VOL. 10, NO. 6, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ONTOLOGY BASED TEXT DOCUMENT SUMMARIZATION SYSTEM USING CONCEPT TERMS

R. Ragunath and N. Sivaranjani
Department of Information Technology, SRM University, Chennai, India
E-Mail: infotechragu@gmail.com

ABSTRACT

In this modern world, due to the dramatic technological development huge amount of information is available in all over the places. So it is difficult to understand the main content of the document without reading the entire document. It takes time, based on the amount of information available in the document. By using the automatic summarization, these problems are solved. In this paper ontology based text summarization system using concept terms is introduced. Concepts are extracted using concept extraction algorithm. By using the ontology model the hierarchical representation is generated for the concept terms. Then by setting the concept depth the required summary is generated.

Keywords: automatic summarization, concept terms, protégé, ontology, rdf, owl.

1. INTRODUCTION

Information plays a key role. Now the days people are overwhelmed by the huge amount of information availability. Due to this, it is becoming harder to understand a topic without reading the entire document. So we need a improved mechanisms to discover and represent the information effectively. Hence the summarization system is emerged. The primary objective of the system is to present the main ideas of the document in less space. Manual summarization would require huge amount of time and cost. Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. Automatic summarization can be classified into two categories: abstraction and extraction. Extraction methods work by selecting a subset of existing words phrases or sentences in the original text to form the summary In contrast abstraction methods build an internal sematic representation and use natural language generation techniques to create a summary.

Input to the system can be one or more documents. Several studies have demonstrated that using a knowledge base in the process of extractive summarization improved results. Many researchers preferred to use ontologies to represent their knowledge.

The rest of the paper is organized as follows, section 2 presents related work, section 3explains the proposed system, and section 4presents experimental results and finally section 5gives the conclusion and future work.

2. RELATED WORK

Ibrahim Imam *et al*. developed the Ontology-based Summarization System for Arabic Documents, (OSSAD) [1]. It is based on user query. The user's query is first expanded by using the Arabic WordNet and then by adding the domain-specific knowledge base to the expansion. Rakesh Verma, Ping Chen and Wei Lu proposed the Semantic Free-text Summarization System Using Ontology Knowledge to generate the summarization [4]. This system also retrieves and ranks information according to a user's query. There is no threshold value for selecting the sentences in the documents. And also this system is used for single domain only. Kamal Sarkar proposed the summarization method [5] developed by using combines several domain specific features with some other known features such as term frequency, title and position. It can be applied for single domain only. Manne Suneetha propose a text summarization system by generates a summary for a given input document based on identification and important sentences in the document [8]. The system use frequent term based test summarization technique with HMM tagger. The summary is obtained by the ranked sentences that have been collected by identifying the feature terms.

3. PROPOSED SYSTEM

Figure-1 shows the sequence of steps involved in Ontology based text Document Summarization using Concept terms.

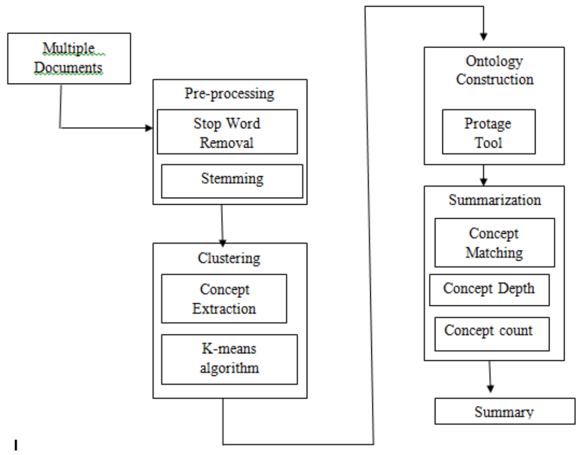


Figure-1.

Proposed system uses extraction summarization approaches to perform the automatic summarization. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary.

The importance of extraction method for summarization and was the first to employ stop-lists in order to filter uninformative words of low semantic content (e.g. most grammatical words such as "of", "the", "a").

Stemming Algorithm can be applied to identify the key words of a text.

A. Pre processing

Data pre-processing is an important step in the data mining process. It has the following steps.

The first stage is tokenization, splitting of the sentences into words by following white space as the separator.

Then the process is followed to Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc.,

Example “And now for something completely different”

And-CC
Now-RB
For-IN
Something-NN
Completely-RB
Different-JJ

The next stage is Stemming. Stemming is the methodology to get the root of the particular word in the document. Porter Stemming algorithm is used to perform the stemming process.

By using the root word,we can calculate the frequency of the word in the text.

Example
connection
connections
connective ---> connect
connected
connecting

After the stemming, process is continued by removing Stopword. Stopword removal is done by comparing each word in the sentence with stoplist (stopword list). Examples of stopword are conjunctions, articles and prepositions. The output word from all of the process are called a term.

B. Concepts extraction

Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

TF: Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

IDF: Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF (t) = log_e(Total number of documents / Number of documents with term t in it).
Tf-idf weight =tf* idf;

Example

Consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log (10, 000, 000 / 1, 000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12.

C. Clustering

Clustering is the process of grouping a similar or related terms into a single group. Concepts are identified by a concept extraction algorithm from text. These concepts are called the key concepts of the target domain.

In the proposed system it uses the k-means algorithm which is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters.

Steps to achieve clusters are:

- Given a high dimensional concept vector
- Generate concepts for clustering (terms and related terms)
- Construct the initial clusters based on concepts (terms and related terms)
- Make the cluster disjoint in order to identify the best initial cluster and keep the document only in that cluster by calculating the goodness score
- Build the cluster

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. Cosine similarity then gives a useful measure of how similar two documents are likely to be in terms of their concepts.

D. Ontology creation

Ontology can be built by two ways. One way is developing tools that are used by knowledge engineers or domain experts to build the ontology like Protégé and Jena. They are called the ontology modelling tools. Another way is semi-automatic or automatic building of ontologies by learning it from different information sources .Some of the ontology extraction tools have reference ontology to update it with the new concepts and relations deducted by the tool.

There are some available tools that extract ontology from text, such as Text-To-Onto, and its successor text2Onto, Onto Learn, protégé plugin Onto LT, and etc. The main objective of proposed system is to make the effective summarization. So we use the protégé tool to construct the ontology.

We collect vocabularies and synonyms. Next, we put those words by the Data model of ontology. The first step of our method is to determine the main subtopics of the article of interest. This is achieved by comparing the words of articles with terms in the ontology. If the word does not exist in the ontology, we ignore it. Otherwise, we record the number of times the word appears in the ontology.

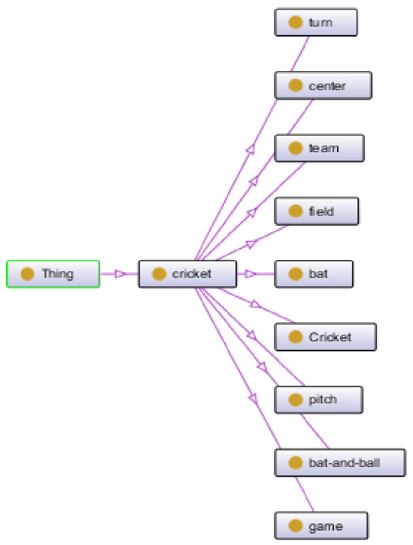
We encode the ontology with a tree structure, and each node includes the concepts represented by the node's children. When the count of any node increases, the counts associated with their ancestors will also increase. By this design, the root of the ontology will always get the highest grade, while nodes in the second level, which represent subtopics, will get different scores. After marking the counts of the nodes in the ontology, we select second-level nodes that have higher counts as the main subtopics of the article.

It is the ontology creating technique by which the data from the document is mapped against its tag and RDF is generated. For example take cricket related content as an input. Then doing as the necessary steps we identified the 'cricket' is the word which has high weight.

Vacoubalary for the education is identified by the use of wordnet then using that the rdf file is generated. Example for the rdf file is shown below:

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
xmlns="http://www.owl-
ontologies.com/Ontology1422958643.owl#"
xmlns:protege="http://protege.stanford.edu/plugins/owl/pr
otege#"
xmlns:xsp="http://www.owl-
ontologies.com/2005/08/07/xsp.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:swrl="http://www.w3.org/2003/11/swrl#"
xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://www.owl-
ontologies.com/Ontology1422958643.owl">
<owl:Ontology rdf:about=""/>
<rdfs:Class rdf:ID="game">
<rdfs:subClassOf>
<rdfs:Class rdf:ID="cricket"/>
</rdfs:subClassOf>
</rdfs:Class>
<rdfs:Class rdf:ID="field">
<rdfs:subClassOf rdf:resource="#cricket"/>
</rdfs:Class>
<rdfs:Class rdf:ID="pitch">
<rdfs:subClassOf rdf:resource="#cricket"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Cricket">
<rdfs:subClassOf rdf:resource="#cricket"/>
</rdfs:Class>
<rdfs:Class rdf:ID="center">
<rdfs:subClassOf rdf:resource="#cricket"/>
</rdfs:Class>
<rdfs:Class rdf:ID="bat-and-ball">
<rdfs:subClassOf rdf:resource="#cricket"/>
</rdfs:Class>
<rdfs:Class rdf:ID="bat">
<rdfs:subClassOf rdf:resource="#cricket"/>
</rdfs:Class>
<rdfs:Class rdf:ID="team">
<rdfs:subClassOf rdf:resource="#cricket"/>
</rdfs:Class>
<rdfs:Class rdf:ID="turn">
<rdfs:subClassOf rdf:resource="#cricket"/>
</rdfs:Class>
</rdf:RDF>
```

By giving the generated rdf file as input to the proteage tool we can generate the owl file and ontograph as an output. Which is look like as below?



E. Summarization

The summary is generated by following the below steps. As the distance measure, we compute the cosine distance between the feature vectors of the sentence and a category. Starting at the root node, the algorithm computes the similarity of a sentence to all child nodes, then determines the mean μ and standard deviation σ of the resulting similarities, and selects all nodes for further exploration whose similarity to the sentence sim (sentence, node) > μ +ασ. The parameter α determines the branching behavior. Setting it to a very high value makes the algorithm choose only a single path. 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 classifier assigns all categories of the sub trees as valid tags, thus allowing us to match nodes sharing common sub trees.

For our ontology-based summarization we compute a set of features for each sentence based on the output of the hierarchical classifier. The α-parameter controlling the number of tags assigned to a sentence is set to 1.5. We create a bag-of tags for each sentence by collecting the nodes computed by the hierarchical classifier. If a sentence is mapped to multiple sub trees in the taxonomy, we include all nodes from every sub tree. We use the classifier's confidence weights to compute a sub tree overlap measure for each sentence. By aggregating the bag-of-tags of the sentences we can form a document's bag-of-tags:

$$wd(t) = \sum_{i \in sentences(d)} conf(t, i)$$

Where w d (t) is the document weight of tag t and conf (t, i) is the confidence value of tag t in sentence i. The tags with the highest weights can be interpreted as the main topics of a document.

We normalize the bag-of-tags with associated confidence values for each sentence and document to unit length. The tag-based similarity measure of a sentence to its document is then the dot product of the two vectors. This measure captures how well a sentence represents the information content of its document in the ontology-space.

The number of subtrees computed by the classifier, as well as the tree depth of its most specific tag, is also assigned as features for each sentence. The latter is interpreted as a measure of the specificity of a sentence: 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. The number of subtrees can be interpreted as a measure of the quantity of a sentence's information content.

Tag overlap

Cosine distance of confidence-weighted sentence tag vector to document tag vector

Concept depth

Depth of the most specific node assigned by the hierarchical classifier

Concept count

Number of sub trees assigned by the hierarchical classifier

4. CONCLUSION AND FUTURE WORK

This paper works on the main techniques to generate multi-document summarization, and describes the details of each step. In this paper we presented our on-going work on user ontology-based summarization system. Ontology knowledge is proven to be an effective way to go beyond the concept term based information retrieval methods.

REFERENCES

[1] Ibrahim Imam, Alaa Hamouda Hebat, Allah Abdul Khalek. "An Ontology-based Summarization System for Arabic Documents (OSSAD)" Vol. 74, No.17, International Conference of Computer Applications. July 2013.

[2] J. Jayabharathy, S. Kanmani and A. Ayeshaa Parveen, "Document Clustering and Topic Discovery based on Semantic Similarity in Scientific Literature" 2nd international conference on Data Storage and Data Engineering, DSDE2011, 13th-15th May 2011, china.

[3] Dragomir R. Radev, Kathleen McKeown, "Introduction to the Special Issue on Summarization", Computational Linguistics - Summarization, Vol. 28, No. 4, pp. 399- 408, 2002.

[4] Rakesh Verma, Ping Chen, Wei Lu, "A Semantic Freetext Summarization System Using Ontology Knowledge", IEEE Transactions on Information Technology in Biomedicine. Vol. 5, No. 4, pp. 261-270, 2009.

[5] Kamal Sarkar, "Using Domain Knowledge for Text Summarization in Medical Domain", International Journal of Recent Trends in Engineering, Vol. 1, No. 1, pp. 200-205, 2009.

[6] A. A.Kogilavani, B., P. Balasubramanie, "Ontology Enhanced Clustering Based Summarization of Medical Documents", International Journal of Recent Trends in Engineering. Vol. 1, No. 1, pp. 546-549, 2009.

[7] Ping Chen, Rakesh Verma, "A Query-based Medical Information Summarization System Using Ontology Knowledge", Computer-based Medical Systems (CBMS), 19th IEEE International Symposium, USA. pp. 37-42, 2006.

[8] Suneetha, M., Sameen Fatima, S.: "Corpus based Automatic Text Summarization System with HMM Tagger". International Journal of Soft Computing and Engineering (IJSCE). 1(3): 118-123(2011) ISSN: 2231-2307.