ONTOLOGY-BASED AUTOMATIC TEXT SUMMARIZATION

by

MEGHANA VISWANATH

(Under the Direction of Krzysztof J. Kochut)

ABSTRACT

This thesis presents an ontology-based approach to automatic extractive summarization of text. Most of the extractive summarization systems so far have used statistical importance measures to determine importance of sentences. We use a knowledge-based approach which makes use of ontological knowledge to determine sentence importance. The Wikipedia ontology is the source of this knowledge. A sub-graph of the ontology is extracted after mapping the input document onto the ontology. The sub-graph, called the Thematic Graph, contains ontology concepts which match the terms in the document and edges from the ontology which represent relationships between the concepts. Hence, the thematic graph represents the theme of the input document. The thematic graph thus obtained is then analyzed using various graph-based importance measures to determine the relative importance of nodes. These values are used ultimately to decide which sentences are included in the summary for the document.

INDEX WORDS: Semantic Web, Extractive Summarization, Ontology

ONTOLOGY-BASED AUTOMATIC TEXT SUMMARIZATION

by

MEGHANA VISWANATH

B.E., Vishweshwaraiah Institute of Technology, India, 2006

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2009

ONTOLOGY-BASED AUTOMATIC TEXT SUMMARIZATION

by

MEGHANA VISWANATH

Major Professor:	Krzysztof J. Kochut
Committee:	John A. Miller I. Budak Arpinar

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2009

DEDICATION

I dedicate this thesis to my parents, Veena and Viswanath Guptha. Their unconditional love and support has made this work possible.

ACKNOWLEDGEMENTS

I am extremely thankful to my major professor, Dr. Krys Kochut, for his guidance, support and timely advice during the course of this thesis. I would also sincerely wish to thank Dr. Budak Arpinar and Dr. John Miller for agreeing to serve on my committee and for introducing me to the field of Semantic Web.

I would like to thank my friends Vinatha Muralikrishna and Jowairia Umeruddeen for their constant support. I also thank my friend Padmashree Ravindra for all her advice and encouragement during the writing of this thesis. I thank also my labmate Gurinder Gosal for his help during the initial stages of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

The amount of information that is being generated in today's world is increasing tremendously with each day. There has literally been an explosion in the amount of information being generated in electronic form and exposed through the World Wide Web. This is especially true in the field of research and development where the frequency of the number of papers and articles being published is augmenting everyday. This presents the need for users to be able to skim through several different documents and quickly find the information that they are searching for. Summaries can help enormously in achieving this objective.

A summary refers to an abridged or a condensed version of a document. It is a concise and brief description of the original document outlining the most important points contained within it thereby removing the need to have to read the full text.

Some of the situations in which summaries are of enormous help are described below. As research students we often have to go through many research papers and the most intuitive way to sort them would be by reading the abstract and the conclusion both of which represent the summary. Hence, summaries of documents and papers help tremendously while conducting research. As mentioned above, researchers need to look through and read from a very large database of potentially useful documents while conducting research. Many documents do not necessarily contain abstracts as part of it. Even if an abstract is present, it gives us an overview of what the document is talking about and does not necessarily list out the most important ideas or points in the document. Hence, the task of sorting out documents can be made easier if summaries can be produced automatically relieving the researchers of having to read through the entire document before deciding if the paper is useful. Another situation in which summaries are useful is while reading news articles. Summaries of news articles help readers browse through the most important aspects of the article instead of having to read the full-length article. Hence, summarization systems prove to be very useful tools in various situations.

Classification of summaries

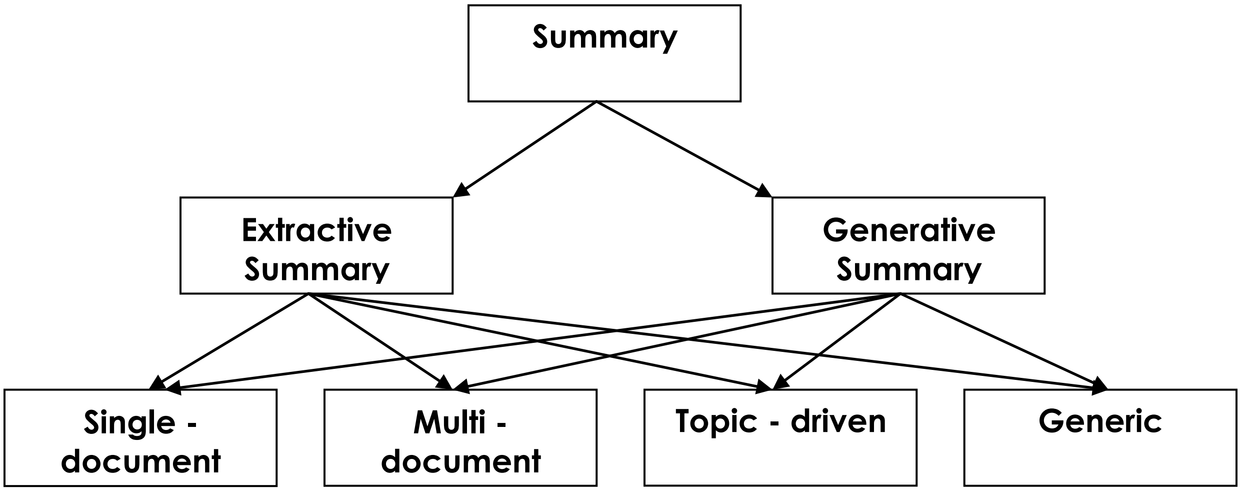


Figure 1.1 Classification of Summary Types

Summaries can be classified mainly into two types – *generative* and *extractive* (or abstractive) summaries. Generative summaries are formed by creating new sentences which together concisely describe the information present in the document. This technique is similar to the manner in which humans summarize documents. Extractive summaries on the other hand use methods such as statistical significance metrics to choose the most important sentences from the original text. These methods assign various importance weights to sentences and terms in the document. A few of the methods based on which weights can be assigned include the location of the term in the document, frequency of the term and inverse document frequency.

Another classification of summaries is based on whether the summary is formed out of a single document or a group of documents all talking about the same topic. Multi-document summarization has some extra issues to be taken care of when compared to single document summarization. Problems such as redundant information from multiple documents, coherence in the summary when summing-up several documents have to be addressed.

A third classification of summaries comes from whether a summary for a document is created focusing on a specific sub-topic in the document or whether it is generic in nature. A topic-driven approach summarizes only those parts of the document that are related to the topic in question. For example, consider a document which talks about the movie 'The Godfather'. If a user wants to get an idea of what the document says about Marlon Brando, he/she would want a summary that focuses only on Marlon Brando. A generic summary, on the other hand, does not focus on any one topic and hence would contain all of the important and informative points in the document.

We present here an extractive, single-document, ontology-based approach to summarization of text documents. The approach makes use of an ontology formed out of the information present in Wikipedia articles to choose sentences from the original document which will become part of the final summary.

By ontology based, we mean that the summary is formed based on the knowledge present in the ontology. An ontology formed out of Wikipedia is chosen since it covers a very wide range of topics. The terms in the document that needs to be summarized are matched against the entities present in the ontology to form a thematic graph. This graph contains entities which represent the theme of the document. Each of the graph nodes is assigned a weight based on various factors. The largest connected component is then chosen to be the thematic graph that best represents the document content with respect to the ontology. Several graph centrality measures are used to further determine the importance of entities with respect to the document and scores are assigned to the nodes. These scores are then used to determine the most important sentences in the document which become part of the final summary.

The rest of this thesis is organized as follows: Chapter 2 presents the background on this thesis explaining the basic concepts. The related work for summarization is presented in Chapter 3. Chapter 4 introduces and elucidates the idea of the Thematic Graph. Chapter 5 presents a novel ontology-based text summarization method, a number of different sentence identification techniques, and a prototype summarization software system. Chapter 6 contains information about the implementation of the project. Chapter 7 includes the experiments conducted and evaluation of the results obtained. Chapter 8 presents the conclusions and the future work.

CHAPTER 2

BACKGROUND

2.1 Summarization

Summarization is the art of abstracting key content from one or more information sources [1]. Summaries are beginning to become a necessity in everyday life. People buy books after reading the summary present at the back of the book. We decide which movie to catch on a weekend based on critic reviews. We also decide which newspaper article to read by first looking at the title of the article. Hence, summaries can take various forms and can sum up various kinds of information. Summarizing speech and video are bigger challenges. Automatic summarization of speech has been tackled in [20] where meeting recordings were summarized using various methods.

As mentioned already, summaries are basically divided into two main types - extractive and generative or abstractive summaries. Extractive summaries are of the kind in which sentences and phrases are extracted from the original document and made part of the summary. An extractive summary basically is a subset of the sentences in the original document. The work here lies in deciding correctly which sentences or phrases to pick such that they reflect the main ideas in the document. Generative summaries, as the name suggests, generates new, original sentences much like how a human would write a summary. This type of summarization entails many more issues and much more work in comparison to extractive summarization. Natural Language Processing is a very important aspect of abstractive summarization since brand new sentences need to be formed.

Below is a piece of text from a news article entitled – "Hurricane Gilbert Heads Toward Dominican Coast".

"[1]Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas.

[2]The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.

[3]"There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday. [4]Cabral said residents of the province of Barahona should closely follow Gilbert's movement. [5]An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.

[6]Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night. [7]The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.

[8]The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm."

The numbers in the brackets in the above text represent the sentence numbers. A human being would pick certain sentences deemed most important in putting across the ideas in this piece of text. The choice of sentences is mostly subjective, but most people's choices would probably concur in this situation. The first sentence that one would probably pick is the first sentence of the article since it tells the whole story in once sentence. One would then pick sentences 3 and 7 because these sentences provide some important information from this article. Sentence 3 conveys what an important official says about the situation and sentence 7 provides information about the exact position and location of the hurricane. The choice of sentences may not be the same for another person, but there usually is some overlap. One person might think that a specific sentence contains information that is more important to him/her than another sentence. It is not possible for a system to understand the requirements of each user, but it attempts to present a summary that most users are comfortable with even if they do not completely agree with it.

Challenges faced while building a summarization system

Automatic summarization of text, whether extractive or generative, has some issues that need to be tackled in order for the summaries to be of better quality.

1) *Coherence* – When a document is coherent, there is a flow or continuity from one sentence to the next. There usually is a loss of coherence and readability in the resulting summary after a document has been summarized by automatic means. In extractive summaries, this loss is expected, especially when the document size is large when compared to the summary size required. This is also true in case of abstracts, but to a much smaller extent. The manner in which abstracts are formed inherently tackle the problem of loss of coherence. In extracts, there is little we can do to reduce this loss. Hence, in this case, *relevance* becomes more important than coherence. Though the summary jumps abruptly from talking about one thing to something completely different in the next sentence, that sentence will be more relevant to the summary than a sentence which follows from the previous without adding much to the information content in the summary.
2) *Dangling anaphoras* – Both extracts and abstracts face the problem of dangling anaphoras. A dangling anaphora refers to a pronoun which has not been previously introduced in the text. Consider a situation where a noun or a proper noun has been previously introduced in a sentence (A) and is now being addressed using a pronoun in the current sentence (B). If the summarization system chooses sentence B to be part of the summary and does not choose sentence A, the pronoun would lead to an ambiguity in the summary.

For example, if the first two sentences in the original text are the following – *"Roger Federer is the greatest tennis player in tennis history. He won his 15th Grand Slam this year beating Pete Sampras' record."* If the summarizer picks the second sentence and leaves out the first in forming the summary, we would not know which person the sentence is referring to unless we have some prior knowledge about the subject.

2.2 Ontology

The concept of an ontology was introduced by the Greek philosopher Aristotle. Wikipedia [30] describes an ontology as "the philosophical study of the nature of being, existence or reality in general, as well as of the basic categories of being and their relations".

An ontology in the field of computer science has been defined formally as "the specification of a conceptualization" by Tom Gruber [4]. An ontology can be described as a collection of entities and relationships between those entities. It can be represented in the form of a graph with entities for nodes and relationships for edges.

Ideally, each and every entity known to man would be represented by a *URI (Uniform Resource Identifier)* for unique identification. All known relationships to all other entities are recorded for each entity. This would form an all-encompassing ontology which is the ultimate fantasy of every computer scientist and also a very unrealistic idea. Hence, ontologies are created such that they are limited to contain entities only from a particular domain. Formation of these domain-specific ontologies would normally require assistance of a domain expert. Examples of domain-specific ontologies include *WordNet* which is a thesaurus in the form of an ontology, *Gene Ontology* which contains information about genes, *OpenCYC*, an upper ontology, which contains encyclopedic knowledge, etc.

An ontology can hence be thought of as a directed graph with nodes and links connecting those nodes. A node can represent a concept such as ‘person’ or an individual such as ‘Barack Obama’.

An ontology provides all the people working in a specific field with a common vocabulary which can be used to share information pertaining to that domain. One of the most widely used ontologies is the *Dublin Core* ontology which is a specification for the description of digital media such as text, videos, images and even web pages. It contains concepts such as ‘title’, ‘creator’, ‘subject’, etc. Concepts in this ontology are incorporated when defining other ontologies to provide information about the ontology itself.

There are several methods that are used to represent an ontology, i.e., serialize an ontology. The *Resource Description Framework* or *RDF* is one way of serializing ontologies. Entities are represented in the form of classes and objects. Edges are represented as relationships between entities. It can be likened to the concepts in an Object Oriented programming language.

Another very popular language for describing ontologies is called the *Web Ontology Language* or *OWL*. This framework is much more expressive when compared to RDF, i.e., it can be used to express more complex relationships between entities as well as constraints on them.

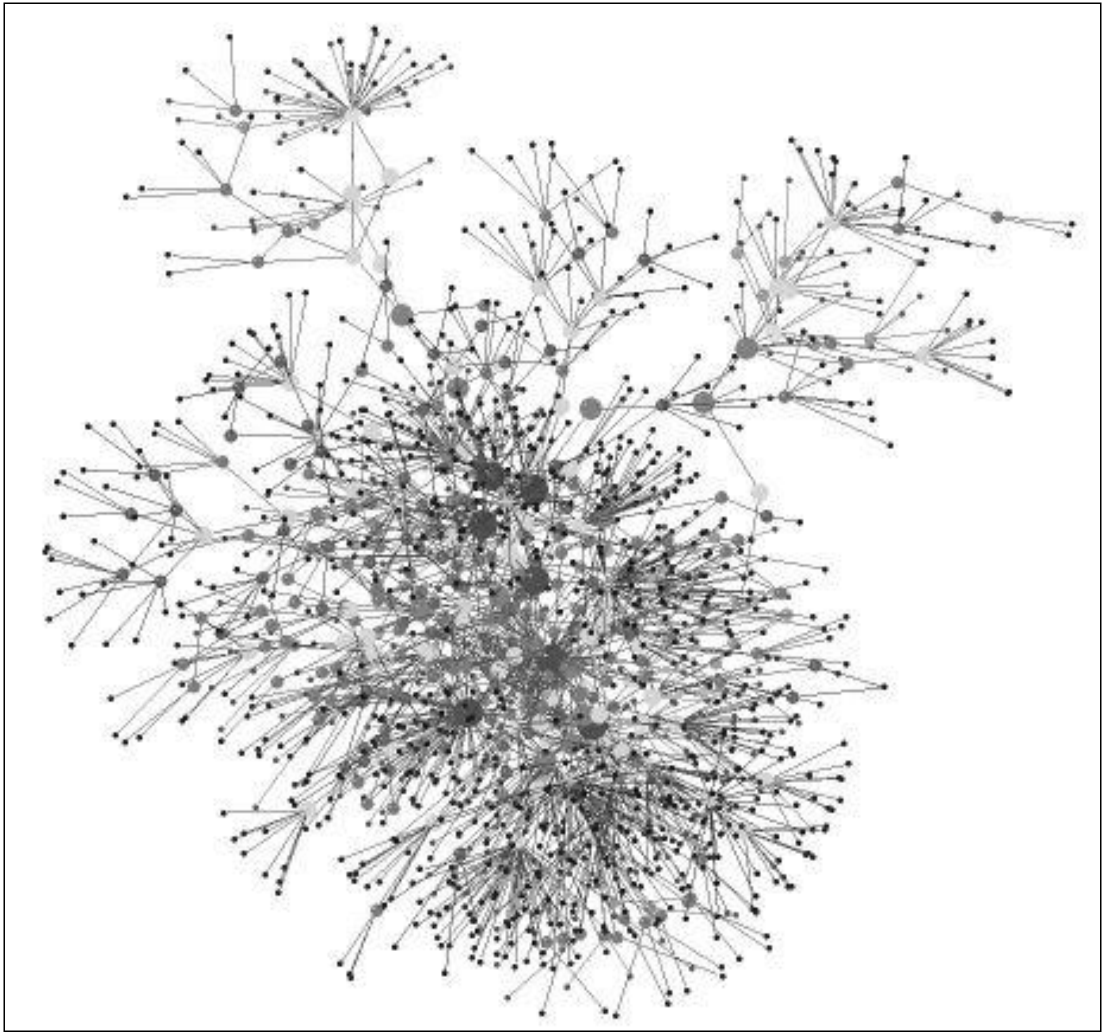


Figure 2.1 The Glycomics Ontology

The above figure is a graphical representation of the Glycomics ontology created at the LSDIS lab at UGA in collaboration with the Complex Carbohydrate Research Center or CCRC also at UGA. We observe from the above figure of the ontology that they are far from being trivial. Even if we are creating an ontology for a small domain, designing the same is not at all a simple task. Ontologies can have hundreds of classes and properties and thousands of individuals. The above figure is not very informative in terms of what an ontology is made up of and is only a pictorial representation which shows the nodes and edges in an ontology.

2.3 Wikipedia

Wikipedia has become quite indispensable to most of us web users. It is a free, online, shared-community, user-updatable, multi-lingual encyclopedia. It contains articles ranging from science to entertainment, from history to politics and any other domain one can possibly think of. Wikipedia being a collaborative effort of countless volunteers is always growing and expanding. Any new event, discovery, invention is almost immediately turned into a new Wiki article. Wikipedia has more than 3 million articles in the English language as of today. Around 1,300 articles are added to the Wikipedia knowledge base everyday as of August 2009 [26]. Hence, Wikipedia is highly up-to-date source of information. Having access to such a large source of information and using it only as an online encyclopedia seems like such a waste. Wikipedia, for the most part, contains information in weakly structured form i.e. in the form of continuous text. It does contain information in the form of structured data in infoboxes and templates. However, that represents only a very small portion of the information available to us from Wikipedia. To glean information from unstructured data is not an easy task. Hence, we need to structure the information in such a way that it can be retrieved and made use of.

The DBpedia project [9] provides one way of structuring the Wikipedia data in the form of an ontology. Each article in Wikipedia is assigned a universal unique identifier which is nothing but a URI in ontological terms. Hence, each Wikipedia page is an entity in the ontology. Certain information about each entity is obtained using several extractors. For example, article titles are associated with a URI through the *rdf:label* property, disambiguation links are represented using the *dbpedia:disambiguates* property, etc. Each Wikipedia page contains an *infobox* which describes the most important information about the topic in question. This data or information is structured in nature and hence is represented using named links to the source page.

Unstructured information also has connections to other information. However, they are represented as *anonymous* or *href links* with no meaning associated with them.

The ontology that we use for our summarization project is also formed from Wikipedia. It is a slightly modified approach of the DBpedia ontology creation process mentioned above. The Wikipedia ontology utilized in this thesis was constructed by Maciej Janik in the LSDIS lab at UGA [8]. DBpedia concentrates more on infobox data extraction. Other types of templates present within the Wiki page do not receive any such special handling. In DBpedia, for each template present in the page, a new entity is created in the ontology though the template contains connections that are directly linked from the page. The Wikipedia ontology eliminates the creation of these intermediate entities and creates direct connections instead.

Separate property names have been created to distinguish between direct names, disambiguations and redirections. In Wikipedia articles, disambiguation is achieved by adding contextual information along with the name of the entity in parenthesis. For example, ‘ontology’ and ‘ontology (information science)’. Phrases of this sort containing contextual information are not seen in documents since readers are capable of performing the disambiguation based on the document content. In the Wikipedia ontology, such names are shortened by removing the disambiguation information or contextual information and added as an alias name of a specific property to distinguish it from its full name.

CHAPTER 3

RELATED WORK

3.1 Overview of Summarization Methods and Systems

“For many purposes, coherence is not a crucial point, but relevance certainly is.” [3] In extractive summarization technique, we try to achieve *relevance* and concentrate less on *coherence*. Also, it has been observed that human summarizers tend to use parts of original sentences verbatim while forming summaries. This tendency would mean that human summaries are in part extractive in nature.

Some of the earliest work in summarization was done in 1958 by Hans Peter Luhn who proposed automatic construction of abstracts on the basis of statistical occurrences of words. He proposed that sentences be weighed based on the frequency of occurrence of words in the text disregarding all stop words. A few more methods in addition to the frequency-based weights method was used by Edmundson in 1969. In this system, term weights, along with being dependent on their frequencies, also depended on their location in the text, presence of the word in the title and presence in a cue phrase list which decides whether the term is to be given a high weight or a low weight. For example, the phrases ‘In conclusion’ and ‘To sum up’ would be given a higher weight because we would expect that sentences containing them would have some important information with respect to the document.

The ANES text extraction system by Brandow et al. (1995) used four major constituents

- *tf*idf* weight calculation for all terms
- selecting high *tf*idf* weight words and title words as signature words
- summing weights of terms over each sentence
- selecting high scoring sentences to be part of the final summary

Term frequency of a term or 'tf' is the number of times the term appears in the document. The intuition here is that more times a term appears in a document, greater its importance in the document.

Inverse document frequency of a term or 'idf' represents the importance of a word or a term in a document corpus. It is obtained by dividing the total number of documents in the corpus by the number of documents which contain the term. It balances out local frequencies which might augment the importance of a certain term just due to its high frequency in a single document.

It was observed that human readers tend to prefer *baseline* or *lead* summaries. These types of summaries contain the leading *n* words of the document. Baseline summaries are very readable when compared to system generated "intelligent" summaries, but have a much less relevance level in comparison to them.

All these above mentioned systems were extractive and used only statistical methods for summarizing documents without any understanding of the text. The drawbacks with extractive summary systems are – the loss of coherence and possible ambiguity in sentences. Ambiguity is introduced in extractive summaries due to, for example, dangling anaphoras.

Hence, a thorough processing of the text which removes all ambiguities is needed if we want a system which works much better than the statistical methods and heuristics used thus far.

3.2 Graph-based Summarization Methods

Currently, methods for deciding term importance have a very strong mathematical base. Graph-based ranking algorithms have become very popular in the process of deciding a textual unit's importance. Graph-based ranking algorithms are a way to decide the relative importance of a node within the graph. These algorithms take into consideration the global information, i.e., the whole graph, when deciding the importance of a node and not just the local, vertex-specific information. The same idea can be applied to a document. Importance of textual units can be decided taking into consideration the full document text instead of only the unit-specific information. We shall now discuss some systems that use graph-based ranking algorithms for summarization.

TextRank

The MEAD system [11] was a popular summarization system which was based on the idea that a sentence is more central to the document (or cluster) if it contains more words from the centroid of the document (or cluster). A centroid of a document (or cluster) represents a set containing all those words from the document/s which have *tf-idf* weights above a specified threshold. This centroid-based system was very promising and was the first multi-document summarization system to have been developed.

Graph-based methods have been gaining popularity in Natural Language Processing where salience of a textual unit needs to be calculated. The use of graph-based ranking algorithms in text summarization was first introduced by Rada Mihalcea from the University of North Texas in 2004. This technique used graph importance measures to extract sentences for the summarization task. A system called TextRank [17, 18] was created which, as the name suggests, was used to rank the sentences in a document according to some node importance measure.

The TextRank system uses three different importance measures or graph-ranking algorithms to decide node salience. The algorithms are – *Hubs and Authorities or HITS* [13], *Positional Power Function* [32] and *PageRank* [31]. All three algorithms mentioned here calculate node importance in an iterative fashion just as any other graph-based ranking algorithm. TextRank creates a graph where nodes represent sentences and edges are added between nodes and they specify the similarity value between the two nodes (sentences) it connects. Once the graph is ready, one of the above mentioned raking algorithms is run on the graph. Top ranked sentences are then chosen to form the summary.

LexRank

Another very popular summarization algorithm which made use of these graph-based methods was developed by Erkan and Radev from the University of Michigan, Ann Arbor [12]. The main idea in the paper is that sentences that are similar to many other sentences in the set are more central to the document. To define similarity between sentences, a feature-vector is created for each of the sentences. A feature-vector is made up of all the words present in the English language. The presence of a word in a sentence is denoted by including its frequency of occurrence as the corresponding dimension in the vector. The similarity between two sentences can then be calculated as a cosine between the two corresponding vectors. A cosine similarity matrix is later created for the sentence set which is a representation of how similar a sentence is to all other sentences in the set. The graph-based method is used here to find the salience of each of the sentences based on the similarity values that have been calculated. Each sentence is represented as a node of a graph. An edge is added between two nodes if the similarity value between them is greater than a specified threshold value. Once the graph is in place, several methods can be used to decide the salience of a sentence. If the degree of a node corresponding to a sentence is high, then that sentence can be considered central to the document (or cluster). Another measure known as Eigenvector centrality may be used to calculate the salience of a sentence. This system fared very well in the DUC 2004 evaluation. The DUC is the Document Understanding Conference which is a common platform where several summarization systems participate to be evaluated and ranked according to their performance. This paper also showed that degree-based methods outperform centroid-based methods in deciding importance of a textual unit.

Event-based Summarizers

Another novel method of summarization is *event-based summarization* which decides sentence salience based on events. An event is nothing but an occurrence or a happening. For example, “America Online bought over Netscape”.

One of the event-based summarization systems was built by Wenjie Li, Mingli Wu and Qin Lu from The Hong Kong Polytechnic University and Wei Xu and Chunfa Yuan from the Tsinghua University [22]. In this method of summarization, a sentence is important if it is describing significant events. The input document is turned into an “event map”. The event map contains named entities and events. Named entities are mostly nouns and pronouns. They can also be thought of as concepts or individuals. In the above example, “America Online” and “Netscape” are named entities. A named entity represents a person, an organization, a location or a date. An event usually represents a verb. A verb is considered an event only if it lied between two named entities. In the above example, “bought” is an event term since it lies between two named entities. In this way, a text document can be turned into an event map. This event map is essentially a graph where the named entities and event terms are the nodes. Relevance values are added to the connections in this graph and then the PageRank graph-based scoring algorithm is used to determine the significance of the nodes in the graph.

The relevance values are calculated based on intra-event and inter-event similarity values. Intra-event similarity refers the event-entity similarity and inter-event similarity is the event-event similarity. The intra-event similarity values are calculated based on the frequency of occurrence of events and inter-event similarity is calculated using the WordNet thesaurus. Once these relevance values are added to the graph, the PageRank algorithm is used to assign significant weights to the nodes in the graph. These significance weights are used then to decide the salience of sentences in the documents. If a sentence contains a highly significant event i.e. the significance weights of the corresponding nodes in the event map are high, the sentence is made part of the summary.

In this system, graph-based ranking is used to determine significance of events in a sentence which are then propagated to the sentences themselves. Hence, it is different in that sense from the aforementioned systems that use the ranking algorithms on the sentences directly.

Extractive Summarization Based on the Document Semantic Graph

Yet another system that uses graph-based ranking is an extractive summarization system which builds a *semantic graph* of the document and runs the ranking algorithms on it. This system was developed by Jurij Leskovec, Natasa Milic-Frayling and Marko Grobelnik at the Microsoft Research Center [21].

We are aware of the fact that abstracts of documents are created by rephrasing the original document. However, studies have shown that humans tend to extract parts of sentences from different sentences and put them together to form new sentences many a time. This shows that there exist relationships between entities in different sentences. This led to the idea that sentence clauses which are subject, predicate and object could be used to extract relationships among entities. The semantic graph constructed from the document contains triples of the form subject-predicate-object. A semantic graph is created by performing deep syntactic analysis of the document to extract these triples and represent them as a connected graph where the subjects and objects are nodes of the graph and predicates are edges between them. Support Vector Learning was performed on a document set, their corresponding human-generated summaries and the sets of triples generated for them to learn a model that distinguished between those triples that were part of the human generated summary and those that were not picked. This model was then used to automatically generate extractive summaries of input documents. The model was basically used to generate the semantic graph that would contain triples that have a high probability of being part of the summary if a human were to create it. The semantic graph obtained was then input to graph-based ranking algorithms such as HITS and PageRank to measure node importance values. Sentence salience was calculated by picking those sentences that had highest weights when their corresponding set of entity weights was summed.

This system is quite similar to our system since it uses a semantic graph of triples formed out of the document. The difference is in the manner in which the semantic graph has been created. In this system, deep syntactic analysis is performed to extract the triples from the document. We, on the other hand, extract triples from an existing ontology for the thematic graph formation along with performing syntactic analysis on the document text for entity relationship extraction.

Syntactic analysis of text may produce some false positives, meaning that the triples may not be accurate in meaning. To overcome this difficulty, the Support Vector Machine or SVM classifier is used to build a model. Our system is completely unsupervised in this respect.

We see from all the above extractive summarization systems that graph-based ranking has been common throughout. What differs from one system to another is how the graph is formed. Whether graphs were formed using sentences, events or entities, they all used one of the many graph-based ranking schemes to decide, directly or indirectly, the salience of sentences in the document.

3.3 Ontology-based Summarization Systems

We shall now discuss some of the recent systems that use ontological knowledge during the process of summarization. We shall look at the manner in which an ontology is used and also the extent to which it is used.

A Semantic Free-text Summarization System Using Ontology Knowledge [5]

This system comes from the collaborative work of researchers from the University of Houston and the University of Texas, Austin. The system uses knowledge from the Unified Medical Language System or UMLS ontology for extractive summarization of medical information. It provides a summary of the document based on the user-query. A semantically connected concept network is created for the original document. This network is constructed taking into consideration only those terms appearing in the ontology and ignoring the rest. The distance of each sentence in the document from the revised user-query is calculated and the ones with the least distance values are selected to be candidate sentences. Query revision is based on UMLS or WordNet ontology knowledge. Intra-pair distances are calculated for the candidate sentences and divided into groups based on some threshold value. Highest-ranked sentences are then selected from each group and put together as the final summary. Ontological knowledge is used here for refining the user-query as well as feature extraction from sentences. Though our system is quite similar in the use of an ontology, we use graph-based methods to calculate importance of sentences as opposed to finding sentence distance from the user-query. While this approach is a topic-driven our system generates generic summaries.

An Ontology-based Approach to Text Summarization [10]

This system is from Technische Universität Berlin, Germany. The main idea of this system is to classify sentences to a hierarchical ontology which captures the theme of the sentence and then calculate a similarity measure between the sentence and the document that it belongs to. For experimentation purposes the first two levels of the taxonomy from the Open Directory Project has been used. Each of the nodes is represented as a *tf-idf*-weighted-bag-of-words formed by processing words from the top 20 pages retrieved from the Yahoo! Search engine. A feature-vector is then created for each category or node in the taxonomy. Similarity measures are calculated for each sentence by taking the cosine distance between feature-vectors of the category and the sentence. The sentence is tagged with all those categories with a similarity measure greater than a specific threshold value. A bag-of-tags is calculated for the entire document by aggregating the bag-of-tags for each sentence. Finally, a sub-tree overlap measure is calculated as the dot product of the two bag-of-tags vectors for the document and the sentence which shows how well a sentence represents the document's content with respect to the ontology.

This system uses an ontology in the form of a taxonomy of categories to basically classify the sentences into different domains. Hence, the ontology acts as a classifier of sentences.

Our system uses ontological knowledge to a greater extent since it is used to form a mapping between the text and the ontology. It uses not only the top-level categorical information but also information about individuals in the ontology. Hence, the ontological meaning of the document is captured by the system.

CHAPTER 4

THE THEMATIC GRAPH

A thematic graph [2] represents the main theme of the document by creating a mapping between entity names or labels in the ontology and phrases present in the document. In other words, it is the thematic representation of the document in the form of a graph. It represents the entity name matches as the nodes of the semantic graph and the relationships between those nodes as edges in the graph. The thematic graph is a small sub-graph of the ontology used to create the mapping, since it captures the content present in a single document which talks about not more than a few topics. As mentioned before, we make use of the Wikipedia ontology for thematic graph creation.

Creation of the thematic graph is one of the most important steps in our summarization system. All the information required to decide the importance of sentences in the document comes from the thematic graph.

4.1 Thematic Graph Construction

Before actually starting the process of thematic graph creation, stemming and/or stop-word removal may be applied to the document. Stemming is the process of reducing different forms of a word into its root form. For example, a word ‘worker’ is reduced to its root ‘work’. Stop-word removal refers to removing very common words that occur in the English language such as ‘to’, ‘the’, ‘a’, etc. which do not add much to the document in terms of conveying important information. They mainly possess grammatical importance. Once either or both of the above mentioned processes have been applied to the document, we can move on to the thematic graph construction process.

The thematic graph construction process has 4 different phases as shown below.

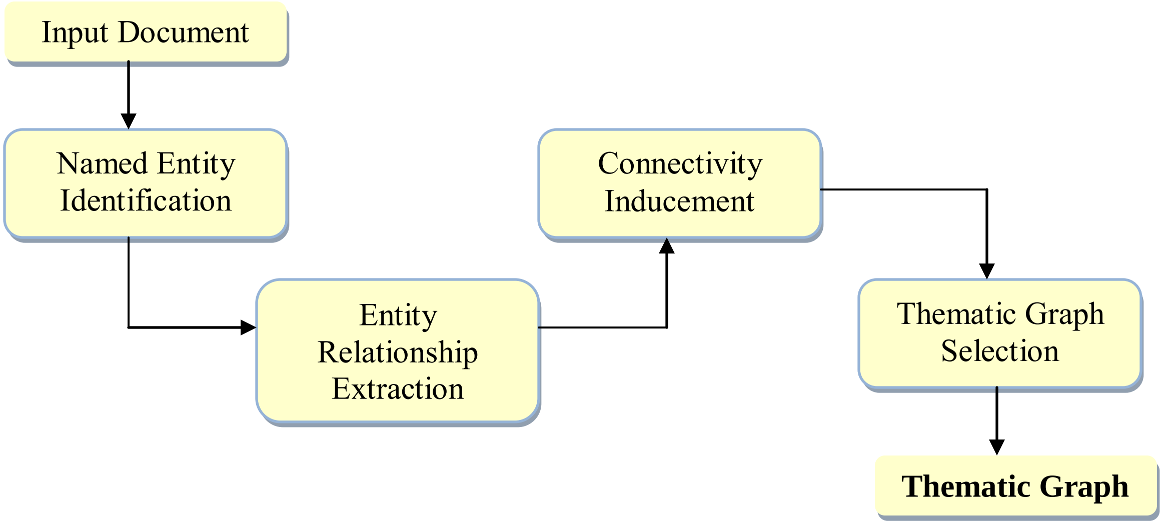


Figure 4.1 Thematic Graph Constructor Architecture

Let us look at an example which will help us understand the process of creation of the semantic graph. Consider the Wikipedia ontology. It consists of terms related to the game of tennis and also the names of the players and the tournaments for the game.

Consider the following piece of text: (*Copyright 2009 Associated Press*)

> **“*****US Open*** champion ***Juan Martin del Potro*** retired because of right wrist tendinitis while trailing ***Jurgen Melzer*** of Austria 7-5, 2-1 Wednesday at the Shanghai Masters.
>
> The third-seeded Argentine, who was shaking his right hand before packing up his rackets, said that he had similar wrist tendinitis this year.
>
> "I'm a little sorry," Del Potro said. "It's a big tournament here in Shanghai, very important for me. But if I want to have a good finish this season, I have to recover, go home to be in good shape for the last tournaments."
>
> Del Potro has already qualified for next month's season-ending ***ATP*** tournament in London.
>
> Top-seeded ***Rafael Nadal*** and second-seeded ***Novak Djokovic*** also advanced. ***Nadal*** defeated ***James Blake*** 6-2, 6-7 (4), 6-4 -- the second straight week in which the ***Spaniard*** needed three sets to defeat the American.**”**

Below is the thematic graph that has been created for the above piece of text. It contains the entities identified, the relationships between them in the ontology and the weights calculated for each of them.

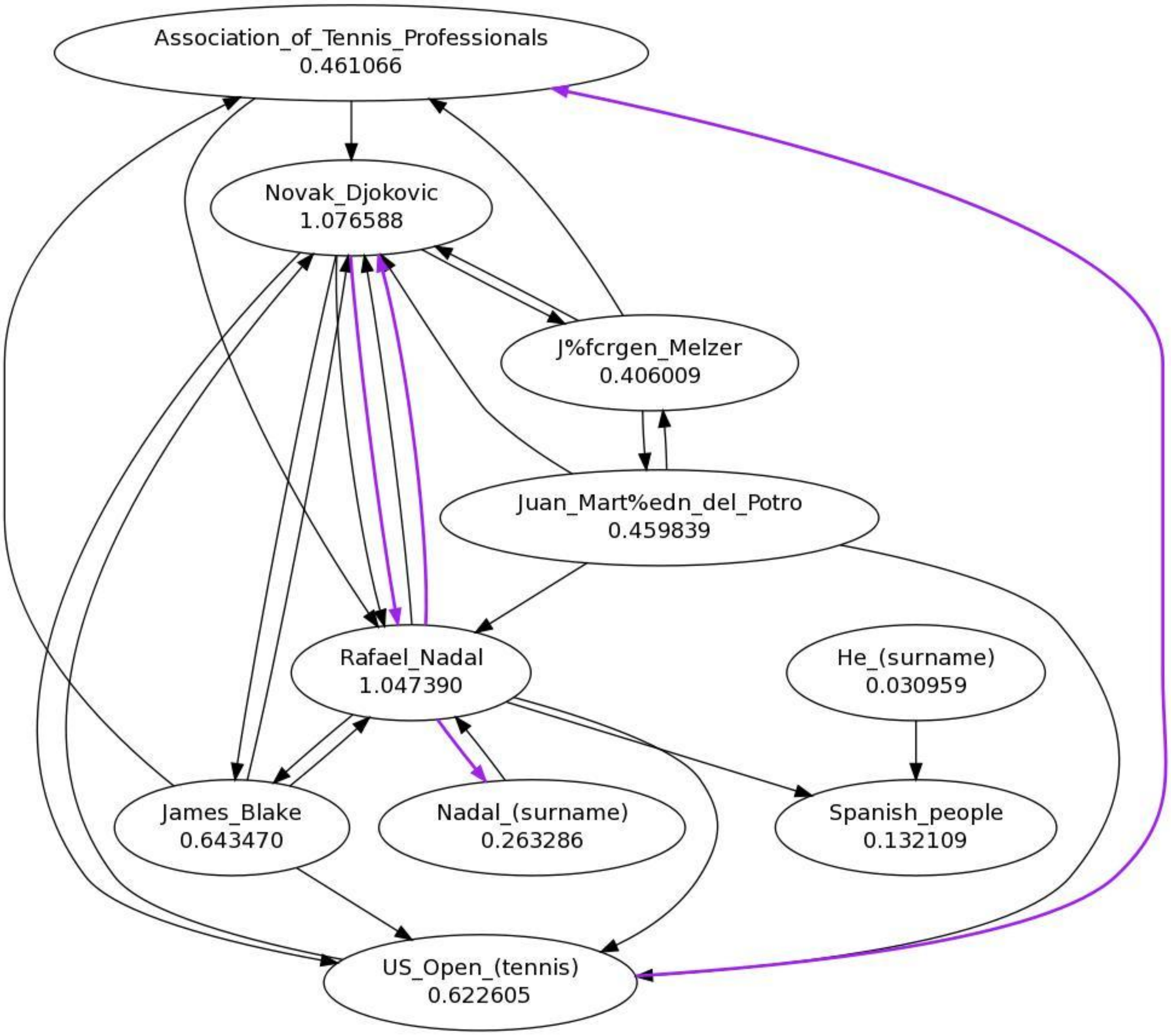


Figure 4.2 An Example Thematic Graph

Below is a step-by-step procedure for creating the above thematic graph.

Named Entity Identification

During this first phase of the thematic graph construction process, all named entities present in the document are identified by matching phrases in the document with the names of entities present in the ontology. Entity names in an ontology are usually values of a certain property. These properties identify the name of an entity and also any synonyms for the name.

In the above example, the underlined phrases are the terms that have been matched to entities in the Wikipedia ontology. Phrases such as *'Rafael Nadal', 'US Open', 'ATP'*, etc. all represent entities that are connected with the game of tennis. These phrases form the nodes of the initial graph. Each node is assigned a weight based on certain factors.

Node Weight calculation

Once all the matching entities have been identified, the next step is to assign weights to all of them. The weight of a node is calculated based on factors such as the property which is used to identify an entity (name or alias), the exactness of the match to the label and the frequency of occurrence of the matched entity. Multiple occurrence of the phrase suggests its importance within the document and hence term frequency is also considered when assigning weights to the nodes. As the frequency of a matched entity increases, it gets reflected in the weight for the entity. To prevent the weight from increasing drastically for entities that occur too often, the formula shown below is used.

$$w = 1 - \frac{1}{1 + \sum_{i=1..n} p_i * s_i}$$

w - is the initial weight of the node

n - represents the total number of matches found in the document

p_i – is the weight of the property that connects the literal value with the entity

s_i – is the similarity measure between the match which takes into account any differences introduced due to stemming and/or stop-word removal. If neither is used, it is set to 1.

Shown below is a figure depicting the relative importance of the various properties (p_i). If the property that connects the literal with the entity is of type ‘wiki_name’, the weight of the corresponding node is greater than if the property type was ‘wiki_redirect_name’. Hence, an exact match is given more importance when compared to an alias or a synonym.

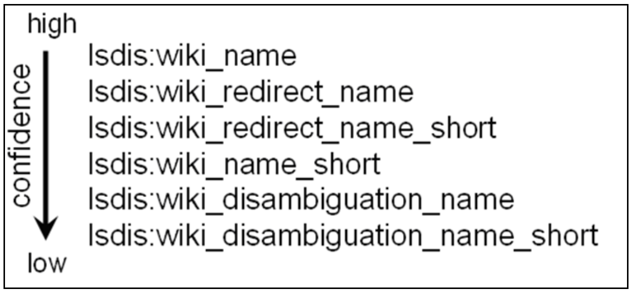

high
confidence
low
lsdis:wiki_name
lsdis:wiki_redirect_name
lsdis:wiki_redirect_name_short
lsdis:wiki_name_short
lsdis:wiki_disambiguation_name
lsdis:wiki_disambiguation_name_short

Figure 4.3 Property Confidence Levels

Redirection

Wikipedia employs a concept called ‘redirection’ to direct a search for a specific phrase to a page with a title which is a more common and popular name for the search phrase. For example, a search for ‘USA’ takes you to a redirection page for ‘United States’. Hence, if the document contains the term ‘USA’, the weight of the corresponding semantic graph node is lower than if the document contained the phrase ‘United States’ i.e. redirection links are attributed lower weights when compared to a direct match for the search phrase. Another example from the above piece of text is the term ‘ATP’ leading to a redirection page for ‘Association for tennis professionals’.

Disambiguation

To disambiguate between entities having the same name, Wikipedia uses a concept known as ‘disambiguation’. There exist entities which may share the same name but mean different things based on what context they are being used in. Hence, a phrase in the document could be matched to many different entities in the ontology. Wikipedia includes the context information in brackets next to the entity name. For example, ‘US Open’ in the context of tennis is denoted as ‘US Open (tennis)’ since US Open could represent the tournament for several different sports such as tennis, golf, bowling, etc. It is very unlikely that we will find a phrase that matches an entity of this sort in any document, since a reader can disambiguate based on the context of the document itself. Hence, these types of matches are given lower scores when compared to exact matches.

Due to the fact that a single phrase from the document can match multiple entities in the ontology, the total number of entities may be greater than the number of matched phrases in the document. The falsely identified entities and weakly related entities are eliminated from the semantic graph at a later stage.

Entity Relationship Extraction

Along with relationships present in the ontology, sentences in the text are parsed in order to obtain more relationships that might exist within them. An NLP parse of each sentence is obtained to obtain its dependency tree. NLP stands for *Natural Language Processing*. Obtaining an NLP parse of a sentence refers to describing the sentence structure with rules that are applicable to a natural language. This description of the sentence is called a dependency tree. Phrases that have been previously matched are used as hints for entity positions. If matched phrases are close in the parse tree, an unnamed relationship is added between them in the graph. Hence, the ontology is not the only source of relationships for the thematic graph. The inherent relationships in the text are also extracted and added to the graph.

Connectivity Inducement

All relationships existing between the nodes in the ontology are now added to the semantic graph. Each relationship in the ontology schema has an importance level indicating the information it represents as shown in Figure 4.4. We now propagate and recalculate the weights of entities in the created graph using the HITS algorithm.

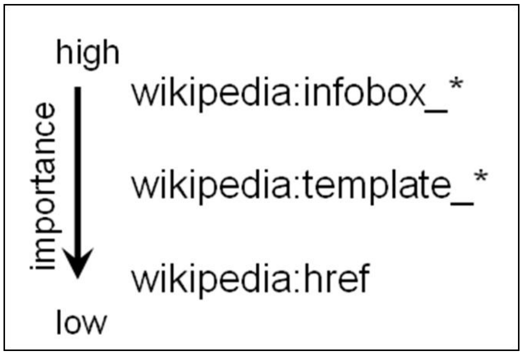


Figure 4.4 Relationship Importance Levels

Thematic graph selection

The graph obtained from this process may cover several topics. Also, during the Named Entity Recognition phase, several unrelated entities might have been added to the semantic graph and entities with the same names but different contexts may have been identified. Hence, the graph may have several connected components. The component with the most number of entities and having the maximum weight is chosen as the dominant sub-graph. This selection is intuitive since the largest component contains the most number of terms from the document and also because entities belonging to related topics are connected within the ontology. Hence, this selection best describes the theme of the document. The selection of the dominant sub-graph also eliminates any false positives and unrelated entities identified during the Named Entity Recognition stage. Hence, the final *thematic graph* obtained is devoid of any unrelated or even weakly related entities and focuses on the core topic in the document.

The thematic graph is then rendered in the form of an XML document which contains the nodes and the relationships between them. It also provides information about the positions of occurrence of the terms within the document. Each node is associated with a particular weight suggesting its worldly/general importance.

A small portion of an example thematic graph in the form of an XML for the above mentioned piece of text is shown in the following page. The first part of the XML lists the nodes of the graph. Some of the nodes identified here include ‘Association of Tennis Professionals’, ‘Rafael Nadal’, ‘US Open’, etc. Here, we observe that though the article does not have the expansion of ‘ATP’, it has been redirected to it and hence, the expansion is identified as one of the nodes.

The second part of the XML lists all the edges in the graph. Each edge is represented by a triple – (subject, property, object). The first edge shows that there exists a relationship between ATP and Rafael Nadal. The property connecting the two nodes is an anonymous href link. Most of the links in the Wikipedia ontology are anonymous href links. There is no meaning attached to these associations. Meaningful names or non-anonymous links are present for some associations which involve nodes represented in a structured format. For example, the information present in the infobox of each Wikipedia page or in tables, etc. has meaningful links. In the below example, the link – 'wiki_template_succession_box_after' connects Rafael Nadal with Novak Djokovic implying that in a particular succession box, Rafael Nadal is a successor of Novak Djokovic.

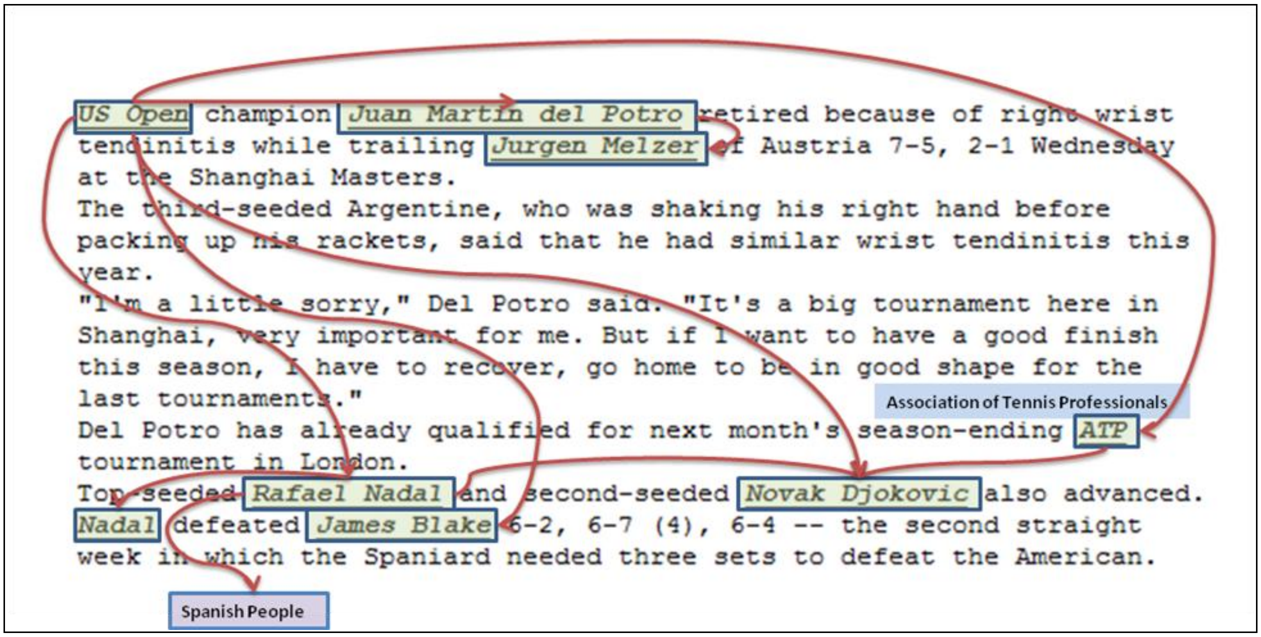
```
<semgraph>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FAssociation_of_Tennis_Professionals" weight="0.461067">
  </node>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FHe_(surname)" weight="0.030959">
  </node>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FJ%25fcrgen_Melzer" weight="0.406008">
  </node>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FJames_Blake" weight="0.643470">
  </node>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FJuan_Mart%25edn_del_Potro" weight="0.459839">
  </node>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FNadal_(surname)" weight="0.263286">
  </node>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FNovak_Djokovic" weight="1.076589">
  </node>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FRafael_Nadal" weight="1.047389">
  </node>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FSpanish_people" weight="0.132109">
  </node>
<node resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FUS_Open_(tennis)" weight="0.622605">
  </node>

<edge>
  <subject resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FAssociation_of_Tennis_Professionals"/>
  <property resource="lsdis%3Awiki_href"/>
  <object resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FNovak_Djokovic"/>
</edge>

<edge>
  <subject resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FRafael_Nadal"/>
  <property resource="lsdis%3Awiki_template_succession_box_after"/>
  <object resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FNovak_Djokovic"/>
</edge>

<edge>
  <subject resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FUS_Open_(tennis)"/>
  <property resource="lsdis%3Awiki_href"/>
  <object resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FNovak_Djokovic"/>
</edge>

<edge>
  <subject resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FUS_Open_(tennis)"/>
  <property resource="lsdis%3Awiki_template_us_open_series_tournaments"/>
  <object resource="http%3A%2F%2Fdbpedia.org%2Fresource%2FAssociation_of_Tennis_Professionals"/>
</edge>
</semgraph>
```



US Open champion Juan Martin del Potro retired because of right wrist tendinitis while trailing Jurgen Melzer of Austria 7-5, 2-1 Wednesday at the Shanghai Masters.

The third-seeded Argentine, who was shaking his right hand before packing up his rackets, said that he had similar wrist tendinitis this year.

"I'm a little sorry," Del Potro said. "It's a big tournament here in Shanghai, very important for me. But if I want to have a good finish this season, I have to recover, go home to be in good shape for the last tournaments."

Del Potro has already qualified for next month's season-ending ATP tournament in London.

Top-seeded Rafael Nadal and second-seeded Novak Djokovic also advanced.

Nadal defeated James Blake 6-2, 6-7 (4), 6-4 -- the second straight week in which the Spaniard needed three sets to defeat the American.

Figure 4.5 Mapping between text and Wikipedia ontology

The above figure shows pictorially the mapping between the input text and the Wikipedia ontology. The boxes in green represent entities present in the ontology that are matched to the terms in the text document. The directed arrows in red show the relationships between the entities as they are in the ontology. The box in purple represents the ‘Spanish people’ entity that is not present in the text but got picked up during the formation of the thematic graph. The entity ‘ATP’ present in the text may refer to many different nodes in the ontology. Disambiguation is performed on it based on the context and the correct entity in the ontology, which is ‘Association of Tennis Professionals’ in this case, is chosen by the system.

CHAPTER 5

ONTOLOGY-BASED SUMMARIZATION

Most extractive summarization systems hitherto have used statistical metrics to find the most informative sentences in a document. Statistical metrics do not capture the meaning or the knowledge present in the document. Take for example a document which talks about ‘diabetes’. The document might introduce diabetes by name in the first sentence and then go on to refer to it as ‘this disease’ or ‘it’ in the rest of the document. If a statistical measure such as term frequency is used, the term ‘diabetes’ is not attributed as much importance as it is supposed to be attributed since it occurs only once in the document. To overcome such shortcomings, in our system, we use a knowledge-rich approach to summarization. The source of the knowledge is the Wikipedia ontology introduced in the previous sections. The original document and the ontology are used to obtain the Thematic Graph for the document. The Thematic Graph which represents the document in the thematic sense provides the information or the knowledge used to choose sentences which are considered to be most informative in the document. In the ‘diabetes’ example mentioned above, the term ‘diabetes’ is assigned a certain weight in the semantic graph as described in the previous sections. Hence, a term’s importance is determined by how important it is in the document with respect to the ontology rather than just a statistic.

5.1 System Architecture

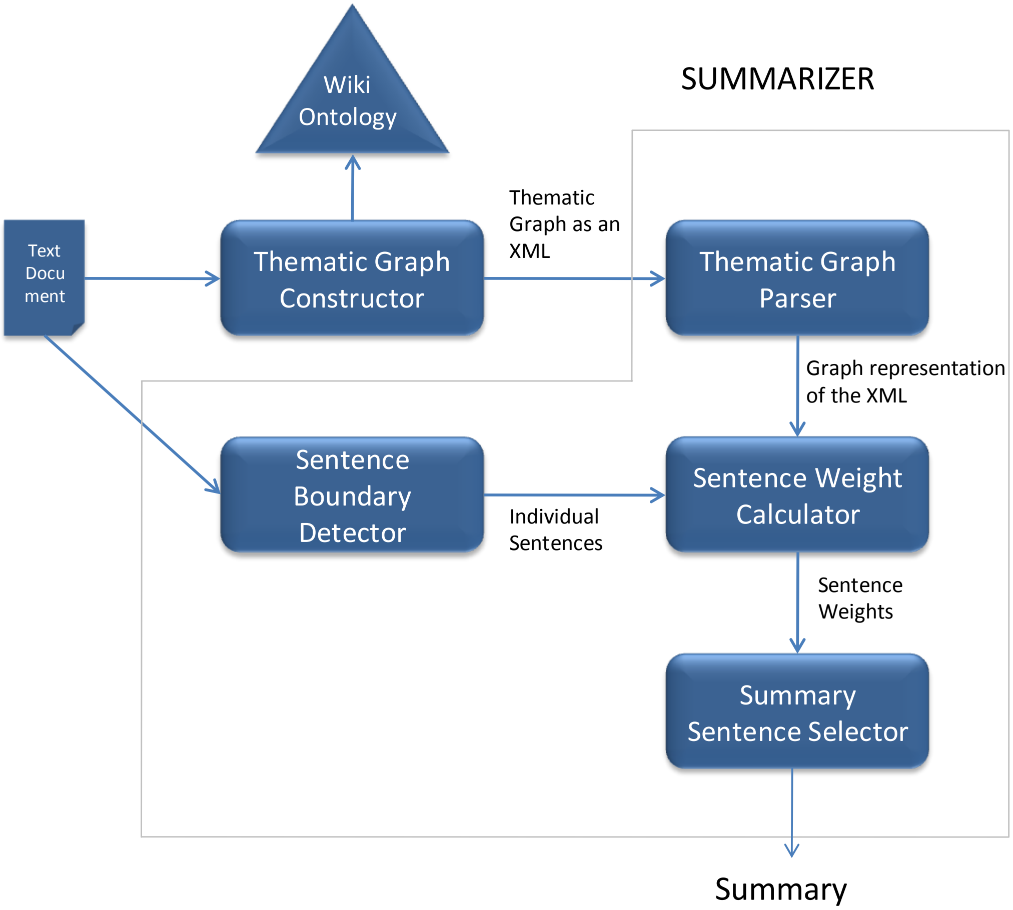

Wiki Ontology
SUMMARIZER
Text Docu ment
Thematic Graph Constructor
Thematic Graph as an XML
Thematic Graph Parser
Graph representation of the XML
Sentence Boundary Detector
Individual Sentences
Sentence Weight Calculator
Sentence Weights
Summary Sentence Selector
Summary

Figure 5.1 Summarization System Architecture

The above figure shows the architecture of our summarization system. All modules except for the Thematic Graph Constructor and Sentence Separator were implemented for this project.

The various modules of the system are described in the following sections:

Thematic Graph Constructor

The input to the Thematic Graph Construction module is the document for which a summary needs to be formed. The Thematic Graph Constructor makes use of the Wikipedia ontology and creates a mapping between the ontology and the input document which is nothing but the thematic graph. The details regarding the graph construction have been described in the previous chapter. The output is the thematic graph in the form of an XML document also described in detail in the previous section.

Sentence Boundary Detector

The Sentence Boundary Detector used in our project was created by Scott Piao from University of Lancaster, UK [23]. The sentence boundary detector is an important and crucial component for an extractive summarization system. Without correct recognition of individual sentences, the quality of the summary will be compromised severely. We used the above mentioned sentence separator software as we found it to be the most reliable. The original document is input to this component which breaks down the text into individual sentences and placed in a sentence data structure. The correctness of sentence boundary detection is crucial to the summarization system. Incorrect splitting will lead to the formation of a summary that makes very little to almost no sense. Hence, the reliability of the sentence detector is very important.

Thematic Graph Parser

The thematic graph in the form of an XML document forms the input to the Thematic Graph Parser. The parser parses the XML document using a DOM parser and converts it into a graph with nodes and edges. Since our source of information is in the form of a graph, we use various importance measures to determine importance of each of the nodes in the graph. Centrality measures such as degree centrality, Eigenvector centrality and Barycenter centrality measures are used by our system. The various importance scores for each node in the graph are calculated in this phase. Entities that are central to the document are assigned higher scores when compared to entities in the document that are weakly related.

Sentence Weight Calculator

Each node in the thematic graph comes with a weight associated with it. These weights, as explained previously, are based on factors such as the exactness of the match, the number of times it appears in the document, etc. Other importance measures as obtained from the previous step are also associated with each node.

Several different sentence weights are measured during this step. For each importance measure, individual sentences are associated with the sum total of the weights of all the nodes that appear in that specific sentence. The higher the weight, the greater is its importance to the document. For example, if a sentence has four nodes from the thematic graph, the individual weights for those four nodes is summed and the total weight is designated as the sentence weight. Hence, a sentence with a high weight implies that the sentence contains terms that are central to the document.

Summary Sentence Selector

With each sentence having been associated with a weight, we now decide which sentences will be part of our summary. Sentences with the highest weights are chosen. The sentences are first arranged in descending order of their weights. Top-ranked sentences, i.e., the ones with the highest weights are chosen until the word limit for the summary is reached. The sentences are then arranged in the order of occurrence in the original document to form the final summary. Additionally, the first sentence of the document is added to the summary if not already chosen by the system. The intuition behind doing this is that many times, especially in news articles, the first sentence conveys a lot of information and sometimes even gives a single-line summary. The first paragraph in a well written article introduces the subject and hence, a good writer will make an informative first sentence.

5.2 Node Importance Measures and System Modules

Importances of nodes in the thematic graph are measured using several graph-based ranking methods. These importance measures are known as centrality measures. The centrality measure of a vertex determines its relative importance within the graph. There exist several centrality scoring algorithms. The measures mentioned below are some of the few we use in our summarization system.

Thematic Graph Weight module

As explained in the previous section, each node in the semantic graph is assigned a weight depending on several factors. These weights signify the importance of a phrase with respect to the document and are hence used as one of the importance measures to calculate sentence weights. This is the only module which uses an entity scoring method which is not a graph-based ranking algorithm.

Degree centrality module

Degree Centrality is defined as the number of edges connected to or incident on a node. In other words, degree of a node is the sum of the fan-in and fan-out values of the node. A node with a high degree centrality score implies that the entity is related to several other entities in the graph and hence is an important and popular entity within the graph. Most of the nodes in a thematic graph can be found in the document on which it is based. Hence, they are all already connected to each other. The degree centrality of a node represents *how well* an entity is connected to the other entities in the document. For example, consider a document talking about football. The thematic graph will contain most certainly an entity for 'football'. Based on the document itself, it may also possibly contain names of football players, coaches, football clubs and other football related terms. Many of these terms are connected to the 'football' entity i.e. 'football' has a high degree centrality value. This makes sense because football is the document's main topic. Hence, the presence of that term in a sentence makes the sentence important to the document.

The Degree Centrality module uses the degree centrality scores of each of the nodes in the thematic graph to compute sentence weights which ultimately decides the inclusion of a sentence in the summary.

Eigenvector centrality module

A more sophisticated version of the Degree Centrality measure is the Eigenvector centrality. Here, the centrality measure of a node is not only dependent on the number of edges but also on the weights of those edges. Eigenvector measure of a node is calculated using the principle that connections to high-scoring nodes contribute more to the score of the said node than low-scoring nodes. Hence, a node with a small number of heavy-weight connections may have a higher score when compared to a node with a large number of low-weight connections. Extending the football example we saw in the previous section, let us also assume that another main theme of the article is a football injury and hence several medical terms connected to the injury are also part of the thematic graph. It may be so that the article concentrates more on the injury caused due to football than on football itself. Therefore, though the 'football' entity may be connected to quite a few other nodes in the thematic graph, the nodes that it is connecting to may not be high-scoring nodes. However, the 'football injury' entity, even if it does not connect to too many nodes, connects to nodes which are high-scoring. We can think of this as, the 'football' entity connected to many redirection or disambiguation nodes and the 'football injury' node connected to a few direct nodes making it more important to the document than the 'football' node.

Barycenter centrality module

The *Distance Centrality* score of vertices is calculated based on their distances to each and every other vertex in the graph. There are 2 types of distance centrality scores – *Closeness Centrality* and *Barycenter Centrality*.

Closeness of a vertex within a graph is higher if it has short geodesic distances to all other vertices in the graph. Geodesic distance is calculated as the number of edges in the shortest path connecting the two vertices. If there exists no path between the two vertices i.e. they belong to different connected components, then the geodesic distance between them is infinity.

Closeness scores are calculated using the formula - 1 / (average distance from vertex v to all other vertices) and Barycenter scores are calculated as - 1 / (total distance from vertex v to all other vertices). Hence, Barycenter scores are assigned to each vertex according to the sum of its distances to all other vertices. If the total sum of the distances to all other nodes in the graph is high, it implies that the vertex is a non-central entity in the graph, since that node is not directly connected to many other entities in the graph. For example, in Figure 4.5, the 'Spanish people' entity is far away from most other nodes in the graph and hence has a low Barycenter score. This makes sense because the article is not concentrating on Spanish people and it is weakly related entity in the document.

HITS module

Hyperlink-Induced Topic Search (HITS) [13] also known as Hubs and Authorities is a popular algorithm used to determine node importance. It is mostly used in determining the importance of web pages on the Internet. The HITS algorithm measures the "hubness" and the "authority" of each node in the graph. "Hubness" of a node is a defined in terms of the authorities of the outgoing nodes and "authority" is defined in terms of the hubness of the incoming nodes. Therefore, a node has a high hubness value if it has many outgoing edges leading to nodes with high authority values and a node has a high authority value if the hubness values of its incoming nodes are high. Hubness of a node depends not only on the number of nodes that it points, but also on the authority of those nodes. A high authority value shows that the node is an important because several other highly connected nodes are pointing to it. A high hubness value shows that the node is important because it has connections to many other high authority nodes in the graph.

The Hub score and Authority score for a node is calculated using the following algorithm [27]:

- Start with each node having a hub score and authority score of 1.
- Run the Authority Update Rule - Update each node's *Authority score* to be equal to the sum of the *Hub Scores* of each node that points to it.
- Run the Hub Update Rule - Update each node's *Hub Score* to be equal to the sum of the *Authority Scores* of each node that it points to.
- Normalize values by dividing each Hub score by the sum of squares of all Hub scores, and dividing each Authority score by the sum of the squares of all Authority scores.
- Repeat from the second step as necessary

Some points to note

- *Sentence Length Normalization:* You may have observed the absence of any sort of normalization being applied to prevent promotion of long sentences. This has been done intentionally because of the manner in which we decide the importance of sentences. Sentence weight depends not on any statistic such as term frequency which without any normalization would encourage the system to choose longer sentences. It depends rather on the importance of terms that are found to match entities in the ontology. If a sentence weight is high, it is not because it is longer, but because it contains several matching entities from the ontology and is hence an important sentence in the document.

- *Application of graph-based ranking algorithms:* Unlike most of the summarization systems using graph-based ranking, we use these algorithms on the thematic graph that we create and not on the sentence graph. In fact, we do not create a sentence graph at all. Graph-based ranking algorithms are used to find importance of individual phrase matches and these scores are later used to determine sentence importance. Many other systems use graph ranking to determine importance of sentences directly by applying the algorithms on a sentence graph.
- *Anaphora Resolution*: The problem of dangling anaphoras is lesser in our system due to the inherent nature of the method which chooses sentences based on entities present in them. Consider for example the following two sentences – "Roger Federer played in the US Open. He went on to win it". Our system would rank the first sentence as being more important due to the presence of entities "Roger Federer" and "US Open". But this may also lead to loss of important information. In the above example, if anaphora resolution had been applied we would have the following text – "Roger Federer played in the US Open. Roger Federer went on to win US Open." Hence, we claim that addition of anaphora resolution component to the system can not only improve the quality of the summary in term of "making sense" but also in terms of improving the content in it.

CHAPTER 6

IMPLEMENTATION

The summarizer has been implemented in the Java programming language. Some of the libraries that were made use of were the Java Universal Network/Graph or the JUNG Framework [25] and a sentence boundary detector created by Scott Piao from the Dept. of Linguistics and MEL, University of Lancaster in UK [23].

The Thematic Graph Constructor is hosted on the *kronos* server at UGA's LSDIS lab. A document that has to be summarized is read by the program and sent to the kronos server which returns the thematic graph for the input file in the form of an XML document.

The sentence boundary detection software [23] used has been created by Scott Piao. The program is written in Java and is a very reliable program for identification of sentence ends. The system was evaluated by Scott Piao and Yoshimasa Tsuruoka on the Genia corpus and it achieved a precision of 0.997352 with a recall of 0.995093. An online endpoint for the system can be found at http://text0.mib.man.ac.uk:8080/scottpiao/sent_detector.

The other system that we also tried was the Sentence Boundary Detector from the *Lingpipe* [24] system. Lingpipe is a suite of Java libraries for the linguistic analysis of human language. It provides several Natural Language Processing (NLP) tools such as named entity recognizer, parts of speech tagger, word sense disambiguator among several others.

We found during the course of the thesis that Lingpipe's sentence boundary detector is reliable for simple sentences. It fails to detect ends of sentences in cases where the sentence ends with a double quote. This occurs when quoting a person on some topic. Take for example the below piece of text from one of the articles in the DUC document set –

❝ "He's just a part of the community," said Curt Loyd, executive director of the Bentonville Chamber of Commerce. "He's a neat person." Walton prefers bird-hunting, tennis and opening new Wal-Mart stores. ❞

The portion of the above text encircled in red shows the place where a sentence end is present. But due to the presence of the double quote at the end of the first sentence, the boundary goes undetected.

It also sometimes identifies false positives. Described below are two situations where this occurs.

1) Abbreviations which are short-forms for certain words and which end with a period are identified as sentence ends. For example, the single sentence - *"Mr. Smith goes to Washington."* gets split up into *"Mr"* and *"Smith goes to Washington"*.
2) Person names with initials with a period after them are also identified as sentence ends. For example, *'Krys J. Kochut'* is split up into *'Krys J'* and *'Kochut'*.

The individual sentences obtained are stored in individual objects of a sentence data structure array. For each sentence, the total sum of the importance values for all the terms in the sentence that occur in the thematic graph is calculated and stored along with it.

Java Universal Network/Graph Framework or JUNG [25] - is a software library that provides a common and extendible language for the modeling, analysis, and visualization of data that can be represented as a graph or network. The distribution of JUNG includes implementations of a number of algorithms from graph theory, data mining, and social network analysis, such as routines for clustering, decomposition, optimization, random graph generation, statistical analysis, and calculation of network distances, flows, and importance measures such as centrality, PageRank, HITS, etc. We used the JUNG package for graph representation of the XML and to run algorithms to calculate the various importance measures for the graph.

Once the thematic graph was obtained in the form of an XML document, the XML-DOM library along with JUNG was used to parse the XML and represent it in the form of a directed graph. The graph nodes contain the names of Wikipedia articles and the weights associated with them. Various importance calculation algorithms such as Degree Centrality, Eigenvector Centrality, etc. available in the JUNG package are run on the semantic graph obtained and these importance measures are also associated with the graph nodes. These importance values are made use of to find the most important sentences in the document.

The summarization system is divided into 5 different modules, one each for every node importance measure used.

CHAPTER 7

EVALUATION AND EXPERIMENTS

7.1 Evaluation Corpus

The document set used for evaluation of our system is the *Document Understanding Conferences* (DUC) [28] data from the year 2002. Sponsored by the Advanced Research and Development Activity (ARDA), the DUC conference series is run by the National Institute of Standards and Technology (NIST) to aid researchers in large-scale evaluation of summarization systems. The DUC has been run annually from 2001 [7] and has provided a common test bed for all automatic summarization systems.

The DUC included four main tasks [6, 7]:

- *Generic single-document summarization task:*

  In this task, all participating systems were asked to create 100-word summaries for each document in the 30-topic DUC document set. Single-document summarization task was held in 2001 and 2002 only. It was dropped after that.

- *Generic multi-document summarization task:*

  This task involved creation of multi-document summaries of sizes 10, 50, 100, 200 and 400 words for each document set related to a specific topic.

- *Headline generation:*

  This task refers to creating headlines which can be thought of as very short summaries for documents. The summary is required to capture the main idea in the document in a single sentence.

- *Topic-driven summarization:*

  The focus here is on generating a summary that is focused on a specific topic in the document.

The DUC document set consists of a sets of documents related to various topics. Summaries are created for each document (in case of single document summarization) or a document set (in case of multi-document summarization) by human summarizers. The document set contains around 10 documents for each of the 60 topics. Each human summarizer was to create summaries for 10 topics out of the 60. The 2002 DUC data includes single document as well as multi-document summaries. We made use of the single document summaries to evaluate our system.

The DUC 2002 document set was cleaned up slightly before using it. Very short documents were eliminated and not used in the testing process. Also, we noticed that some documents used a semi-colon instead of a period to identify sentence ends. These types of documents were modified such that periods identified sentence boundaries. Out of the 567 documents originally present in the DUC document set, we were left with 556 documents after the clean-up process.

7.2 Automatic Summary Evaluation using ROUGE

For automatic evaluation of our system generated summaries against human created gold standard summaries from DUC, we used a popular system known as *ROUGE* [29]. ROUGE expands to *Recall Oriented Understudy for Gisting Evaluation* and is basically used to measure the quality of system-generated summaries. DUC data along with the ROUGE system has been a popular combination for evaluation of automatic summarization systems.

The ROUGE system calculates several measures for evaluation of system-generated summaries or peer summaries against human-generated summaries also known as model summaries. These measures show how well the peer summaries correlate to the model summaries and the measures are based on many different methods.

The various ROUGE metrics include [6]:

- ***ROUGE-N***: measures the N-gram co-occurrences between the peer and the model summary. N-gram refers to the number of phrase matches of length N between the two summaries. N can take any integer value. Greater the value of N, greater is the *fluency* of the peer summary. ROUGE-1 measures the number of single term matches found in the peer summary when compared with the model summary.

$$\frac{\sum_{S \in \{Referemce\,Summaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{Reference\,Summaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

The ROUGE-N score is calculated using the above formula where ‘n’ is the length of the n-gram, $gram_n$ and $Count_{match}(gram_n)$ is the maximum number of n-grams matched between the peer and model summaries.

- ***ROUGE-L***: 'L' stands for Longest Common Subsequence (LCS). Given a pair of sequences, the LCS of the two is a common subsequence with the maximum length. The LCS provides a measure of *similarity* between two given sequences.

- ***ROUGE-W***: 'W' here stands for Weighted Longest Common Subsequence. It is a variant of the ROUGE-L measure which takes into consideration the fact that some matches are consecutive in nature and hence they should be given a higher weight.

- ***ROUGE-S***: A Skip-bigram is any pair of words or terms in a peer summary sentence that occur in the same order in the model summary. The Skip-bigram metric measures all of the pairs that are in correct order ignoring any gaps between them. Hence, every pair of co-occurrence makes an impact when calculating the quality of the summary unlike LCS which takes into account only the longest common subsequence. ROUGE allows us to specify a skip distance between a pair of terms that are allowed to become skip-bigrams.

- ***ROUGE-SU***: is an extension of the ROUGE-S metric. Some sentences do not have any matching word-pairs and hence ROUGE-S discounts them completely. To avoid this from happening, ROUGE-SU considers unigrams also.

A Simple Example

Let us consider an example to illustrate score calculation for some of the above mentioned metrics. Consider the following three sequences, one from the peer summary and the last two from the model summary:

From peer: police killed the gunmen

From model: S1 - police kill the gunmen

From model: S2 – the gunmen kill police

The ROUGE-N score = 2 for both S1 and S2. The n-grams are – (‘police’, ‘the gunmen’).

The ROUGE-L score for S1 = 3/4 since LCS = ‘police the gunmen’.

For S2, ROUGE-L score is 2/4 since LCS = ‘the gunmen’.

The ROUGE-S score for S1 = 3/6 where the skip-bigrams are – (‘police, the’, ‘police, gunmen’, ‘the, gunmen’).

The ROUGE-S score for S2 = 1/6 where the skip-bigrams are – (‘the, gunmen’).

Studies conducted by the ROUGE team have shown that using unigram co-occurrences i.e. ROUGE-1, for automatic evaluation of summary pairs correlates very well human evaluations [14]. The same measure has been used in conducting our experiments.

Many of the extractive summarization systems so far have used the ROUGE system to measure quality of their system-generated summaries. Therefore, the use of ROUGE in this work has made it feasible to compare our system with other ROUGE-using systems.

7.3 Evaluation Setup

Since the DUC corpus that we are dealing with is quite large with 556 documents, we setup a testing environment to run experiments and evaluate our system modules. The test environment had three different components –

- Thematic Graph Formation component
- Summary Generation component
- ROUGE Evaluation component

Thematic Graph Formation component

The system which generates the thematic graph for an input document is present on the *kronos* server of the LSDIS lab at UGA. The modules are designed such that an HTTP request is sent to the server and an XML file is received as output. To avoid sending requests for the same input document multiple times for each module, the thematic graphs or XMLs for all the documents in the DUC document set were retrieved and stored for use by all summarization modules.

Summary Generation component

The DUC test documents are all in XML format where the text of the article is present in the <TEXT> field of the XML. A simple extraction program was used to extract the text of the article from the XML and create a simple text file containing the article's text.

The extracted text documents were input to each summarization module and their summaries were obtained. The thematic graphs that were obtained in the previous step were utilized in the creation of the summaries. The system took approximately a minute and a half to form all the 556 summaries.

The human generated summaries in the DUC set are also part of an XML file which contains all the summaries for a specific document set and also specifies which human summarizer created the summary. Again, a simple extraction program was used to extract each summary along with the summarizer ID and create a text file with the summarizer ID appended to the end of the filename for the summary.

ROUGE Evaluation component

The ROUGE system requires as input an XML file that specifies, among other things, the peer summaries or the system summaries and the corresponding model summaries against which the system summaries have to be evaluated. Below is an example of what the XML file looks like. The peer summaries are specified inside the <PEER> tag and the model summaries are specified inside the <MODEL> tag. The 'P ID' denotes the ID of the summarization system being evaluated and the 'M ID' denotes the human summarizer ID.

```
<ROUGE_EVAL version="1.0">
    <EVAL ID="1">
        <MODEL-ROOT>
            Models
        </MODEL-ROOT>
        <PEER-ROOT>
            Peers
        </PEER-ROOT>
        <INPUT-FORMAT TYPE="SPL">
        </INPUT-FORMAT>
        <PEERS>
            <P ID="1">d062j/WSJ891019-0021_BaryCsummary.txt</P>
        </PEERS>
        <MODELS>
            <M ID="A">d062ja/WSJ891019-0021-A</M>
            <M ID="G">d062jg/WSJ891019-0021-G</M>
        </MODELS>
    </EVAL>
</ROUGE_EVAL >
```

Figure 7.1 An Example Input XML file for ROUGE

A program was written to create the XML file automatically. This XML file was then used as input to the ROUGE system. When running ROUGE, several options can be chosen some of which specify preprocessing tasks and some specify which of the 5 ROUGE metrics have to be calculated. Some of the options that were set when we ran the evaluation task are –

- **-c** : In ROUGE, a single value of a ROUGE measure can be computed for all the summaries created by the summarization system. The aim is to calculate a value which is the representative for all values generated for all summaries [16]. All the values for a specific measure are used to calculate an interval within which an evaluation value for a summary is 95% likely to fall. This percentage value denotes the confidence. The mean of the confidence interval is the ROUGE value returned.
- **-r** : Generalization of the evaluation values is done by using bootstrapping by resampling method. 'r' denotes the number of samples to be used during this process.

The set of ROUGE scores generated for the set of summaries being evaluated specifies one sample of the data. To increase the confidence level we spoke about before, we need to have many such samples available to us. Bootstrapping is the process of creating pseudo-samples out the single available sample so we have a large number of datasets over which the scores can be calculated.

- **-n** : specifies the ‘n’ value for N-gram calculation. We used an ‘n’ value of 2 meaning ROUGE-1 and ROUGE-2 scores were calculated.
- **-m** : denotes the stemming task which is a preprocessing operation. The system summary and the model summary are stemmed to reduce the words to their root forms before they are compared and evaluated.

When running ROUGE we set the confidence value to 95%, use 1000 samples and set the ‘n’ value equal to 2.

7.4 Results

As mentioned previously, studies have shown that the ROUGE-1 metric correlates best with human summaries [14]. Hence, the same metric has been used to evaluate the system.

The ROUGE-N measure as we saw is a recall oriented measure since the denominator is the total number of N-gram matches [15]. A recall-related score in summarization refers to the extent to which a summary retains the content of the original document. Precision is a measure of how well a system summary matches the reference summary. The F-measure is the harmonic mean of the previous two measures. The ROUGE-1 or unigram scores that are calculated refer to single term matches found between the document and the summary.

Shown below are the values obtained for the ROUGE-1 Recall, precision and F-measure scores for each of the 5 modules in the system.

ROUGE-1 measure Module	**Recall**	**Precision**	**F-measure**
Thematic graph Weight	0.45506	0.42758	0.43776
Degree Centrality	0.47969	0.43507	0.45433
Eigenvector Centrality	0.48092	**0.43537**	**0.45514**
Barycenter Centrality	**0.48166**	0.43411	0.45458
HITS	0.47930	0.43700	0.45501

Table 7.1 ROUGE-1 Scores

The graphs below show the ROUGE-1 Recall, Precision and F-measure score comparison for each of the five summarization modules. We observe that for ROUGE-1 Recall, the highest scores have been obtained for Eigenvector centrality and Barycenter centrality modules which are about 0.48.

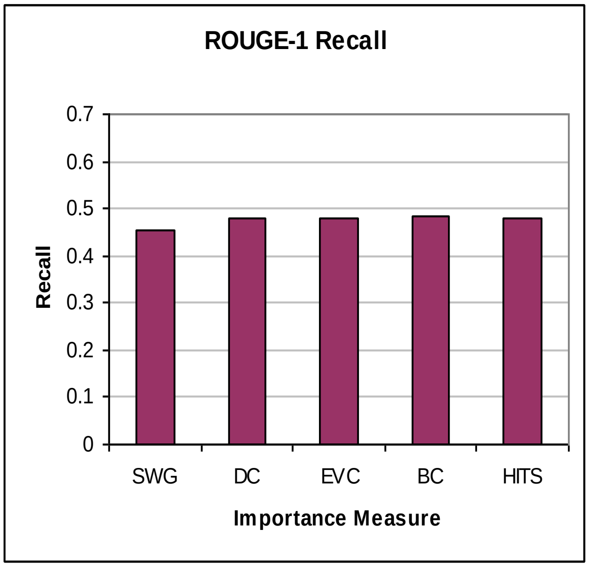


Figure 7.2 ROUGE-1 Recall Scores

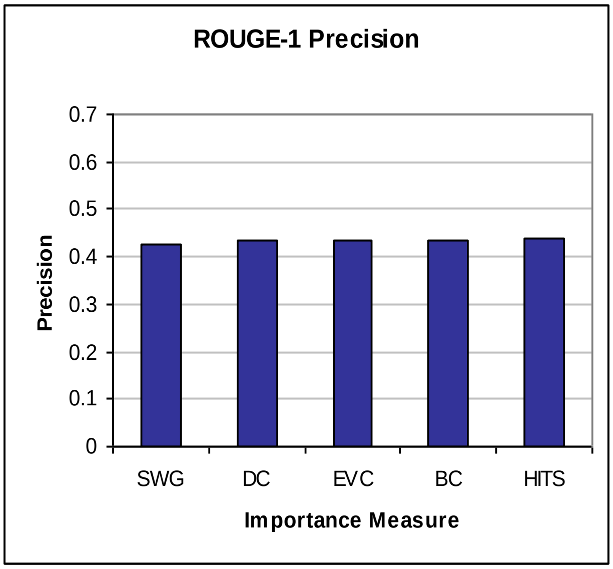
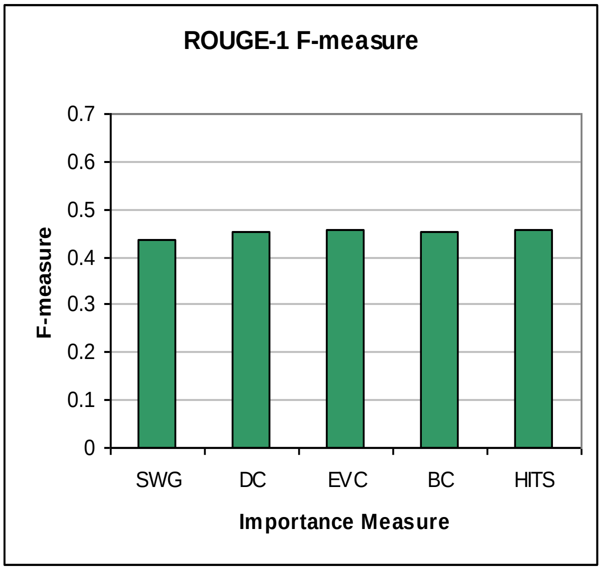


Figure 7.3 ROUGE-1 Precision Scores



Legend:

SWG	*Semantic Graph Weight*
DC	*Degree Centrality*
EVC	*Eigenvector Centrality*
BC	*Barycenter Centrality*
HITS	*Hubs and Authorities*

Figure 7.4 ROUGE-1 F-measure Scores

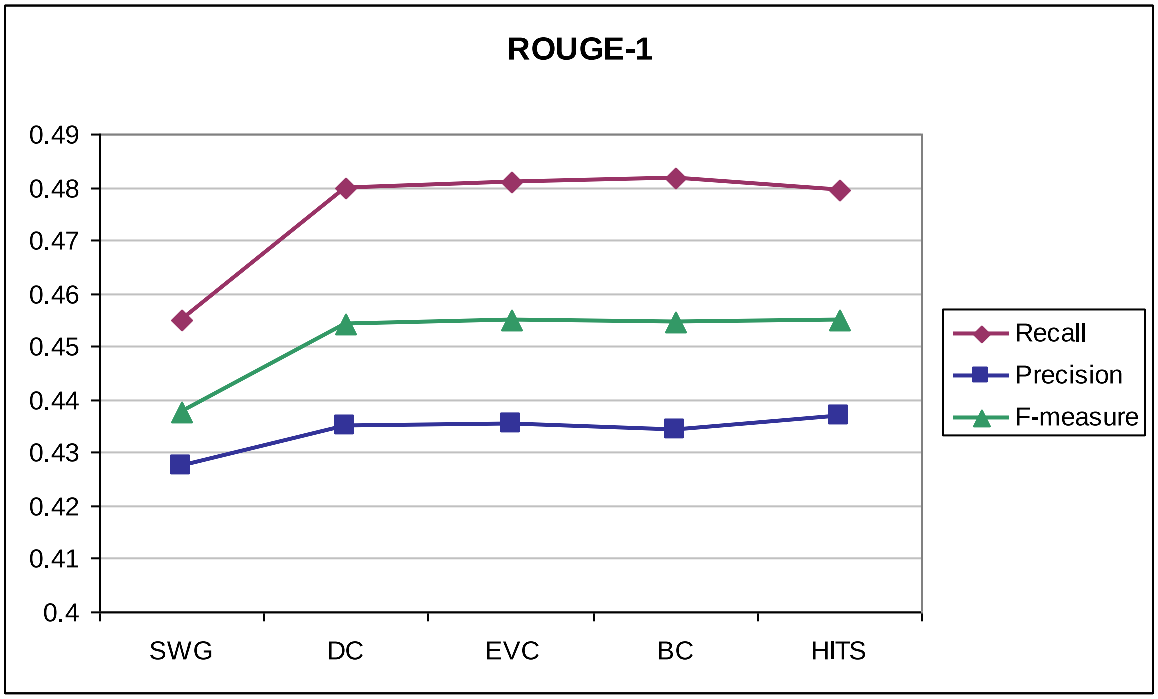


Figure 7.5 ROUGE-1 Scores

The figure above shows all the ROUGE-1 measures for all the 5 modules.

Comparison with other systems

Below is a table showing the ROUGE-1 scores for the top 5 systems that participated in the DUC 2002 single-document summarization task which required creation of 100-words summaries by all participating systems.

Top 5 systems for the DUC 2002 single document summarization task

System	S27	S31	S28	S21	S29	Baseline
ROUGE-1 score	0.5011	0.4914	0.4890	0.4869	0.4681	0.4799

We observe here that baseline summaries also score quite well since humans several times tend to prefer them because of their continuity and coherence.

Mihalcea’s TextRank system, as seen previously, is based on graph-based ranking method of summarization. It performed better than the above mentioned top 5 systems that took part in the DUC 2002 single-document summarization task scoring 0.5023 for the method that used the HITS ranking algorithm. The ontology-based system [10] has a ROUGE-1 F1 score equal to 0.4636 for 200 word summaries. The event-based summarization technique in [22] results in a best ROUGE-1 score of 0.28584 for 100-words summaries for the DUC 2001 document set.

Our system has a ROUGE-1 recall score of 0.48166 for the Barycenter centrality module which is the best performing module in the system. The Eigenvector module comes a close second with a score of 0.48092. In context of these other systems, we believe that this initial score is very encouraging and that with further improvements, as specified in the next section on future work, we will be able to achieve an even better score.

CHAPTER 8

CONCLUSION AND FUTURE WORK

The system we have built is a knowledge-based summarization system with the knowledge coming from an ontology. The knowledge is composed of not only in recognizing important phrases in the document, but also in recognizing the relationships and the relationship types that exist between them. This extracted knowledge is represented in the form of a thematic graph. Even without the summary, just looking at the nodes and relationships in the thematic graph gives us a rough idea about what the document is taking about. A summary however gives us the actual details. The presence of the knowledge in the form of a graph allows us to exploit several importance measures to help us decide how important a specific entity is to the document by allowing us to score each node. Hence, our idea basically lies in the coupling of these two steps. To our knowledge, this is the first system that uses ontological knowledge in this manner to obtain extractive summaries. Intuitively, this method makes a lot of sense. Improvements and further experimentation would most definitely make the existing system more reliable than it is now. We made use of an ontology formed out of Wikipedia since it is covers almost all topics that one can think of. Though from the perspective of coverage, Wikipedia is the best way to go, it has one short coming. Most of the relationships in Wikipedia ontology are anonymous relationships. Hence, there is not much meaning associated with an edge in the ontology. If we have information about the domain of the article that has to be summarized, specific domain ontologies can be chosen to form the thematic graph. In this way we could exploit the meanings of relationships between the entities in a better manner eventually leading to the formation of summaries of better quality. Hence, there is scope for improving the system in order to produce summaries of a higher caliber.

Future Work

We plan to drive our future work in summarization in the following major directions which we believe will help improve the summaries obtained both in terms of quality and content.

Anaphora resolution – As discussed earlier, anaphora resolution improves the quality of an extractive summary. We plan to incorporate anaphora resolution into out system to remove any ambiguities occurring in the summaries.

Combining with Traditional methods – We plan to integrate our knowledge-based method with statistical methods and find out if it improves the results obtained. Traditional methods, though mostly statistical in nature, do provide some intuition in discovering important terms in the document.

Usage of a counter-measure – The summarization system give more importance to central entities when deciding sentence importance. There may be situations where a node might be assigned a low score but it might be quite important in the document context though it is weakly associated with the main document theme. Such entities are seemingly of low importance but may specify some important detail which when added to the summary would improve its quality.

We plan to develop a technique to identify such entities and apply a counter-measure which brings up its score.

Extensive human evaluation – We plan on carrying out an extensive human evaluation of the system. Automatic evaluation of summarization systems, though work quite well, come second to human evaluation. Hence, conducting experiments with human subjects evaluating the system with the help of several volunteers will lead to a better understanding of what a human being is looking for in a summary.

We also plan to dive into the area of *Multi-document summarization* and extend our system to accommodate the same.

Our current system forms a thematic graph representing the document which has to be summarized. For a multi-document summarization system, we plan to obtain all the individual thematic graphs which represent individual documents in the set and combine them to form a single semantic graph which represents the entire set of documents. Merging the thematic graphs would be a simple procedure involving the following steps –

1) Create a graph using JUNG toolkit that we are currently using. For each unique node tag identified in the set of XML documents representing thematic graphs, a node is added to the graph.
2) For each unique edge tag identified, an edge is added to the graph.

Once the thematic graph representing the document set has been formed, the same procedure used for single-document summarization is used to rank the sentences based on one of the importance measures mentioned previously. The difference comes in the choosing of the sentences. Every time a sentence is picked to be included in the summary, it is compared with each sentence that is already part of the summary. Similarity measures are calculated based on the cosine between the vectors representing the two sentences. If the similarity value is above a particular threshold, the sentence is not included in the summary.

DUC discontinued the single-document summarization task after 2002. Only multi-document summarization systems have been participating in the DUC conference from 2003 onwards. We plan to be one of the participating systems in the DUC multi-document summarization task once our system is ready.

REFERENCES

[1] U Hahn, I Mani. The Challenges of Automatic Summarization, 2000

[2] Maciej Janik, Krys Kochut. Training-less Ontology-based Text Categorization. 2008

[3] Klaus Zechner. A Literature Survey on Information Extraction and Text Summarization, 1997

[4] Thomas Gruber. A translation approach to portable ontology specifications In: Knowledge Acquisition. 5: 199-199, 1993

[5] Rakesh Verma, Ping Chen, Wei Lu. A Semantic Free-text Summarization System Using Ontology Knowledge, 2008

[6] Chin-Yew-Lin. Looking for a few good metrics: ROUGE and its evaluation, 2004

[7] Ani Nenkova. Automatic Text Summarization of Newswire: Lessons Learned from the Document Understanding Conference, 2005

[8] Maciej Janik, Krys J. Kochut. Wikipedia in action: Ontological Knowledge in Text Categorization

[9] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Soren Auer, Christian Becker, Richard Cyganiak, Sebastian Hellmann. DBpedia - A Crystallization Point for the Web of Data

[10] Leonhard Hennig, Winfried Umbrath , Robert Wetzker. An Ontology-based Approach to Text Summarization, 2008

[11] Dragomir R. Radev, Hongyan Jing, Malgorzata Budzikowska. Centroid-based summarization of multiple documents, 2000

[12] Gunes Erkan, Dragomir R. Radev. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization, 2004

[13] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment, Journal of the ACM, 1999

[14] Chin-Yew Lin and Eduard Hovy. Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics, 2003

[15] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries, 2003

[16] M.S. Binwahlan, N. Salim and L. Suanmali. Intelligent Model for Automatic Text Summarization, 2009

[17] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Text, 2004

[18] Rada Mihalcea. Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization, 2004

[19] Rada Mihalcea and Paul Tarau. A Language Independent Algorithm for Single and Multiple Document Summarization, 2005

[20] Gabriel Murray, Steve Renals, Jean Carletta. Extractive Summarization of Meeting Recordings, 2005

[21] Jurij Leskovec, Natasa Milic-Frayling, Marko Grobelnik. Extracting Summary Sentences Based on the Document Semantic Graph, 2005

[22] Wenjie Li, Mingli Wu and Qin Lu (The Hong Kong Polytechnic University) and Wei Xu and Chunfa Yuan (Tsinghua University). Extractive Summarization using Inter- and Intra- Event Relevance, 2006

[23] Sentence and Paragraph Breaker - http://text0.mib.man.ac.uk:8080/scottpiao/sent_detector, Scott Piao

[24] Sentence Boundary Detector from Lingpipe - http://alias-i.com/lingpipe/

[25] The Java Universal Network/Graph Framework or JUNG - http://jung.sourceforge.net/index.html

[26] Wikipedia Statistics - http://stats.wikimedia.org/EN/TablesArticlesTotal.htm

[27] HITS algorithm - http://en.wikipedia.org/wiki/HITS_algorithm

[28] Document Understanding Conference - http://www-nlpir.nist.gov/projects/duc/intro.html

[29] ROUGE, Automatic summary evaluation - http://berouge.com/default.aspx

[30] Ontology (computer science) – http://www.wikipedia.org

[31] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1–7), 1998

[32] P.J. Herings, G. van der Laan, and D. Talman. Measuring the power of nodes in digraphs. Technical report, Tinbergen Institute, 2001

APPENDIX A

SUMMARIZATION EXAMPLE

Input text

Actor Gale Gordon, a sidekick of Lucille Ball from radio days who appeared with her in three television series, recalled on Wednesday how the comedian never thought of herself as particularly funny.

``Oh, but when she told a story or recalled an event that happened she always illustrated it with her body and her face,'' the 83-year-old Gordon said in a telephone interview. ``She never saw it, but she was extremely amusing.''

Gordon, who is appearing in a play in Edmonton, spent the day in an Edmonton hotel room, mourning the death of Lucille Ball in a Los Angeles hospital Wednesday at age 77 and quietly remembering his funny friend.

``The world will miss her greatly,'' the actor said. ``I personally will remember her as the best talent of her kind in the world ... a wonderful, warm friend and a bit of a genius _ the only one I've ever really known.''

In his first television series alongside Miss Ball, Gordon was the flustered banker Mr. Mooney in ``The Lucy Show'' from 1962-68. The next six years saw him as Harrison Carter on ``Here's Lucy.''

They were reunited briefly in 1986 for the short-lived ``Life With Lucy.''

Gordon first met Miss Ball in 1947 on the radio program ``My Favorite Husband,'' which later became ``I Love Lucy'' on television.

``I thought she was a very good-looking young woman, and very talented,'' he recalled. ``She was very pleasant to work with. None of us knew what was going to happen _ television was very young then and it wasn't taken too seriously. We never dreamed it would ever become the power it was ... and the great stars it would produce.''

Gordon last spoke to Miss Ball from Edmonton on his birthday, Feb. 20, when ``she called to wish me a happy birthday as she has done for years and years.''

Gordon has done regular stints for years at a chain of Edmonton-based dinner theatres called Stage West. He is currently appearing with that company's Mayfair Dinner Theatre as a priest in the play ``Mass Appeal.''

The play has another month to run and Gordon says that means he won't be able to attend Miss Ball's funeral.

``I couldn't do that because it would put people out of work,'' he said. ``She would be the first one to scream her head off if I did that.''

Semantic Graph Weight Module Summary

Actor Gale Gordon, a sidekick of Lucille Ball from radio days who appeared with her in three television series, recalled on Wednesday how the comedian never thought of herself as particularly funny. Gordon, who is appearing in a play in Edmonton, spent the day in an Edmonton hotel room, mourning the death of Lucille Ball in a Los Angeles hospital Wednesday at age 77 and quietly remembering his funny friend. Gordon first met Miss Ball in 1947 on the radio program "My Favorite Husband," which later became "I Love Lucy" on television. Gordon last spoke to Miss Ball from Edmonton on his birthday, Feb. 20, when "she called to wish me a happy birthday as she has done for years and years."

Degree Centrality Module Summary

Actor Gale Gordon, a sidekick of Lucille Ball from radio days who appeared with her in three television series, recalled on Wednesday how the comedian never thought of herself as particularly funny. Gordon, who is appearing in a play in Edmonton, spent the day in an Edmonton hotel room, mourning the death of Lucille Ball in a Los Angeles hospital Wednesday at age 77 and quietly remembering his funny friend. In his first television series alongside Miss Ball, Gordon was the flustered banker Mr. Mooney in "The Lucy Show" from 1962-68. Gordon first met Miss Ball in 1947 on the radio program "My Favorite Husband," which later became "I Love Lucy" on television.

Eigenvector Centrality Module Summary

Actor Gale Gordon, a sidekick of Lucille Ball from radio days who appeared with her in three television series, recalled on Wednesday how the comedian never thought of herself as particularly funny. Gordon, who is appearing in a play in Edmonton, spent the day in an Edmonton hotel room, mourning the death of Lucille Ball in a Los Angeles hospital Wednesday at age 77 and quietly remembering his funny friend. In his first television series alongside Miss Ball, Gordon was the flustered banker Mr. Mooney in "The Lucy Show" from 1962-68. Gordon first met Miss Ball in 1947 on the radio program "My Favorite Husband," which later became "I Love Lucy" on television.

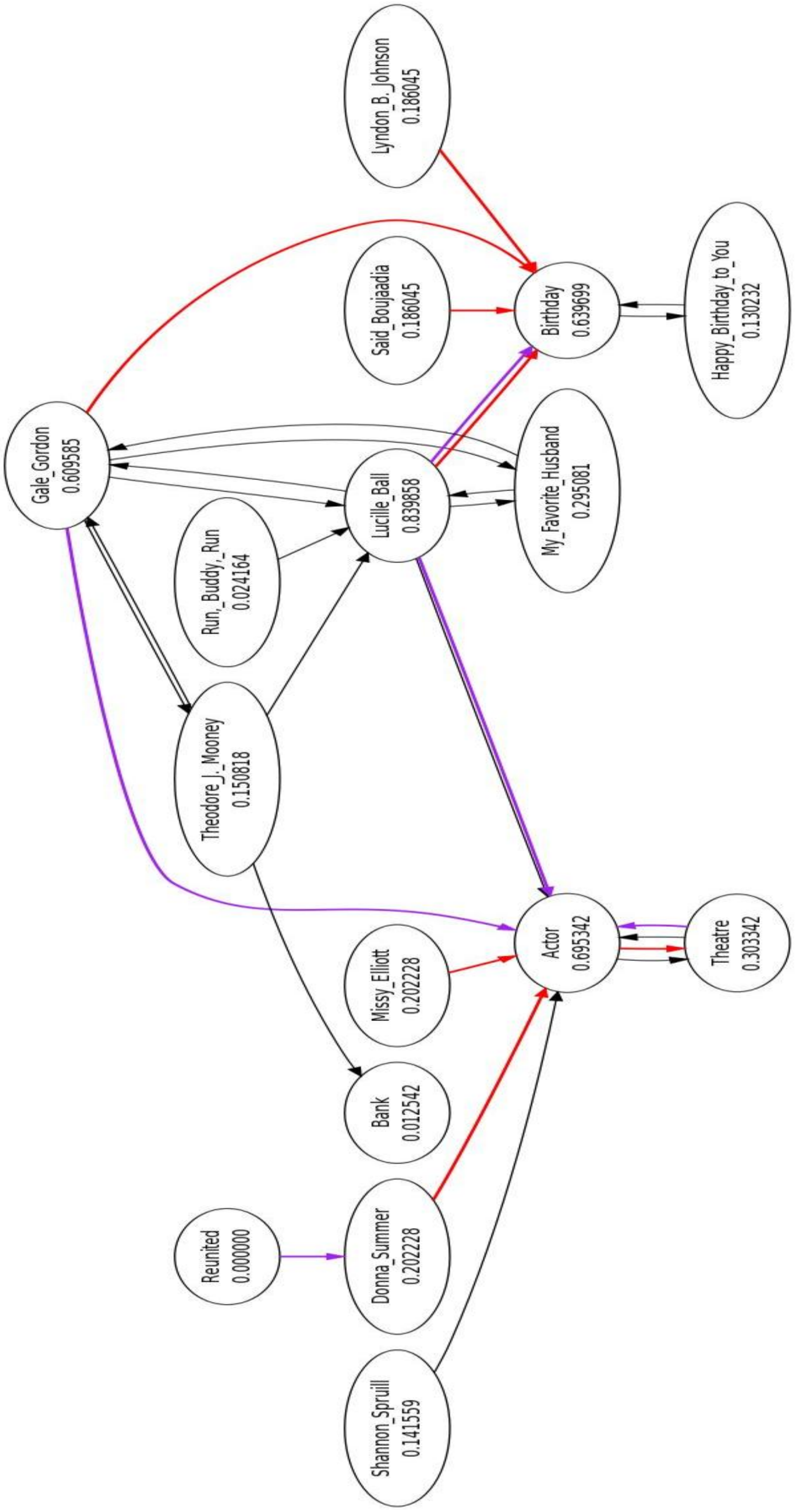
Barycenter Centrality Module Summary

Actor Gale Gordon, a sidekick of Lucille Ball from radio days who appeared with her in three television series, recalled on Wednesday how the comedian never thought of herself as particularly funny. In his first television series alongside Miss Ball, Gordon was the flustered banker Mr. Mooney in "The Lucy Show" from 1962-68. They were reunited briefly in 1986 for the short-lived "Life With Lucy." Gordon first met Miss Ball in 1947 on the radio program "My Favorite Husband," which later became "I Love Lucy" on television. Gordon has done regular stints for years at a chain of Edmonton-based dinner theatres called Stage West.

HITS Module Summary

Actor Gale Gordon, a sidekick of Lucille Ball from radio days who appeared with her in three television series, recalled on Wednesday how the comedian never thought of herself as particularly funny. Gordon, who is appearing in a play in Edmonton, spent the day in an Edmonton hotel room, mourning the death of Lucille Ball in a Los Angeles hospital Wednesday at age 77 and quietly remembering his funny friend. In his first television series alongside Miss Ball, Gordon was the flustered banker Mr. Mooney in "The Lucy Show" from 1962-68. Gordon first met Miss Ball in 1947 on the radio program "My Favorite Husband," which later became "I Love Lucy" on television.

These summaries were formed after turning the original text of the document into the following semantic graph.


Lyndon_B._Johnson
0.186045
Said_Boujaadia
0.186045
Birthday
0.639699
Happy_Birthday_to_You
0.130232
Gale_Gordon
0.609585
Lucille_Ball
0.839858
My_Favorite_Husband
0.295081
Run,_Buddy,_Run
0.024164
Theodore_J._Mooney
0.150818
Actor
0.695342
Theatre
0.303342
Missy_Elliott
0.202228
Bank
0.012542
Reunited
0.000000
Donna_Summer
0.202228
Shannon_Spruill
0.141559