Chapter 4

Ontology based Approach to Automated Text Summarization

4.1 Introduction
Ontology is defined as an explicit specification of a conceptualization. Conceptualization consists of the existents and their characteristics. Ontology is interpreted as the formal representation of the conceptualization.

As an analogy, one can describe the ontology of a domain as the relational schema of a database. Relational schema represents both the entities (concepts) and the dependency relations between the entities whereas the ontology consists of concepts and semantic relations between the concepts.

Anatomy of Ontologies: In general, ontology of a domain consists of four major components listed below.

- Concepts: Concepts of a domain are an abstract or concrete entities derived from specific instances or occurrences.

- Attributes: Attributes are characteristics of the concepts which may or may not be concepts by themselves.

- Taxonomy: Taxonomy provides hierarchical relations between the concepts.

- Non-taxonomic Relations: Non-taxonomic relations specify non-hierarchical semantic relationships between the concepts.

4.2 Ontology Based Approaches to Text Summarization
Extrative text summarization aims to create a condensed version of one or more source documents by selecting the most informative
sentences. Research in text summarization has therefore often focused on measures of the usefulness of sentences for a summary. By considering ontology attributes we are able to improve the semantic representation of a sentence’s information content [90].

Each ontology node is populated by a bag-of-words constructed from a web search. Sentences are represented by subtrees in the ontology space, which allows to apply similarity measures in the ontology space and to compute relations between sentences based on graph properties of the subtrees. Furthermore, node confidence weights computed by the classifier enable us to identify the main topics of a document. The classifier maps a sentence to the taxonomy by choosing the subtrees which best represent the sentence. If the maximum similarity of a child is lower than the current node’s similarity to the sentence, or if a leaf node has been reached, the algorithm stops. A sentence is therefore not necessarily classified to a leaf node, but may be assigned to an internal node. For ontology based summarizer we compute a set of features for each sentence based on the output of the hierarchical classifier. If a sentence is mapped to multiple subtrees in the taxonomy, we include all nodes from every subtree. If a sentence is classified as a leaf node of a certain depth, it is assumed to contain more specific information than a sentence that is classified to a higher-ranked internal node. We create a bag-of-words for each sentence by removing stop words and applying stemming.

*EuroWordNet (EWN)*

EWN4 is a multilingual extension of WordNet, a lexical semantic ontology developed at Princeton University. The basic semantic unit is the synset (synonymy set), grouping together several words that can be considered synonyms in some contexts. Synsets are linked by means of semantic labels (hyperonym, hyponym, meronym, etc.). Due to polysemy, lexical entries can be attached to several synsets. Figure below shows the general architecture system. Firstly, a classic linguistic processing is performed over the text (common to most NLP applications: sentence segmentation, tokenization and pos-tagging).
Then, the resulting file is processed by the term extractor YATE in order to obtain a sorted list of all the term candidates that are present in the text. Finally, the summary is produced, taking into account the term candidates, their similarity/termhood values and the input text. Some configuration information is necessary to define some useful parameters: thresholds, number of sentences in the final abstract, etc [15].

Fig 4.1: General Architecture

Below shown in figure summarization technique is knowledge rich and user query-based. It represents the original document with a semantically connected concept network, which is chosen a subset of sentences from the original document as its summary. Approach is totally term based, i.e., we recognize and process only terms defined in Wordnet for general documents (UMLS for medical documents) and ignore all other words. Figure shows the architecture for summarization system.
Here is the summarization procedure:

1. Revise the query with ontology knowledge. Will add relevant keywords, delete redundant keywords and return the revised query and let the user finalize it.

2. Calculate distance of each sentence in the document to the finalized query. Distance function used will be metrics. If the distance is smaller than a threshold, the sentence will be a candidate to be included in the summary.

3. Calculate pair-wise distances among the candidate sentences (metrics can reduce the number of computations required). Then, divide candidate sentences into groups based on a threshold and select highest-ranked one from each group.

### 4.2.1 Ontology Text Summarization

Ontology summarization is the process of distilling knowledge from ontology to produce an abridged version for a particular user (or users) and task (or tasks). It is argued that RDF statements can be the basic unit in the process of distilling. However, extracted summaries in the form of RDF statements will encounter a blank node problem: it is possible that RDF statements sharing common blank nodes are separated, which could lead to a summary containing some meaningless RDF statements. RDF sentences provide integrated
information, which is a combination of a set of RDF statements connected by blank nodes. Thus, they encapsulate the blank nodes in local structures. Using the notion of RDF sentence as a basic analytic unit, the shortcoming of RDF statements will be overcome. Further, an ontology (including RDFS and OWL ontologies) can be viewed as linked RDF sentences, which leads to an RDF Sentence Graph. Figure below exhibits the architecture of our ontology summarization, which is composed of four major components:

![Architecture of Ontology Summarization](image)

**RDF Sentence Builder:** This component accepts user-provided ontology. Usually, a user may also provide the expected length of the final summary, and he is allowed to further customize his navigational preference, which will be used to determine the weight of links between RDF sentences. The ontology is mapped to a set of RDF sentences.

**Graph Builder:** Based on the set of RDF sentences and user's preference, an RDF Sentence Graph is built, which characterizes the links between RDF sentences in the ontology. We will give a formal definition of RDF Sentence Graph.

**Salience Assessor:** This component assesses the salience of RDF sentences based on link analysis of RDF Sentence Graph. In our approach, the salience of an RDF sentence is determined by the centrality of corresponding vertex in RDF Sentence Graph. Since the
RDF Sentence Graph carries the user's preference, the salience of RDF sentence indicates its importance in the ontology from the user's view. The Salience Assessor finally gives a ranking to RDF sentences according to their salience [25].

Re-ranker: This component produces the final summary of the ontology. It does not simply extract the user-specified number of most salient RDF sentences. The coherence of the summary and its coverage on the original ontology are also considered. Salient RDF sentences are re-ranked before extracted into the summary.

![Diagram of animal ontology]

Fig 4.4: Graph Summary of the Animal Ontology

**LINKED RDF SENTENCES**

i. RDF Sentence

An RDF graph $G(T)$ can be mapped into a set of RDF statements $T$, composed of URI references, literals and blank nodes, making descriptions about resources. According to the RDF semantics, blank node is a kind of existentially quantified resources whose meaning is in the scope of the graph it occurs. RDF statements sharing a common blank node form a structure providing a joint context of the blank nodes. If such RDF statements are separated into different graphs, the context is broken. The structure is important to certain applications, which will reference or extract a sub-graph of an RDF graph and
meanwhile require the extraction to retain meaningful. However, RDF semantics does not provide any intrinsic mechanism to identify this kind of structure. We define it as an RDF Sentence:

B-connected: We say two RDF statements are b-connected if they share common blank nodes. Besides, the b-connected relation is transitive, i.e., two RDF statements are said to be b-connected if they are both b-connected to another RDF statement. To guarantee that the b-connected RDF statements are always grouped together, we introduce the notion of RDF sentence, which corresponds to a maximum set of b-connected RDF statements.

An OWL DL ontology can be mapped to a corresponding RDF graph [20], and further to a set of RDF sentences. In OWL abstract syntax, facts or axioms of atomic class (property) are usually mapped to “generic” RDF sentences. Some OWL DL Axioms are mapped to “special” RDF sentences, such as axioms specifying the equivalence between class restrictions. While such axioms are important for the tasks of reasoning, they are less important to certain tasks such as ontology summarization.

The subject and predicate of a generic RDF sentence are the subject and predicate of its main RDF statement; the object of a RDF sentence is the set of terms occurring in the RDF sentence, except the subject and predicate of the main RDF statement.
Here, term refers to URI reference which is defined by users, not belonging to the build-in vocabulary of ontology language. An example is shown in Figure above, which is a subgraph of the RDF graph derived from the Animal Ontology. The graph can be divided into three RDF sentences: S1, S2 and S3.

ii. Links between RDF Sentences

Natural language sentences are sequential in text, while RDF sentences can be structured as a graph. It is natural to define the links between RDF sentences based on common terms they shared. The links can be classified into two classes: sequential or coordinate, depending on the position of common terms.

Sequential Link: There is a sequential link from one RDF sentence to another if the predicate or an element in the object of the former is the same term with the subject of the latter. We name this type of link as sequential since it represents the relation similar to the sequential relation between natural language sentences.

Coordinate Link: There is a coordinate link from one RDF sentence to another if the subject of both RDF sentences are the same. We name this type of link as coordinate since it represents the relation similar to the coordinate relation between natural language sentences, which often appears as a compound sentence.

iii. RDF Sentence Graph

Considering a scenario that a user is navigating inside ontology, when he reads an RDF sentence, he wants to further look up some RDF sentences concerning the object of the current one. In this case, he will follow a sequential link; or he may want to read more RDF sentences talking about the subject of the current one, and then follows a coordinate link. The user may have a preference on which type of link to follow. We characterize the preference by a parameter p, which is a value between 0 and 1 representing the probability of following
sequential links, and thus \( 1 - p \) of following coordinate links. We define an RDF Sentence Graph, which is a weighted and directed graph, characterizing the links between RDF sentences from the viewpoint of a user.

From the definition of RDF Sentence Graph, we can see it carries the preference of a user, which can be a person or software agent. RDF Sentence Graph is customizable, and different users could have different RDF Sentence Graphs for a given ontology. A default preference can be defined based on a generally accepted navigational preference, which will be discussed in the evaluation section.

The RDF Sentence Graph built from RDF sentences in Figure above is shown in figure below:

![Graph](image)

Fig 4.6: A RDF Sentence Graph

iv. SALIENCE OF RDF SENTENCE

A naive method to assess the salience of RDF sentences can be rooted from the idea of “centroid” in text summarization [21] the salience of RDF sentences can be determined by the salience of terms referred by them. Treating the name of each term as a sequence of words and the ontology a bag of terms, the salience of terms can be assessed by comparing to the centroid of the ontology, which is a pseudo-document consisting of words whose frequencies are above a predefined threshold. Although the centroid based salience is easy to assess, the method ignores the graph nature of ontologies. In our approach, the salience of an RDF sentence is assessed in terms of its centrality in the RDF Sentence Graph. The notion of centrality defined on the vertices of a graph is a measurement originated from the analysis of social networks. They are designed to rank the actors according to their positions.
in the network and interpreted as the salience of actors embedded in a social structure.

In this section, we give a survey on different notions of centrality and their measurements, which can be classified into three categories: Degree Centrality, Shortest Path Based Centrality and Eigenvector Centrality. Before we make a conclusion that which measurement is suitable to assess the salience of RDF sentences, five different measurements are selected from the three categories.

4.1.2 Degree Centrality
Degree centrality is a simple measurement of vertices' salience. In an undirected graph, the degree centrality of a vertex is measured by the number of connections it has. For a directed graph, the degree centrality of a node is measured by the number of its incoming links, which is called in-degree centrality; or by the number of its outgoing links, which is called out-degree centrality. Links between vertices can be seen as conferral of authority. Vertices with high degree centrality are intuitively salient in the graph since they receive many conferral of authority from others. The conferral can be discrete for unweighted graph; or continuous for weighted graphs.

Given a weighted directed graph $G = (V; E; W)$, where $V$ is the set of vertices, $E$ is the set of edges, $W$ is the set of edge weights. CI and CO of each vertex are computed as following:

$$C_I(i) = \sum_{(j, i) \in E} w(j, i),$$
$$C_O(i) = \sum_{(i, j) \in E} w(i, j).$$

In this category, we select weighted in-degree (CI) centrality to assess the salience of RDF sentence in an RDF Sentence Graph.

4.2.3 Shortest Path based Centrality
In social network analysis, many notions of centrality are based on shortest paths linking pairs of actors. For example, the salience of an actor can be measured by the maximum or total distance from it to others, which are called Graph Centrality denoted by CG and Closeness Centrality denoted by CC respectively; or the ratio of shortest paths
across the actor in the network, which is called Betweenness Centrality denoted by CB. Shortest-path-based centrality characterizes the position of a vertex in a graph, which brings more information than degree centrality. We select the betweenness centrality (CB) in this category since it provides more topological information than others, which is essential in the analysis of social networks. The original notion of betweenness centrality is defined on undirected and unweighted graph. An algorithm is introduced in to efficiently compute the betweenness centrality in directed and weighted graphs. Salient RDF sentences with high betweenness centrality can be seen as bridges between clusters of RDF sentences: the bridge RDF sentences have direct links to lots of sentences in clusters, while few links lie between sentences belonging to different clusters.

4.2.4 Eigenvector Centrality

Graphs can be encoded in adjacency matrices. The entries in the matrix are either 1 if a connection exists between two vertices, or 0 if not. The matrix is symmetric if the connection is undirected or asymmetric otherwise. Two well-known measurements of eigenvector centrality on the Web is PageRank and HITS. PageRank is used by the Google search engine for ranking web pages. The authority of a page is computed recursively as a function of the authorities of the pages that link to it. HITS computes two values related to topological properties of the web pages, the authority and the hubness. The authorities indicates the page relevance as information source, which are parallel to the main eigenvector of the bibliographic coupling matrix; while the hubness refers to the quality of a page as a link to authoritative resources, which are parallel to the main eigenvector of the co-citation matrix.

4.3 Case Study

The Animal ontology describes a conceptual framework of person and relations between persons, such as parent relation and spouse relation. The ontology are selected as the test case since they are rather small, which can be reviewed by human to produce ground truths. Below figure 4.7 shows the Animal Ontology:
Fig 4.7: Animal Ontology

Fig 4.8: Graph1 of Animal
Fig 4.9: Graph2 of Animal

Table 4.1 OUTPUT - SUMMARY OF ANIMAL ONTOLOGY
4.4 Results

The experiments are performed on various domains as shown in figure 4.10 below and we found that ontology based approach summarization provides 100% accuracy. By considering ontology attributes we are able to improve the semantic representation of a sentence’s information content.

<table>
<thead>
<tr>
<th>Category / Domain</th>
<th>Statistical Based Approaches</th>
<th>Ontology Based Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animal</td>
<td>92.18</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Pizza</td>
<td>News</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>89.18</td>
<td>90.12</td>
</tr>
<tr>
<td>Accuracy</td>
<td>90.49</td>
<td>100</td>
</tr>
</tbody>
</table>

Fig 4.10: Comparison between Statistical approaches and Ontology Based Approached Summarization