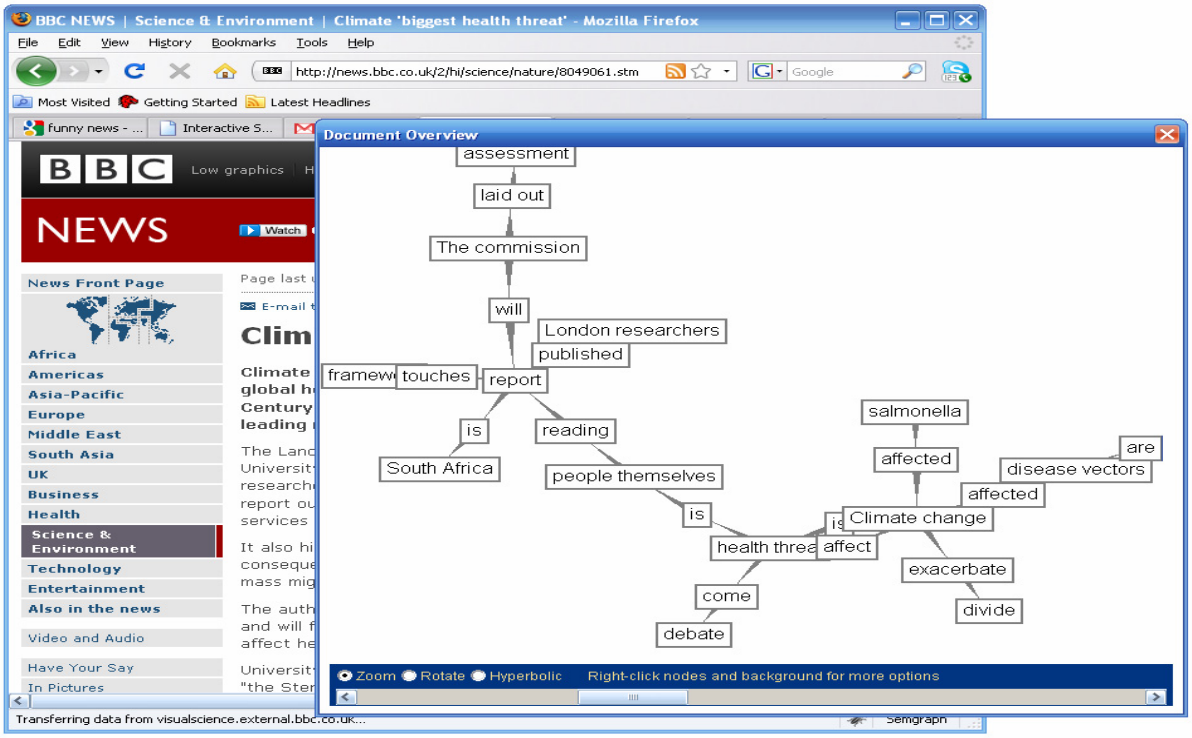


Figure 5 Screenshot showing a web page and its visualization as a graph.

Once co-reference and anaphora resolution have been performed, the next step is semantic normalization [12] where the obtained triplets are compact in order to generate a more coherent semantic graphical representation. For this task, the synonymy relationship between words plays the important role . Synsets are attached to each triplet element found with WordNet. If the triplet element is composed of two or more words then for each of these words the corresponding synsets are determined . So, this procedure is helpful in the next phase when we merge the triplet elements that belong to the same synset. Based on the semantic normalization procedure, the subject and object elements can be merged that belong to the same normalized semantic class [12]. Therefore, a semantic graph is generated having as nodes the subject and the object elements and as edges the verbs. Verbs label the relationship between the subject and the object nodes in the graph.

RESULTS AND DISCUSSION

The visualization of data of the web communicate information clearly and effectively through graphical means. Web pages are designed only for human understanding by mixing content with presentation. The data cannot provide any information if it is not understandable. To understand the data one should understand the meaning since data without meaning is useless and to understand the meaning, the relations of that datas with other datas should be clearly understood which is also called semantic relations. So semantic graph can be one of the better option to show the semantic relations and produce a enhanced web pages.

In future work, addition of some domain ontologies can better disambiguate domain specific terminology. Integration of other Semantic Web resources from LOD (Linking Open Data) datasets such as DBpedia and investigate differences in disambiguation results when using distinct resources and the potential for combining different resources in the same task. Application of WSD (Word Sense Disambiguation) algorithm to improve the Enrycher generated semantic graphs. Besides that colouring of the edges which differentiate the relationship importance in a graph. For example thin edge represent the relation of lower importance and thick edge represent the higher importance relation based on the context of the data.

One of the improvement of the proposed method is the fragmentation of every word annotation with the appropriate sense in context and linking them to the associated RDF resources defining the sense in both WordNet and OpenCyc where the input text fragment word or word sequence will be annotated with the appropriate sense in context and linked to the associated RDF (Resource Description Framework) resources. The motivation behind adding this extension is to provide richer disambiguated annotations of words that are not named entities and to improve semantic graph quality by merging nodes that refer to the same disambiguated concept.

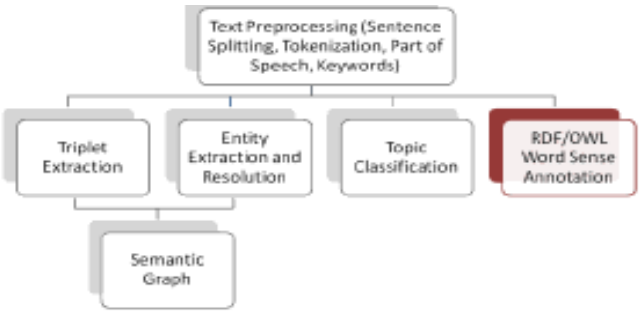


Figure 6 Enrycher component and their dependencies.

ACKNOWLEDGEMENT

This survey was done as a research work to fulfill the requirement of Artifial Intelligence (AI) course under Prof. Bob Mckay, Department of Computer Science and Engineering, Seoul National University, S. Korea.

REFERENCES

[1] Hebert L, The Semantic Graph. *Summary: A tool introduced by Rastier (and based on Sowa, 1984)*.

[2] Mohanty R, Limaye S, Prasad M K & Bhattacharya P, Semantic Graph from English Sentences, *Proceedings of ICON-2008: 6th International Conference on Natural Language Processing Macmillan Publishers, India*. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2008.

[3] Dali L, Rusu D & Mladenic D, Enhanced Web Page Content Visualization with Firefox, 2009.

[4] Leskovec J, Frayling N M & Grobelnik M, Extracting Summary Sentences Based on the Document Semantic Graph, 2005.

[5] Dali L, Rusu D, Fortuna B, Mladenic D & Grobelnik M, Question Answering Based on Semantic Graph, 2009.

[6] Wong P C, Jr G C, Foote H, Mackey P & Thomas J, Have Green – A Visual Analytics Framework for Large Semantic Graphs, *IEEE Symposium on Visual Analytics Science and Technology* , 2006.

[7] Lee M, Kim W, Hong J S & Park S, Semantic Association –Based Search and Visualization Method on the Semantic web page, International Journal of Computer Networks and Communications (IJCNC), Vol.2, No.1, 2010.

[8] Stajner T, Rusu D, Dali L, Fortuna B, Mladenic D & Grobelink M, A Service Oriented Framework for Natural Language Text Enrichment, *Informatica* 34 (2010) 307.

[9] Rusu D, Dali L, Fortuna B, Grobelnik M & Mladenić D, Triplet Extraction from Sentences, 2007.

[10] Stanford Parser: http://nlp.stanford.edu/software/lex-parser.shtml, October 2010.

[11] GATE (General Architecture for Text Engineering): http://gate.ac.uk/, October 2010.

[12] Rusu D, Fortuna B, Grobelnik M & Mladenic D, Semantic Graphs Derived from Triplets with Application in Document Summarization, *Informatica* 33 (2009) 357**.**

[13] Rusu D, Stajner T, Dali L, Fortuna B & Mladenic D, Demo: Enriching Text with RDF/OWL Encoded Senses, 2010.