International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 1, Issue 4, August 2014, PP 1-9
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
www.arcjournals.org

Ontology-Based Indexing and Semantic Indexing in Information Retrieval Systems

Thinn Lai Soe

University of Technology, Yatanarpon Cyber City
thinnlaisoe@gmail.com

Abstract: *There are so many increasing amount of information in the today's world-wide web. For these increasing amounts of information, we need efficient and effective index structure when we have to find needed information. Most indexing techniques directly matched terms from the document and terms from query. The role of usage of domain specific ontology is very wide for the specific application area. Ontology can be defined as a formal explicit specification of a shared conceptualization. It is a formal and declarative representation which includes vocabulary for referring to the terms in that subject area and logical statements that describe the relationships among the terms. This system proposes a semantic-based indexing structure that is built on the basic context document by using semantic suffix tree clustering, context ontology and semantic suffix tree clustering with context ontology. This paper shows the results of these three methods on the same application area.*

Keywords: *clustering, indexing, ontology.*

1. INTRODUCTION

With the tremendous growth of World Wide Web, it has become necessary to organize the information in such a way that it will make easier for the end users to find the information they want efficiently and accurately. . The ever-increasing amount of useful information on the web requires techniques for effective search and retrieval. A major problem is that users can be easily overwhelmed by the amount of information available. The transfer of irrelevant information in the form of documents retrieved by an information retrieval system and that are of no use to the user simply wastes network bandwidth and frustrates users. Another reason for this loss of information and retrieval of irrelevant information is the inability of users to express their queries efficiently and accurately. Therefore, the major issue to be addressed in information selection is the development of a search mechanism that will help in getting maximum relevant documents. In the current scenario, the documents are retrieved if they contain keywords specified by the user. However, many documents contain the desired semantic information, even though they do not contain the user-specified keywords. Ontology is a technique that can be applied to extract the domain and sub domain of the specific keywords.

Traditionally, a single modality, either text or images, has been used to retrieve content in multimodal documents. A simplified diagrammatic view of a single-modality IR system is shown in Figure 1. In general, there are four components in an IR system:

1.1. Indexing

Indexing uses characteristic features to represent documents. Different features are extracted from either textual content of text blocks or visual content of images in a document depending on which modality is used. The indexing here refers to automatic indexing, i.e. Indexes are automatically built without human intervention. The ideal indexing is to dynamically choose a set of features to represent documents given user's information needs.

1.2. Query Formulating and Analyzing

A user formulates a query through the query interface provided by the system. The system analyzes the query and represents it in the same internal format as used for document representations. Different systems differ in their friendliness and complexity of query interfaces depending on which modality is used for retrieval. A user query may be formulated in different ways. The query interface is important for users to form queries to represent their information needs. The details of query formulation and query interface are outside the scope of this paper.

1.3. Retrieval

The system compares document representations and a query representation to retrieve documents using various retrieval models. Retrieval models are surveyed in Section 3. The result of a search is a set of hits containing both relevant (positive) documents and irrelevant (negative) documents.

1.4. Performance Evaluation

Precision and recall are the two most popular metrics to evaluate the effectiveness of text retrieval. Precision is the proportion of retrieved documents that are relevant, and recall is the proportion of relevant documents that are retrieved. Image retrieval and multimodal IR borrow these two terms for effectiveness evaluation. Techniques used for performance evaluation and benchmarking in IR are outside the scope of this paper.

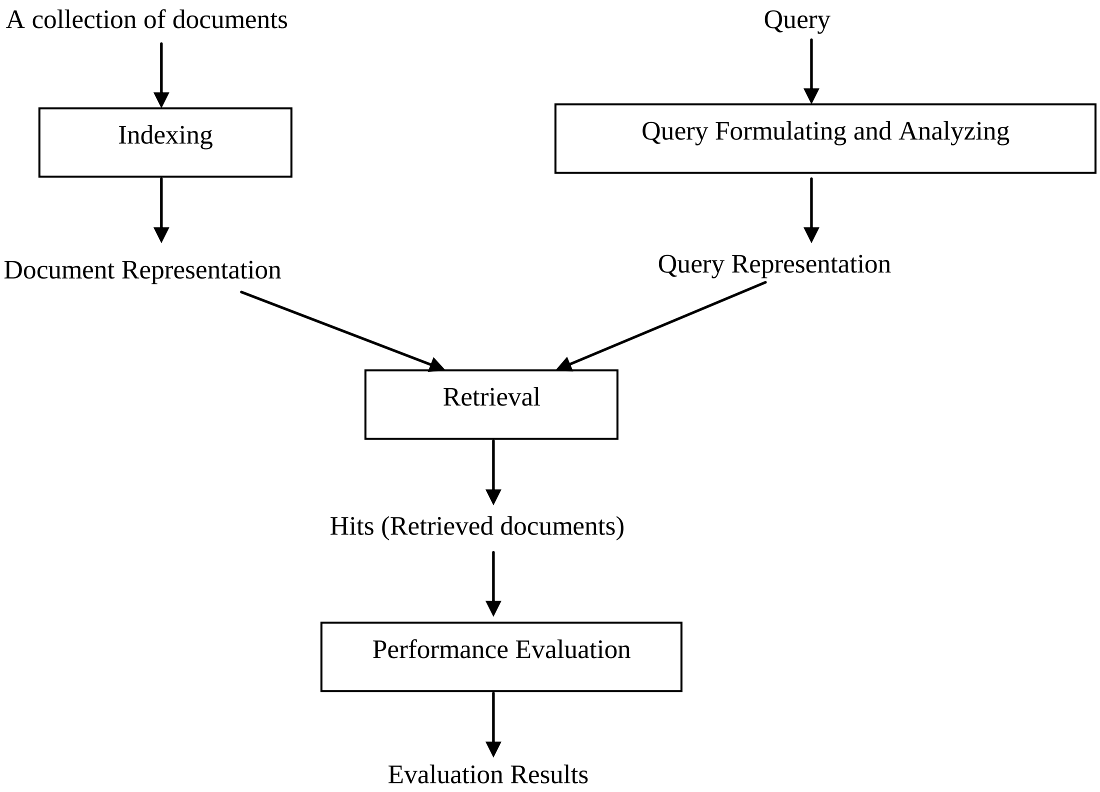

Fig1. *Diagrammatic View of Simplified IR System*

This paper is organized as follows. Section 2 presents the related works on indexing in information retrieval systems. The detail representation and construction of context ontology for our specific application is described in Section 3. Section 4 describes the types of clustering algorithms, semantic suffix tree clustering and semantic suffix tree clustering with context ontology and Section 5 shows the analysis of these three methods. Conclusion of this paper is described in Section 6.

2. Related Works

This section describes the previous works on this domain of indexing and usage of ontology in information retrieval systems. Firstly, related works on indexing are described and then related works on ontology are presented.

In [15], the author proposes an index structure which is based on the concept of ontology. After document's preprocessing has completed, term with maximum frequency matched with the title is extracted from the document. After that, the context of the maximum frequency keyword is searched in thesaurus and ontology repository. This system depends on that frequency value while extracting keyword.

In [1] describes pre-ranking of the underlying similar documents after the formation of the index. Thereafter the ranking of the search results in response to a query takes place which provides relevant results to user. This paper proposes an ontology driven pre ranking of the documents with identical context and hence post ranking of the search results using keyword matching of the expanded query terms and document keywords in the pre-ranked search results.

3. Representation and Construction of Context Ontology

This section describes the representation of our context ontology and how to construct our context ontology. Ontology can be defined as a formal explicit specification of a shared conceptualization. It is a formal and declarative representation which includes vocabulary for referring to the terms in that subject area and logical statements that describe the relationships among the terms. It also provides a vocabulary for representing and communicating knowledge about some topic and the relationships that hold among the terms in that vocabulary.

3.1. Ontology

Ontologies have been realized as the key technology to shaping and exploiting information for the effective management of knowledge and for the evolution of the Semantic Web and its applications. In such a distributed setting, ontologies establish a common vocabulary for community members to interlink, combine, and communicate knowledge shaped through practice and interaction, binding the knowledge processes of creating, importing, capturing, retrieving, and using knowledge.

3.2. Context Ontology

From a computer science perspective, ontology can be viewed as a knowledge base that describes a set of concepts through definitions sufficiently detailed to capture the semantics of a particular domain. In the Artificial Intelligence field, ontology defined as the basic terms and relations comprising the vocabulary of a topics area as well as the rules for combining terms and relations to define extensions to the vocabulary. Current web is the biggest global database that lacks the existence of a semantic structure and hence it makes difficult for the machine to understand the information provided by the user. The Semantic Web provides shared understanding, well structured content and reasoning for extending the current web. Ontologies are essential elements of the semantic web.

The main aim of developing context ontology is lack of WordNet when it faces with context related words (such as a word with different multiple contexts). Other systems may not produce accurate results at every time it meets with context-conflicted words if it doesn’t construct context ontology for the specific domain. Owl ontology consists of Classes, Properties, and Individuals, which presents in turns. They are described as follows:

3.2.1. Classes

Classes are the basic building blocks of OWL ontology. A class is a concept in a domain. Classes usually constitute a taxonomic hierarchy (a subclass-superclass hierarchy). Classes are defined using the owl:Class element. OWL comes with two predefined classes: owl:Thing and owl:Nothing. Owl:Thing is the most general class which contains everything; owl:Nothing is an empty class. Every class we define is a subclass of owl:Thing and a superclass of owl:Nothing. Example of class in celebrity domain is described later.

3.2.2. Define the Properties of Classes

The classes alone will not provide enough information to search from the user desire questions. Once we have defined some of the classes, we must describe the internal structure of the concepts. Thus, the person class has the following data type properties: earphone_url, earphoneName, filmName, filmName_url, footballClub, footballClub_url, lotion_url, lotinName, name, perfume_url, perfumeName, shoe_url, shoeName, showerGelName, showerGelName_url, song_url, songName, sunglasses_url, sunglassesName, T-shirt_url, and the last is T-shirtName.

In general, two types of properties are distinguished in OWL classes:

- Datatype properties, relations between instances of classes;
- Object properties, relations between instances of two classes;

3.2.3. *Create Instances*

The next step is creating individual instances of classes in the hierarchy. Defining an individual instance of a class requires (1) choosing a class, (2) creating an individual instance of that class, and filling in the property value. For example, we can create an individual instance of Person to represent a specific people. Jennifer Lopez is an instance of the class Person. The next figure is the detail explanation of the celebrity context ontology that used in this system.

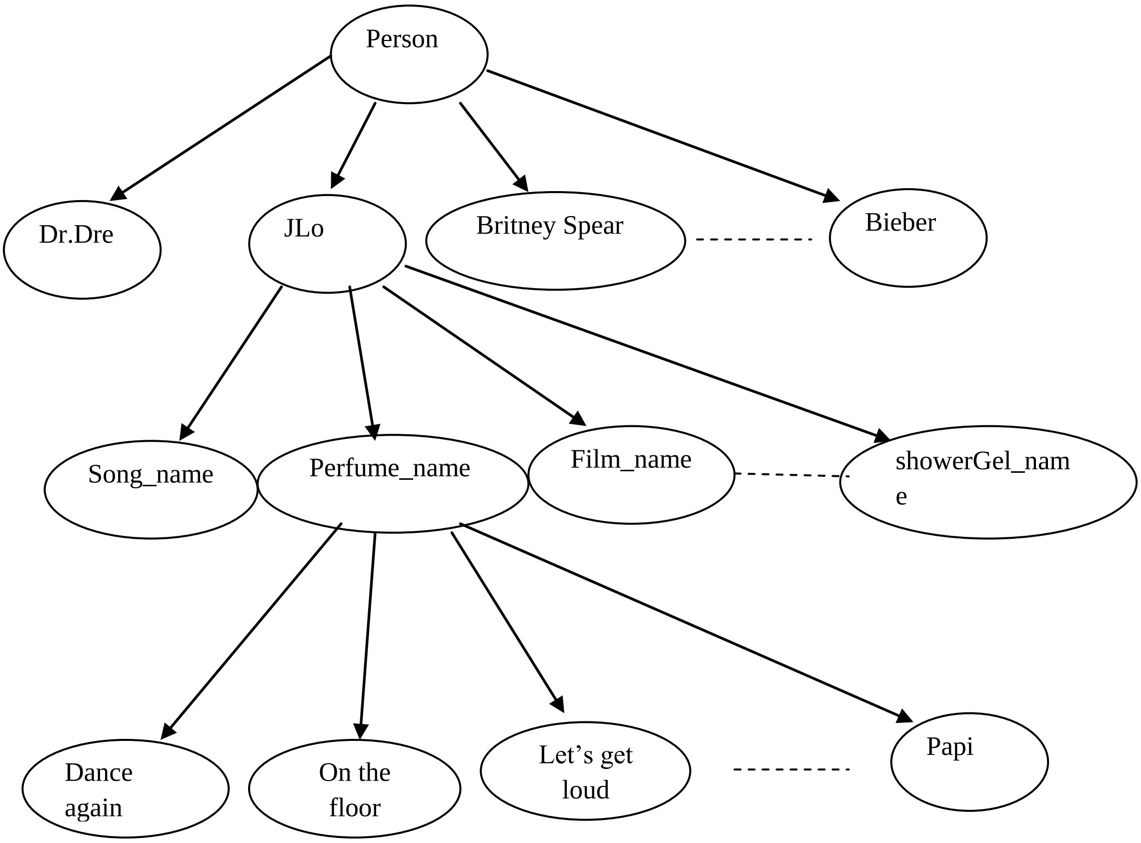

Fig2. *Representation of Person class, its instances and their data type property*

4. Clustering Algorithms

While more and more textual information is available online, effective retrieval is different without good indexing and summarization of document context. This paper uses clustering method called SSTC for indexing in retrieval system in order to get effective search. Document clustering is still a developing field which is undergoing evolution. It started off on the popular vector based approach where documents were treated as a bag of words and clustering criteria was the presence of common words in the documents. Several modifications were applied on this method to improve this method as the result set would only provide us information on what words were present in a group of documents, not the actual content or context of the documents. Numerous document clustering algorithms appear in the literature (see [Willet, 88] for a review). These can be classified into two groups – those producing hierarchical clusters and those producing a flat partition (note that partition algorithms can be applied recursively to create a hierarchy of clusters). Most clustering algorithms rely on an external similarity measure between documents. *Hierarchical Agglomerative Clustering* (HAC) algorithms have been widely applied. These algorithms start with each document in a cluster of its own, iterate by merging the two most similar clusters, and terminate when some halting criterion is reached. Algorithms differ based on their definition of cluster similarity.

4.1. Semantic Suffix Tree Clustering (SSTC)

Suffix Tree Clustering (STC) is a clustering algorithm designated to meet the requirements of post-retrieval clustering of Web search results. Furthermore, STC is also used in indexing process of Information Retrieval system. STC is unique in treating a document as a string, not simply a set of words, thus making use of proximity information between words. STC relies on suffix tree to efficiently identify sets of documents that share common phrases and uses this information to create clusters and to succinctly summarize their contents for users. Document clustering techniques are widely used in these areas like information retrieval (IR for short), organizing documents and indexing and linking.

Semantic Suffix Tree Clustering (SSTC) is an semantic extension of STC. Weakness of STC is that it cannot cluster semantic similar document. Semantic suffix tree (SST) is a new data structure that extends the suffix tree on the meaning of word strings not characters. SSTC is used to cluster web search results and such clusters can also be used in indexing phase for not only a quick and effective search but also an accurate result. Since SSTC have included semantic similarity, it can solve synonyms or hyponyms. According to evaluations, the SSTC can generate specific clusters and more readable labels than conventional STC.

Semantic Suffix Tree Clustering (SSTC) simultaneously constructs the semantic suffix tree through on-depth and on-breadth pass by using semantic similarity and string matching. Semantic similarity is derived from the WordNet lexical database for English language. Therefore, a semantic suffix tree will use both semantic similarity and string matching as conditions to create the suffix tree.

Semantic similarity is a central concept that extends across numerous fields such as artificial intelligence, natural language processing, cognitive science and psychology. Accurate measurement of semantic similarity between words is essential for various tasks such as, document clustering, information retrieval, and synonym extraction. It is a measure that derives the synonyms set from WordNet database. The semantic similarity equation is shown in below:

$$\mathrm{SemSim}\,(w_a,w_b) \quad = \quad 1 \qquad \text{if } \left|\,\mathrm{synset}(w_a) \cap \mathrm{synset}(w_b)\,\right| \geq 1$$

$$\mathrm{SemSim}\,(w_a,w_b) \quad = \quad 0 \qquad \text{otherwise}$$

The above equation uses the association of synonyms set of w_a and w_b to calculate the similarity. For example, a synonyms set of eat= {feed, consume, corrode} and a synonyms set of ate= {eat, feed, consume}, both synonyms sets of eat and ate contain feed and consume, which means more than one word [46]. Therefore, a SemSim (eat, ate) is equal to1. The following figure shows how to check the semantic similarity between two words using WordNet. Since SSTC checks semantic similarity, it can cluster semantically related words in order to improve the quality of the cluster. But it doesn't include context similarity. So, it leads to construct the context ontology when we face with context conflicted words.

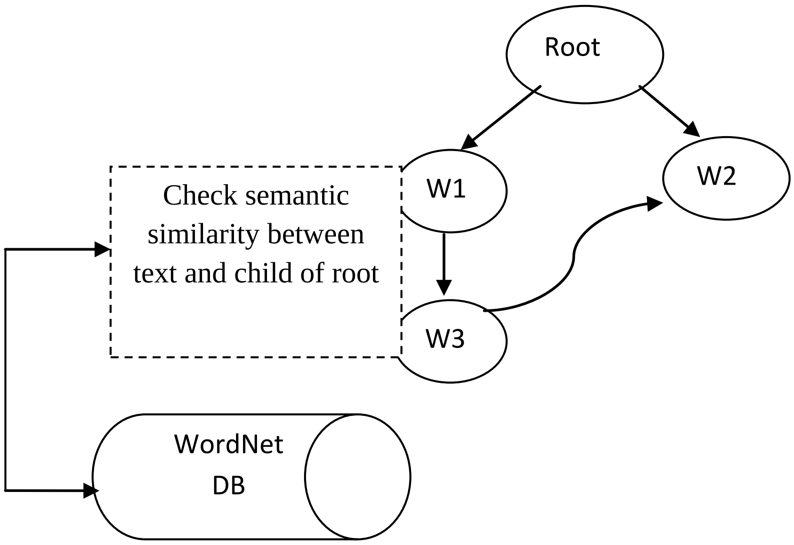

Fig3. *Semantic Similarity between word w1 and w2*

Semantic similarity between words changes over time as new words are constantly being created and new meaning is also being assigned to the existing words. Also there can be a problem with person name detection and alias detection. One person may have multiple names to identify. So there are some problems with the precompiled databases. The new senses of words cannot be immediately listed in any precompiled database. Maintaining an up-to-date taxonomy of all the new words and new usages of existing words is difficult and costly. A solution to this problem is to construct the domain specific ontology for our own specific application.

Here is the example for constructing of two strings using semantic suffix tree clustering. The two sentences are as follows.

S1: Victoria (VT) has 3sons.

S2: David Beckham (DB) has also 3sons.

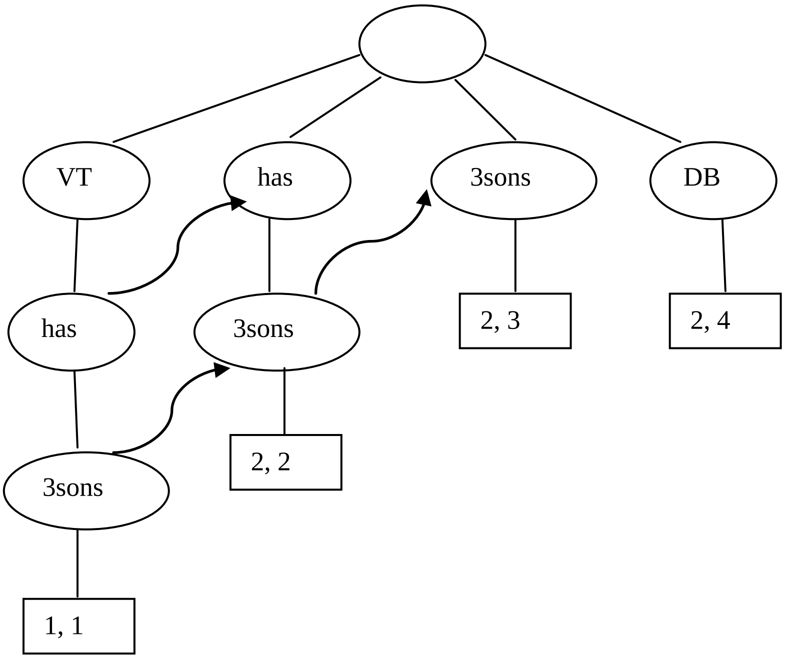

Fig4. *SSTC for both of the above two strings*

4.2. SSTC with Context Ontology

SSTC is a semantic improvement of STC. As presented earlier, SSTC can solve synonyms which are one of the problems of information retrieval. But SSTC cannot distinguish the word which have multiple meaning and is used in many different domains. Some words can be used in two or more domains such as arts domain, electrical domain, cosmetics and so on. Context can handle this situations- a word with different multiple contexts. An additional feature, context, is added onto SSTC. Context ontology can be constructed using the concept of ontology which includes concepts and its relationships. It contains various concepts with their relationships among objects. After clustering the documents using SSTC, different contexts of a word from the documents are extracted using context ontology. SSTC with context ontology can solve the two information retrieval problems which described earlier.

Figure 4 shows the construction of SSTC with context ontology for these two strings. SSTC with context ontology can solve the words with different multiple contexts. Context ontology can enable a words with different multiple contexts such as device (earphone) or human (singer, rapper) in the following figure.

5. Analysis of Three Methods

Context ontology can determine whether the name jlo and Glow after dark means the perfume context. It doesn’t concern with the name of songs. We use context ontology when we would like to know the context of words and how it easily determines among multiple contexts of the words using the context ontology. But it can’t know the words which doesn’t include in our context ontology. If our context ontology constructs completely, it can search well. If not, it can’t search well. So, we have to construct our context ontology more completely. It is one drawback of our context ontology and our search results may depend on the completeness of our context ontology. So, it may reduce our support count probability.

SSTC only use the WordNet when it faces with semantic-conflicted words. So, it can solve semantic problems easily but it doesn't know the name of celebrity (our main application area). So, it can produce all the files in database which related with music when we search the words "jlo music". But the results files doesn't include the files of music of the celebrity name "jlo" and it may produce all the files of Beyonce, Jlo, Justin Bieber etc which is one of drawbacks of not using context ontology.

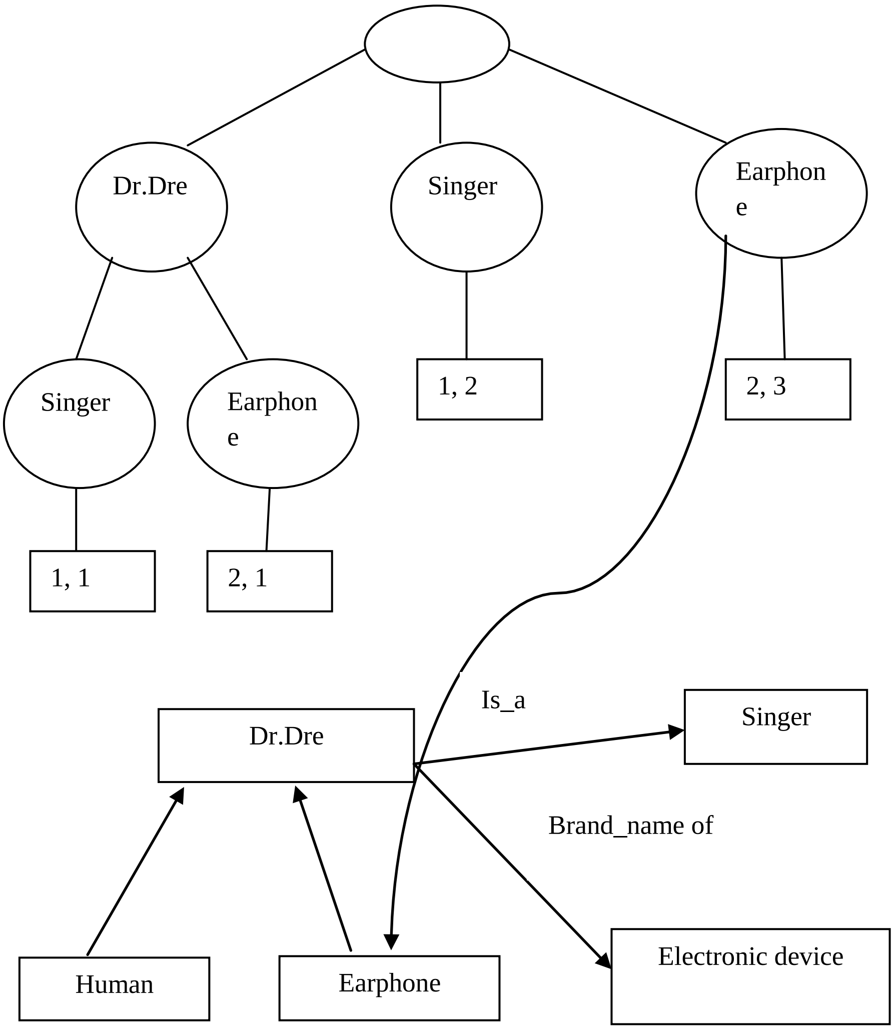

Figure 5. *SSTC with context ontology for both strings*

SSTC with context ontology works well with WordNet when we face with semantic-conflicted words (eg; synonyms of words) and also works well with context ontology (domain dependent ontology) when we face with context-conflicted words (eg; polysemy of words). For example, when we can search not only "jlo and song" but also "jlo and music", SSTC with context ontology produce only the results file related with jlo and her songs. It doesn't produce the result files of other celebrities because SSTC with context ontology include context ontology where it solves context-conflicted words and our context ontology has the class name of celebrity. Furthermore, SSTC with context ontology know whether it search with the keyword using "music" or "songs" because it uses WordNet for semantic related words. We don't have no worry about semantic related words when our context ontology doesn't construct completely. So, it can overcome the drawback of context ontology (complete construction of context ontology) and another drawback of WordNet (lack of context of our application domain). Our SSTC with context ontology can do well and produce more coverage results than the other two.

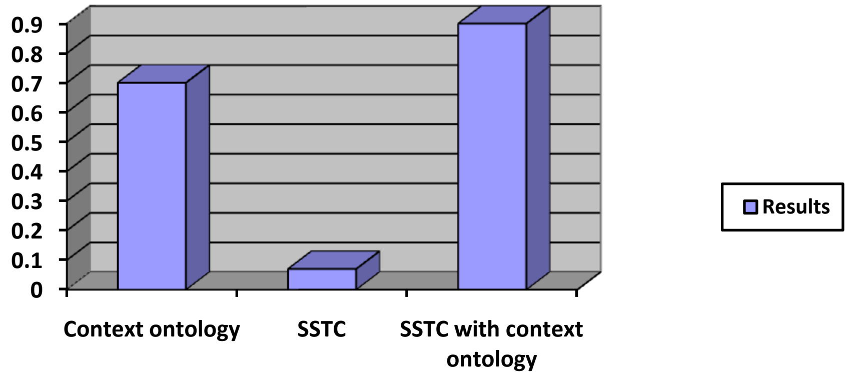

Fig6. *Support count for the three methods*

6. CONCLUSION

In this paper, we can easily see the results of these three methods which are best suits with the desired applications. Index construction is an important part in information retrieval system. This system constructs an index structure which is called context semantic index structure to support the applications related with information retrieval. This paper presents how to construct context ontology for specific domain and also shows the differences between domain specific applications and general purpose applications like WordNet. This system can perform and give the relevant results that the user really wants in a context conflict domain (Dr.Dre may be a name of rapper and also a brand name of earphone etc) used in this paper.

REFERENCES

[1] Parul Gupta et al., “Ontology driven Pre and Post Ranking based Information Retrieval in Web Search Engines”, International Journal on Computer Science and Engineering (IJCSE), ISSN : 0975-3397, Vol. 4 No. 06 June 2012.

[2] Andrew Krizhanovsky, “Related Terms Search Based on WordNet / Wiktionary and its Application in Ontology Matching”, Proceedings of the 11th Russian Conference on Digital Libraries RCDL’2009, Petrozavodsk, Russia, 2009.

[3] Oren Zamir and Oren Etizioni. Web Document Clustering: A feasibility demonstration. In the proceedings of SIGR, 1998.

[4] Janruang, J., Guha, S.: Semantic Suffix Tree Clustering. In: DEIT 2011, IEEE, Bali, Indonesia (2011).

[5] Chim, H., Deng, X.: A new suffix tree similarity measure for document clustering. In: Proceedings of the 16th international conference on World Wide Web (WWW 2007).

[6] Nagwani N. K., Verma S., Software Bug Classification using Suffix Tree Clustering algorithms.

[7] Information Retrieval and Web Search (chapter 6) from Web Data Mining, Exploring Hyperlinks, Contents and Usage Data.

[8] S. Deerwester, S.T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 1990.

[9] N Chen, Technical Report 2006-505 “A survey of Indexing and Retrieval of Multimodal Documents: Text and Images”.

[10] K. Kotis, G. A. Vouros, K. Stergiou, Department of Information and Communication System Engineering, “Towards Automatic Merging of Domain Ontology: The HCONE- mearge approach.

[11] A-Alhenshiri, “Web Information Retrieval and Search Engines Techniques”, in proceding of Al- Satil journal.

[12] http://www.cs.washington.edu/research/clustering.

[13] http://www.w3.org/TR/2004/REC-owl-features-20040210.

[14] http://en.wikipedia.org/wiki/Ontology_(information_science)

[15] Gupta Parul and Sharma, A.K., Context based Indexing in Search Engines using Ontology. In the International Journal of Computer Applications, Vol 1-No.14.

[16] Sajendra Kuar, Ram Kumar Rana, Pawan Singh, "A Semantic Query Transformation Approach Based on Ontology for Search Engine", International Journal on Computer Science and Engineering (IJCSE), May 2012.