2004

Automatic text summarization using lexical chains : algorithms and experiments

Kolla, Maheedhar

Lethbridge, Alta. : University of Lethbridge, Faculty of Arts and Science, 2004

AUTOMATIC TEXT SUMMARIZATION USING LEXICAL CHAINS:

ALGORITHMS AND EXPERIMENTS

Maheedhar Kolla

B.Tech, Jawaharlal Nehru Technological University, 2002

A Thesis

Submitted to the School of Graduate Studies

of the University of Lethbridge

in Partial Fulfillment of the

Requirements for the Degree

MASTER OF SCIENCE

Department of Mathematics and Computer Science

University of Lethbridge

LETHBRIDGE, ALBERTA, CANADA

Abstract

Summarization is a complex task that requires understanding of the document content to determine the importance of the text. Lexical cohesion is a method to identify connected portions of the text based on the relations between the words in the text. Lexical cohesive relations can be represented using lexical chains. Lexical chains are sequences of semantically related words spread over the entire text. Lexical chains are used in variety of Natural Language Processing (NLP) and Information Retrieval (IR) applications. In current thesis, we propose a lexical chaining method that includes the glossary relations in the chaining process. These relations enable us to identify topically related concepts, for instance *dormitory* and *student*, and thereby enhances the identification of cohesive ties in the text.

We then present methods that use the lexical chains to generate summaries by extracting sentences from the document(s). Headlines are generated by filtering the portions of the sentences extracted, which do not contribute towards the meaning of the sentence. Headlines generated can be used in real world application to skim through the document collections in a digital library.

Multi-document summarization is gaining demand with the explosive growth of online news sources. It requires identification of the several themes present in the collection to attain good compression and avoid redundancy. In this thesis, we propose methods to group the portions of the texts of a document collection into meaningful clusters. Clustering enable us to extract the various themes of the document collection. Sentences from clusters can then be extracted to generate a summary for the multi-document collection. Clusters can also be used to generate

summaries with respect to a given query.

We designed a system to compute lexical chains for the given text and use them to extract the salient portions of the document. Some specific tasks considered are: headline generation, multi-document summarization, and query-based summarization. Our experimental evaluation shows that efficient summaries can be extracted for the above tasks.

Acknowledgments

There are many people, without whose support and contributions this thesis would not be possible.

First, I want to thank my supervisor Dr. Yllias Chali for his constant feedback and ideas during the entire course. Also I would like to thank Dr. Shahadat Hossain and Dr. Daya Gaur for their suggestions and constant help to keep me focussed. I would like to thank Dr. Ian Witten and his research group, for their help with the MG software. I would also like to acknowledge my family, friends and colleauges for their support and encouragement all through my course.

Contents

v

Bibliography

List of Figures

List of Tables

Chapter 1

Introduction

Popularity of the internet has contributed towards the explosive growth of online information. Search engines provide a means to access huge volumes of information by retrieving the documents considered relevant to the user's query. Even with search engines, the user has to go through the entire document content to judge its relevance. This contributes towards a well recognized information overload problem.

Similar information overload problems are also faced by corporate networks, which have information spread across various kinds of sources - documents, web pages, mails, faxes, manuals etc. It has become a necessity to have tools that can digest the information present across various sources and provide the user with condensed form of the most relevant information. Summarization is one such technology that can satisfy these needs.

Summaries are frequently used in our daily life to serve variety of purposes. Headlines of news articles, market reports, movie previews, abstracts of journal articles, TV listings, are some of the commonly used forms of summaries. Oracle's Text uses the summarization technology to mine textual databases. InXight summarizer [1] provides summaries for the documents retrieved by the information retrieval engine. Microsoft's Word provides the AutoSummarize option to highlight the main

---

[1]http://www.inxight.com/products/sdks/sum/

1

concepts of the given document. BT's ProSum, IBM's Intelligent Miner [2] are some of the other tools providing summaries to speed the process of information access.

Several advanced tools have been developed in recent times using summarization techniques to meet certain requirements. Newsblaster (McKeown et al., 2003), and NewsInEssence (Radev et al., 2001) allow the users to be updated about the interesting events happening around the world, without the need to spend time searching for the related news articles. They group the articles from various news sources into event related clusters, and generate a summary for each cluster. Meeting summarizer (Waibel et al., 1998) combines the speech recognition and summarization techniques to browse the contents of the meeting. Persival (McKeown, Jordan, and Hatzivassiloglou, 1998), and Healthdoc (Hirst et al., 1997), aid physicians by providing a "recommended treatment", for particular patient's symptoms, from the vast online medical literature. Broadcast news navigator (Maybury and Merlino, 1997) is capable of understanding the news broadcast and present the user with the condensed version of the news. IBM's Re-Mail (Rohall et al., 2004) and (Rambow et al., 2004) can summarize the threads of e-mail messages based on simple sentence extraction techniques.

Summarization can be defined in several ways: According to Mani and Maybury (1999), "summarization is the process of distilling the most important information from the source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)". According to Mani (2001), "goal of summarization system is to take an information source, extract content from it and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's need". In brief (Sparck-Jones, 1999), "given the *input* source, summarization is the process of generating *output* to satisfy specific *purpose*".

Input to the summarization process can be in different formats like text, video, audio, image. We concentrate only on the textual format of the input. Summaries generated are dependent on various factors (Mani, 2001) (Sparck-Jones, 1999) -

---

[2]http://www-306.ibm.com/software/data/iminer/

e.g. different summaries can be generated for the same input source depending on their functionality and usage. The most important factor in summarization is the *compression rate*. It can be defined as the ratio of the summary length to the source length.

Summaries generated can contain information from a single document (*single document summaries*) or a collection of documents (*multi-document summaries*). Multi-document summarization involves identification of the various concepts spread across the collection in order to obtain more compression and reduce redundancy. Summaries can serve variety of functions; they can be "indicative", highlighting the salient content of the document without much of an explanation. They can also be "informative", explaining certain concept to the maximum possible detail at the given compression rate. Summaries can also be "evaluative", rating the work of the author (book reviews etc). Summaries can be generated by just copying and pasting the text from the source (*extracts*), or can be generated in abstractor's own words (*abstracts*).

Another distinction between summaries can be made based on the intended audience. Generic summaries are intended to be read by broader section of people and contain the information considered salient in the author's viewpoint. User-focused summaries are generated to be read by a specific group of people having interests in a specific topic or concepts. These summaries include information relevant to the user's interests irrespective of its salience in the document. Summaries can be *fragments* of sentences providing the gist of the document (useful for indexing); or can be highly polished *fluent* text that can be used as substitute for the actual documents, like abstracts of journal articles.

The process of summarization, as shown in Figure 1.1, can be sub-divided into three stages (Mani and Maybury, 1999) (Mani, 2001) (Sparck-Jones, 1999):

- Analysis: This phase builds an internal representation of the source.

- Transformation: This phase generates a representation of the summary based

Documents

Compression 0   30   100

Generic	Indicative	Fragments
User-focused	Informative	Connected text
	Evaluative	

Analysis   Transformation   Synthesis
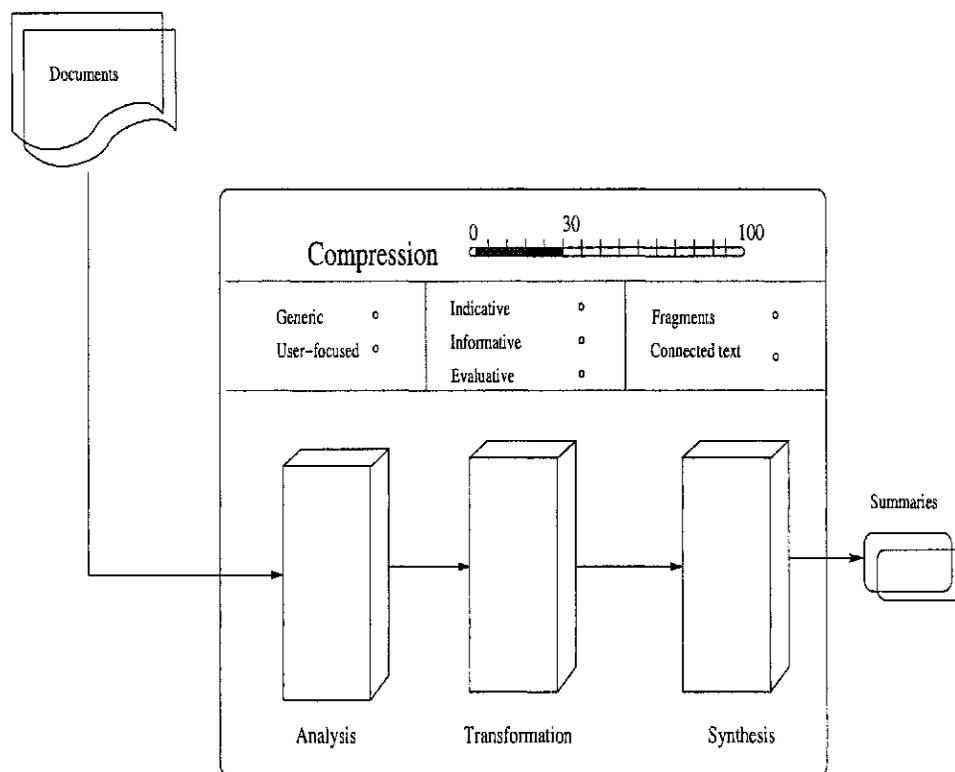
Summaries

Figure 1.1: Basic architecture of an automatic text summarization system

on the internal representation of the source.

- Synthesis: This phase interprets summary representation back into the natural language form.

Only methods involving multi-document summarization or abstract generation go through the *transformation* phase. Methods to generate extracts for single document directly go to the synthesis phase after the analysis phase. Each phase undergoes one or more of the following basic condensation operations (Mani and Maybury, 1999) (Mani, 2001):

- Selection : To filter unimportant and redundant information.

- Aggregation : To group information from various portions of the document.

- Generalization : To substitute a concept with more general or abstract one.

These basic condensation operations can be applied during various phases of summarization on elements such as words, phrases, clauses, sentences, or discourse. Elements in these condensation operations can be analyzed at various linguistic levels: *morphological, syntactic, semantic and discourse/pragmatic*. Based on the level of linguistic analysis of the source, summarization methods can be broadly classified into two approaches (Mani, 2001):

1. *Shallow approaches*: These methods tend to identify the salient portions of the text based on the surface level analysis of the document. These methods extract the sentences, considered salient, and then re-arrange them to form a coherent summary. Since these methods extract the complete sentence(s), they cannot achieve greater compression rates compared to the deeper approaches.

2. *Deeper approaches* : These methods perform deeper semantic analysis of the document content to identify the salient portions. They require highly domain-specific information to be able to perform deeper analysis. Lack of such widely available knowledge bases factors makes these methods hard to implement. One major advantage of these methods is the level of compression obtained.

Earlier shallow approaches were mainly superficial nature. They considered features such as word count (Luhn, 1958), presence of certain cue phrases (Edmundson, 1969), position of the sentence (Edmundson, 1969) (Lin and Hovy, 1997) to determine the important concepts of the document and saliency of the information. These features fail to capture the "aboutness" or "theme" of the content.

Concepts of coherence and cohesion enable us to capture the theme of the text. Coherence represents the overall structure of a multi-sentence text in terms of macro-level relations between clauses or sentences (Halliday, 1978). Cohesion,

as defined by Halliday and Hasan (1976), is the property of holding the text together as one single grammatical unit based on relations between various elements of the text. Cohesive relations can be classified into five categories: *ellipsis, conjunction, substitution, reference* and *lexical cohesion.*

Lexical cohesion is defined as the cohesion that arises from the semantic relations between the words in the text (Morris and Hirst, 1991). Lexical cohesion provides a good indicator for the discourse structure of the text, used by professional abstractors to skim through the document. Lexical cohesion does not occur just between two words but a sequence of related words spanning the entire text, *lexical chains* (Morris and Hirst, 1991).

Lexical chains are used in a variety of NLP and IR applications such as summarization (Barzilay and Elhadad, 1997) (Silber and McCoy, 2002), detection of malapropism (Hirst and St-Onge, 1997), indexing document for information retrieval (Stairmand, 1996), dividing the text into smaller segments based on the topic shift (Kan et al., 1998) (Hearst, 1997), automatic hypertext construction (Green, 1999).

Several methods have been proposed to compute lexical chains (Barzilay and Elhadad, 1997) (Silber and McCoy, 2002) (Hirst and St-Onge, 1997) (Stairmand, 1996) (Galley and McKeown, 2003) (Stokes, 2004). Almost all of the methods use WordNet (Miller et al., 1993) to identify the semantic relations. In current work, we investigate various methods to compute lexical chains and then propose method to compute lexical chains by including topical relations, not directly obtained using WordNet relations. These relations are identified using the eXtended WordNet (Mihalcea and Moldovan, 2001).

The goal of the thesis is text summarization using the lexical chains. We compute lexical chains and extract sentences based on the spread of the lexical chains to satisfy user's criteria. More specifically, we propose methods to perform the following tasks: *headline generation, Multi-document summarization,* and *query based summarization.*

Lexical chains computed are used to extract sentences to generate a cohesive summary for the document. Headlines can be generated by compressing the most relevant sentences extracted from the document. These compression techniques are motivated by certain linguistic principles and thus can be used in various domains. Multi-document summarization requires identification of various themes present in the collection to avoid redundancy. In this thesis, we propose methods to cluster the segments of the document collection based on the similarity of theme, determined by using lexical chains. We then extract the sentences from each cluster to generate a multi-document summary.

We evaluate the summaries generated by our system in comparison with human generated "ideal" summaries. We compare our system-generated summaries with the summaries generated by other methods and find that our system performs better than most of the systems. We also compare the quality of the summaries generated and find that our summaries are of better quality than most of the systems. These comparisons were made in the context of an evaluation workshop organized by National Institute of Standards and Technology (NIST) (Over, 2004).

The thesis is organized as follows: chapter 2 provides the background information on the concept of lexical cohesion. We also discuss about the lexical resources used to identify the semantic relations. We then explain various methods to compute lexical chains. In chapter 3, we detail the methods to generate the summary from the given document(s) based on the user's criteria. We explain the method to generate the summaries for three specific tasks: headline generation, multi-document summarization and query-based summarization. Chapter 4 explains the tools used and methods followed to evaluate our summarization techniques. Finally, we draw some conclusions and examine possible future work.

Chapter 2

Lexical Chains

Human abstractors construct a structured mental representation (*theme*) of the document and synthesize the document based on the theme to generate the summary. Primitive computational methods used word count measure (Luhn, 1958) to determine the theme of the document. Motivation behind this approach was that frequent words contain the core information of the document. One major drawback of this approach is that it does not consider the importance of a word in the given context. Lack of such consideration fails to capture the "aboutness" or the "theme" of the document. For example (Barzilay and Elhadad, 1997):

(1) *"Dr. Kenny has invented an anesthetic machine. This device controls the rate at which an anesthetic is pumped into the blood".*

(2) *"Dr. Kenny has invented an anesthetic machine. The doctor spent two years on this research.".*

Both texts have the same frequency of the words "Dr. Kenny" and "machine", but the first text is about the machine whereas the second one is about Dr. Kenny. This distinction can only be made by considering the relation between the words in the text (e.g. machine and device in first text).

Cohesion, as defined by Halliday and Hasan (1976), enables us to capture the "theme" of the document. It can be defined as the property of the text to hang

8

together as one large grammatical unit, based on relations between words. For example,

(3) Wash and core six cooking *apples*.

(4) Put *them* into fireproof dish.

In the above set of sentences, *them* in the second sentence refers to the *apples* in the first one. This property of cohesiveness is not visible between un-related sentences. For example,

(5) Wash and core six cooking apples.

(6) Toronto is the biggest city in Canada.

Cohesion relations influence the comprehensibility of the text (Mani, 2001). Cohesive structure can be represented as graphs with elements of the text as the nodes and relations between the elements as edges connecting the nodes. Saliency of the information can then be determined based on the connectivity of the nodes in the graph.

Halliday and Hasan (1976) divided cohesive relations into the following categories:

Reference: reference relations, in general, involve the usage of pronouns to refer to an entity mentioned in the preceding or the following text. In the following example, *he* and *John* both refer to the same person "John".

(7) John went to Australia. He had to attend a conference.

Substitution: relations in which one particular phrase or word is replaced with an article such as one or several etc. In the following example, *several* is used to replace the word car.

(8) I bought a new car today. There were several I could have had.

Ellipsis: relations established by elimination of certain phrases or words. In the following example, the word "distant" is not mentioned for the second time.

> (9) New York is as distant from San Francisco as Boston is *[distant]* from London.

Conjunction: relations achieved by using connectors to show the relationships between statements.

> (10) He gave me directions *but* I lost it.

> (11) *When* you have finished, we shall leave.

2.1   Lexical Cohesion

Lexical Cohesion: Lexical cohesion is the device to hold the text together based on the semantic or identical relations between the words of the text (Morris and Hirst, 1991). For example :-

> *Mars* is a truly intermediate "environment" between the two bodies, being about half the size of the *Earth* and twice the size of the *Moon*. Size is a very important factor in determining a *planet's* "environment", not only because of gravity but because of "atmosphere" and internal heat.

In the above example, the text can be identified as cohesive based on the relationship between words such as { *environment, atmosphere, environment* } and { *Mars, Earth, Moon, planet* }.

Lexical cohesion relations can be broadly divided into the following categories:

1. **Reiteration category:** Reiteration includes relations such as repetition of the word, words having synonymy relation and also super-ordinate/subordinate relations.

- **Reiteration with/without identity of reference:** Relations between identical words or words referring to the same entities.

  (12) Kerry is riding a *horse*.

  (13) *The horse* is white in colour.

  In the above examples, both of these sentences refer to the same entity *horse*.

- **Reiteration by using synonyms:** Relations between words, which have the same meaning and could be used interchangeably. For example,

  (14) Microsoft filed seven *lawsuits* against defendants identified only as "John Does."

  (15) The *suits* are believed to be first under anti-spam rules established earlier this year.

- **Reiteration by means of super-ordinate:** This kind of relation occurs when reference is made to the superclass of the entity previously mentioned. For example,

  (16) Scientists have found a way of triggering a runaway greenhouse effect using *gases* more effective than *carbon dioxide*.

  In the above text, *carbon dioxide* is a sub-class of *gas* (singular for gases).

2. **Collocation category:** Collocation includes the relations between words that occur in similar lexical contexts. These relations are comparatively hard to identify than the reiteration relations.

   - **Systematic semantic relation:** In this relation, entities referred in two different sentences are the subsets of the same class. For example,

     (17) Scientists believe that they can turn *Mars* into a world with characteristics like *Earth*

In above example, *Earth* and *Mars* belong to the same class i.e. planets.

- **Non-systematic semantic relation:** This relation occurs among words used in similar context. For example (Morris and Hirst, 1991),

  (18) Mary spent three hours in the *garden* yesterday.

  (19) She was digging *potatoes*.

  In the above example, *garden* and *potatoes* are words normally used in similar lexical contexts.

2.1.1 Lexical cohesion and coherence

Coherence is a discourse property that describes the meaning of the text based on the macro-level relations, such as *elaboration, explanation, cause*, between sentences or clauses or phrases. For example,

  (20) Walk out the door of this building.

  (21) Turn left.

Mani (2001) identified the relation between the sentences as *occasion*, in which the first sentence details the change in location and that the state holds true even in the second sentence. While this relation could be easily identified as "occasion", it is difficult to identify the exact coherence relation in many cases (Morris and Hirst, 1991). Consider the example:

  (22) John can open the safe.

  (23) He knows the combination.

Hobbs (1978) identified the relation between the two sentences as "elaboration", but Morris and Hirst (1991) claim that the relation could also be "explanation". They proceeded to state that the precise identification of the coherence relation depends on the belief and context of the reader.

By identifying the relation between the words *safe* and *combination*, a cohesive relation could be established between the two sentences. Based on the intuition that cohesion is only possible when the document is coherent (with some exceptions), Morris and Hirst (1991) concluded that cohesion can be used to approximate the coherence of the text. Lexical cohesion doesn't occur just between two words, but over a sequence of semantically related words called *lexical chains* (Morris and Hirst, 1991). Lexical chains enable us to identify the lexical cohesive structure of the text, without need for complete understanding of the text. Semantic relations between the words can be identified by using lexical resource such as WordNet.

2.2 WordNet 2.0

WordNet is a machine readable dictionary built on the basis of psycholinguistic principles. It contains English nouns, verbs, adjectives, and adverbs organized on the basis of their word meanings, rather than word forms (Miller et al., 1990). Each "word form" in WordNet represents some underlying lexical "meaning". Some word forms can represent several meanings and some meanings can be represented by various forms. Table 2.1 illustrates the concept of lexical matrix, used to map word meanings to word forms. Entry '$E_{i,j}$' in the lexical matrix symbolizes that word form 'WF_j' refers to the meaning 'M_i'.

Word	Word Forms				
Meaning	WF_1	WF_2	WF_3	$\ldots$	WF_n
M_1	$E_{1,1}$	$E_{1,2}$			
M_2	$E_{2,1}$		$E_{2,3}$		
M_3		$E_{3,2}$			
M_j	$E_{j,1}$	$E_{j,2}$			$E_{j,n}$

Table 2.1: Mapping between word forms and lexical meanings

Word forms referring to the same underlying concept are said to be *synonymous* for instance (WF_1, WF_2). WordNet organizes the word forms belonging to

same syntactic category that refer to the common underlying concept into synonym sets called *synsets*. For example, synset {*standard, criterion, measure, touchstone*} refers to the lexical meaning "*a basis of comparison*". Word forms that refer to more than one underlying concept are called *polysemous* (WF_1). Table 2.2 illustrates number of synsets and number of polysemy words, average polysemy of WordNet 2.0.[1]

Category	Synsets	Word-Sense pairs	Polysemous words	Average polysemy
Noun	79689	141690	15124	1.23
Verb	13508	24632	5050	2.17
Adjective	18563	31015	5333	1.44
Adverb	3664	5808	768	1.24

Table 2.2: WordNet-2.0 statistics

WordNet connects the synsets by certain lexico-semantic relations (Table 2.3). The most dominating relation is *hypernym/hyponym* relation, in which one synset is a *whole class/member of class* of another synset. Hypernym/hyponym relation organizes nouns and verbs into 11 and 512 hierarchies. Underlying hierarchical organization of synsets can be seen in Figure 2.1. Generality of concepts increases while traversing upwards in the hierarchical structure. Figure 2.2 shows the Word-Net entry for the word *modification*.

Relation	Examples
Synonym	*weather - atmospheric condition*
Hypernym/Hyponym	*car - vehicle*
Antonym	*good - bad*
Meronym/Holonym	*steering - navigation.*

Table 2.3: Sample WordNet relations

---

[1] http://cogsci.princeton.edu/~wn
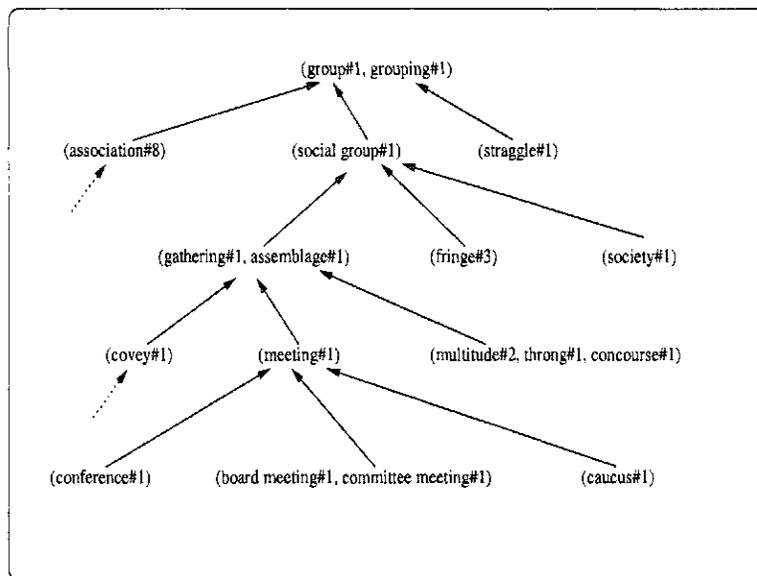
Figure 2.1: WordNet hierarchical structure

2.2.1 Gloss

Gloss of each synset consists of definition(s), comment(s) and some example(s) for the underlying lexical concept. For example, gloss of the synset {*weather, weather condition, atmospheric condition*} contains the definitions {*the meteorological conditions: temperature and wind and clouds and precipitation*}, followed by examples {"they were hoping for good weather"; "every day we have weather conditions and yesterday was no exception"}.

Gloss definitions can be used to identify the relations between two concepts not directly related using direct WordNet relations. For example, consider the words *dormitory* and *university*: there exists no direct WordNet relation between the two words although the relation can be identified by humans. Considering the gloss of the word *dormitory* "a college or university building containing living quarters for students", we can establish relation between the two words (we also obtain relation between *dormitory* and *students* from the same gloss definition).

Lesk (1986) used the presence of the gloss concepts of a word in the current

4 senses of modification

Sense 1
alteration, modification, adjustment
    => change
        => action
            => act, human action, human activity

Sense 2
modification
    => copy
        => representation
            => creation
                => artifact, artefact
                    => object, physical object
                        => entity
                    => whole, whole thing, unit
                        => object, physical object
                            => entity

Sense 3
modification, qualifying, limiting
    => grammatical relation
        => linguistic relation
            => relation
                => abstraction

Sense 4
change, alteration, modification
    => happening, occurrence, natural event
        => event

Figure 2.2: WordNet entry for the word *modification*

surroundings to narrow down the sense of the word being used in current context. Banerjee and Pedersen (2003) went further and measured the semantic relatedness between two concepts based on their gloss definition overlap. Harabagiu and Moldovan (1998) considered the gloss related concepts to infer the information not explicitly stated in the text.

2.3 eXtended WordNet

eXtended WordNet (Mihalcea and Moldovan, 2001) is a semantically enhanced tool based on the gloss definitions of synsets present in WordNet. It can be used in various applications such as question answering, text coherence, information retrieval.

Each synset's gloss in WordNet is processed to separate the definition from the examples and comments. Each definition is then processed to generate a parse tree representation and further processed to generate a logical transform. Each definition is also part of speech tagged. The open class words (nouns, verbs, adjectives and adverbs) are then transformed into their baseform (e.g: word 'senses' into 'sense'). Each open class word is then disambiguated to identify the sense of the word used in the definition, using both manual and automatic disambiguation methods. (See Figure 2.3 for the eXtended WordNet entry of the word "phenomenon").

The main goal of the eXtended WordNet is to extend the normal WordNet relations by including the topically related concepts. This would support *text inference*, problem of extracting relevant, unstated information from the text (Harabagiu and Moldovan, 1998), and thus provides means for better understanding of the "theme" of the text. For example (Moldovan and Novischi, 2002) :

(24) John was hungry.

(25) He opened the refrigerator.

Humans would consider this text coherent based on the "cause" relation, relating it to their daily activity that *hunger is the "cause" for John to open the refrigerator,*

```
<gloss pos="NOUN" synsetID="00029881">
  <synonymSet>phenomenon</synonymSet>
  <text>
    any state or process known through the senses rather than by intuition or reasoning
  </text>
  <wsd>
    <wf pos="DT" >any</wf>
    <wf pos="NN" lemma="state" quality="normal" wnsn="4" >state</wf>
    <wf pos="CC" >or</wf>
    <wf pos="NN" lemma="process" quality="silver" wnsn="2" >process</wf>
    <wf pos="VBN" lemma="know" quality="silver" wnsn="5" >known</wf>
    <wf pos="IN" >through</wf>
    <wf pos="DT" >the</wf>
    <wf pos="NNS" lemma="sense" quality="normal" wnsn="1" >senses</wf>
    <wf pos="RB" lemma="rather" quality="normal" wnsn="1" >rather</wf>
    <wf pos="IN" >than</wf>
    <wf pos="IN" >by</wf>
    <wf pos="NN" lemma="intuition" quality="silver" wnsn="1" >intuition</wf>
    <wf pos="CC" >or</wf>
    <wf pos="NN" lemma="reasoning" quality="silver" wnsn="1" >reasoning</wf>
  </wsd>
  <parse quality="NORMAL">
  (TOP (S (NP (NN phenomenon) )
      (VP (VBZ is)
        (NP (NP (DT any) (NN state) (CC or) (NN process) )
          (VP (VBN known)
            (PP (IN through)
              (NP (DT the) (NNS senses) ) )
            (PP (RB rather) (IN than)
              (PP (IN by)
                (NP (NN intuition) (CC or) (NN reasoning) ) ) ) ) ) )
      (. .) ) )
  </parse>
  <lft quality="NORMAL">
    phenomenon:NN(x1) -> any:JJ(x1) state:NN(x2) or:CC(x1, x2, x3) process:NN(x3) know:VB(e1, x8, x1)
    through:IN(e1, x4) sense:NN(x4) rather:RB(x4) than:IN(e1, x4) by:IN(x4, x7) intuition:NN(x5)
    or:CC(x7, x5, x6) reasoning:NN(x6).
  </lft>
</gloss>
```

Figure 2.3: eXtended WordNet entry for synset *phenomenon*

where food is kept for storage. Similar inference can be obtained using eXtended WordNet. Consider the gloss of the words *hungry* and *refrigerator:*

> *hungry: feeling a need or desire to eat food.*
>
> *refrigerator: a kitchen appliance in which food can be stored at low temperature.*

Moldovan and Novischi (2002) proposed that a chain can be created between hungry and refrigerator, which explains the implicit meaning that opening of refrigerator is mainly to eat food and thus identifying the cohesive property of the text.

2.4   MG

eXtended WordNet consists of "XML" format files for each syntactic category of WordNet. Each XML file consists of the gloss definitions processed as explained in Section 2.3. We considered only gloss relations between the nouns in a definition. In order to extract the *noun concepts* present in the gloss of a synset, we need to query the noun.xml file with the *synsetId.*

MG [2] is collection of programs, used to create and query full text inverted index of a document collection. MG creates an inverted index of all the words in the document using *mgbuild.* It is capable of indexing large volumes of data within shorter time. Once indexed, the document collection can be queried using *mgquery.* With the help of mgquery, we can perform complex queries including boolean operators such as "AND", "OR", and "NOT".

Given the amount of time taken to extract the information from the noun.xml file, using traditional XML query modules, we decided to index and query the noun.xml file using MG. We extract the gloss related concepts by querying the indexed files with the *synsetID.* Illustration of this procedure is shown in Figure 2.4.

---

[2]http://www.cs.mu.oz.au/mg/

```
Enter a command or query (.quit to terminate, .help for assistance).
>3107676

  <gloss pos="NOUN" synsetID="3107676">
  college#n#1
  university#n#1
  building#n#1
  living_quarters#n#1
  student#n#1
  </gloss>
```

Figure 2.4: Sample MG query

As illustrated in the figure, we obtain the nouns present in the gloss for the synsetID "03107676". Concepts extracted can then be used to compute lexical chains.

2.5 Lexical Chains

Lexical chains are sequence of semantically related words, spanning over the entire text (Morris and Hirst, 1991).

For example:

> *Ammonia* may have been found in *Mars' atmosphere* which some scientists say could indicate *life* on the *Red Planet*. The tentative detection of *ammonia* comes just a few months after *methane* was found in the *Martian atmosphere*. *Methane* is another *gas* with a possible *biological* origin.

- {*Mars, Red Planet, Martian*}

- {*Ammonia, Ammonia, Methane, Methane, gas*}

- {*life, biological*}

Lexical chains can be computed by the surface level analysis of the text and would help us to identify the theme of the text (e.g.: "life on mars" for the above text). Lexical chains are used in various NLP applications; indexing for information retrieval (Stairmand, 1997), to correct malapropism (Hirst and St-Onge, 1997), to divide the text into smaller segments (Hearst, 1997), automatic hypertext construction between two texts (Green, 1999).

Lexical chains are also useful in identifying the sense of the word being used in the current context (Morris and Hirst, 1991). For example, consider the word *"bank"* which can have two senses such as *"a financial institution"* or *"river side"*. Given the lexical chain *"{bank, slope, incline}"*, we can narrow down the sense of the word *"bank"* being used in this context to the *"river side"*. This process of identifying the sense of the word in the given context is called as "word sense disambiguation" (WSD). WSD is important to identify the topic or theme of the document and is helpful in various tasks: summarization, query processing, text similarity, etc.

2.5.1 Computation of lexical chains

Several methods have been proposed to perform both manual (Morris and Hirst, 1991) and automatic computation of lexical chains (Barzilay and Elhadad, 1997) (Silber and McCoy, 2002) (Hirst and St-Onge, 1997) (Galley and McKeown, 2003) (Stokes, 2004). In general, the process of lexical chaining consists of the following steps (Barzilay and Elhadad, 1997) :

1. Selection of the candidate words.

2. For each candidate word sense, find the compatible chain in which it is related to the chain members.

3. If found, insert the word and update the chains.

Lexical chaining requires identification of the semantic relations to determine the compatibility of the word with respect to the chain. Almost all of the methods to compute lexical chains use WordNet [3] to identify the semantic relations between the word senses.

Hirst and St-Onge's algorithm:

Hirst and St-Onge (1997) proposed the first algorithm to automatically compute lexical chains, using WordNet as lexical source. They classified the relations into three categories:

- Extra strong relations: relations involving repetition of the words (machine, machine).

- Strong relations: includes relations such as synonymy (machine, device), hypernym/hyponym (car, machine) , holonym/meronym, etc.

- Medium-strength relations: special relations based on some specific semantic relations (apple, carrot).

Only those words that contain noun entry in WordNet are used to compute lexical chains. Each candidate word sense is included in one lexical chain, in which it has relation with the last entered chain member. In case of multiple compatible chains, extra strong relations are preferred over the strong relations, both of which are preferred over the medium-strength relations. Once the word sense is inserted into a chain, all the non-compatible senses of the word are discarded. If no compatible chain is found then a new chain is created with all the senses of the word.

Barzilay and Elhadad's algorithm:

Barzilay and Elhadad (1997) proposed the first dynamic method to compute lexical chains. They considered all possible "interpretations" of the word and assign

---

[3]http://cogsci.princeton.edu/~wn

the best possible interpretation for the word based on its connectivity. Barzilay's method differs from the Hirst and St-Onge (1997) method in the following aspects:

- Selection of candidate words: both nouns and compound nouns are considered as the candidate words for the chain computation. Input text is part of speech tagged using Brill's tagger (Brill, 1992). This eliminates the wrong inclusion of the words such as *read*, which have both noun and verb entries in WordNet. Compound nouns are identified using the shallow based parse of the text.

- Segmentation of the text: using Hearst (1994) algorithm, they divided the text into smaller segments. This enhances the analysis of the document content for better understanding various topics in the text.

Barzilay and Elhadad computed all possible interpretations for all the words and then retained the best possible interpretation. They defined a component as a list of interpretations exclusive to each other. Word read from the text is inserted into the compatible components, in which it influences the selection of the senses for the other words. If no compatible component is found, a new component is created with all possible senses of the word.

Each interpretation score is equal to sum of all the chain scores. Each chain score is determined by the semantic relation and also the distance between the two chain members. Under the assumption that the text is cohesive, the higher scoring interpretation is retained as the best possible interpretation.

This method of retaining all possible interpretations, until the end of the process, causes the exponential growth of the time and space complexity. Barzilay and Elhadad dealt with this problem by discarding the "weaker interpretations", based on their scores, when the number of interpretations exceed certain threshold.

Silber and McCoy's algorithm:

Silber and McCoy (2002) implemented the Barzilay and Elhadad (1997) method of lexical chain computation in linear time. They created an internal representation, *meta-chains*, to implicitly store all the interpretations in order to reduce the runtime of the algorithm. Each meta-chain value is equal to the *offset* value in WordNet.

Words are inserted into those meta-chain entries with which they have the relations such as *identical, synonymy, hypernym/hyponym, sibling*. Score of each relation is determined by the semantic relation between the two words and also the distance between them in the text. Each meta-chain score is computed as the sum of scores between each relation in the chain. This process continues until all the words in the text are inserted into the meta-chains. Now, the words from the text are processed again and for each word instance, it is retained in the meta-chain to which it contributes the most (based on the meta-chain scores).

Galley and McKeown

Galley and McKeown (2003) first identify the sense of the word. Their approach can be classified into the three stages:

1. Building representation of all possible words.

2. Disambiguation of all the words.

3. Computation of the chains.

At first, an implicit representation of all possible word interpretations in the text called *disambiguation graph*, is created in linear time. Each node represents the word in the text and is divided into portions to represent the various senses of the word in WordNet. Edges connecting the nodes represent the weighted relation between the two particular senses. Each edge is given weight based on the type of semantic relation and proximity between the two words.

Once every word is processed, the disambiguation graph is used to identify the sense of the word based on its relations. Each node is processed to retain only the sense with the highest score, determined by the sum of weights. Once all the nodes have been narrowed down to one sense, the semantic links from the graph not compatible with the retained sense are discarded. Residual edges from the graph are then considered as the lexical chains for the text.

Stokes algorithm

Stokes (2004) proposed an enhanced version, LexNews, of the Hirst and St-Onge (1997) algorithm. Their approach tend to differ from the previous methods in the following ways:

- LexNews includes the domain specific information into the lexical chaining process in the form of statistical word association. These statistical word collocations tend to identify the topically related words, such as 'tennis', 'ball', 'net' and also missing compound noun phrases such as 'suicide bombing' or 'peace process', which are not present in WordNet.

- LexNews also includes the proper nouns in the chaining process. This is important when dealing with the text in news domain and can be used to build distinct set of chains.

2.5.2 Our algorithm

We considered nouns, compound nouns and proper nouns as candidate words to compute lexical chains. This is based on the intuition that nouns characterize the topic in the documents and that most of the documents describe a certain topic or a concept having various topics.

Each candidate word is expanded to all of its senses. In case of compound noun, only those compound nouns that have a valid entry in WordNet are retained (e.g.:

weather condition). If the compound noun does not have a valid entry, the modifiers are removed and only the main noun is considered (e.g.: In word "distribution graph", only the noun *graph* is considered for lexical chains and word *distribution* is discarded as the modifier).

We created a *hash structure* representation to identify all possible word representations, motivated from Galley and McKeown (2003). Each word sense is inserted into the *hash entry* having the index value equal to its *synsetID*. For example, *celebration* and *jubilation* are inserted into the same *hash entry* (Figure 2.5).
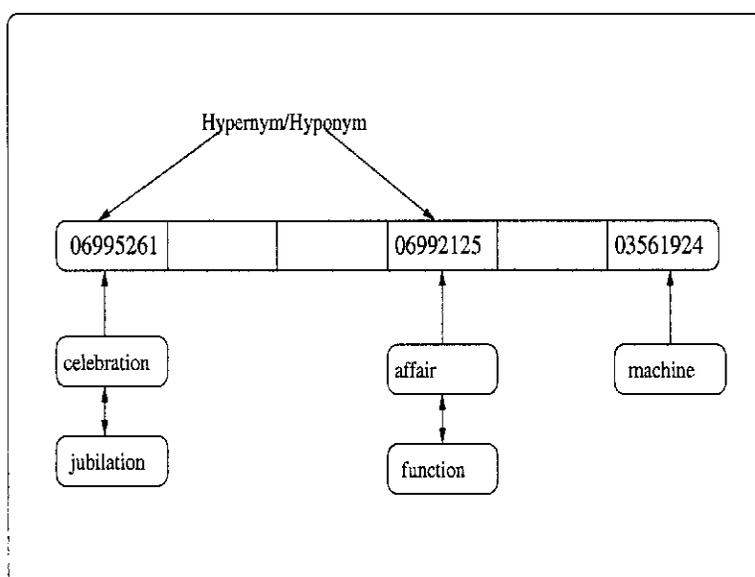
Figure 2.5: Hash structure indexed by synsetID value

On insertion of the candidate sense into the hash structure, we check to see if there exists an entry into the index value, with which the current word sense has one of the following relations:

For each candidate sense inserted, we check to see if it is related (semantically) with any of the already present members in the structure. The relations considered are:

- Identical relation:

    eg:- *Weather* is great in Atlanta. Florida is having a really bad *weather*.

- Synonym relation (words belonging to the same synset in WordNet):

    eg:- Not all *criminals* are *outlaws*.

- Hypernym/Hyponym relation:

    eg:- Peter bought a *computer*. It was a Dell *machine* .

- Siblings (If the words have the same hypernym):

    eg:- *Ganges* flows into the Bay of Bengal. *Amazon* flows into the South Atlantic.

- Gloss (If the concept is present in the gloss of the word):

    eg:- gloss of word *"dormitory"* is {a college or *university* building containing living quarters for students}

Each relation is scored based on the distance (dist) between the two concepts in WordNet hierarchy $(1/(dist + 1))$ (Table 2.4).

Relation	Score
Identical	1
Synonym	1
Hypernym/Hyponym	0.5
Sibling	0.33
Gloss	0.4

Table 2.4: Score of each relation (based on the length of path in WordNet)

For each candidate word sense, we identify the chains in which there exists a relation with each and every member of the chain. If found, we insert the word sense into the chain and update the score of the chain. Chain score is computed as the *sum of scores of each relation in the chain* which also includes the repetition count of each word.

$$score(chain) = \sum_{i=1}^{n}(score(R_i)) \qquad (2.1)$$

Where R_i is the semantic mesaure between two members of the lexical chain. Once the chains are computed, we sort the chains based on their score to determine the strength of the chains. We then filter out the chains which are not compatible with the higher ranked chains (i.e. having word from a higher ranked chain used in different sense). We retain the rest of the chains, which do not have words used in different sense to ones already assigned by higher ranked chains.

This process of retaining only certain chains enables us to disambiguate the sense of the word being used in a particular context. This property can be used to evaluate the efficiency of lexical chaining algorithms based on their efficency to correctly disambiguate the sense of a word.

2.6 Discussion

Lexical chains are sequences of semantically related words, which represents the cohesive ties in the text. Several methods have been proposed to compute lexical chains. Almost all of the methods use WordNet to identify the semantic relations between the words. In this chapter, we explained several methods used to compute lexical chains. We then proposed our own method to compute lexical chains, which includes gloss relations in the computation of lexical chains.

In the next chapter, we detail the methods to extract the sentences from the document based on the lexical chains to satisfy user's reuqests. We explained the methods to generate summaries for three specifc tasks: Headline generation, Multi-document summarization, and Query based summarization.

Algorithm 1 Algorithm to compute lexical chains

1: Start.
2: **for all** candidate words **do**
3:     expand the words into possible senses $(S_1, S_2, \ldots, S_n)$.
4:     determine the "offset" for each sense in WordNet.
5: **end for**
6: **for all** senses **do**
7:     insert the sense into the respective element of the *synsetID* list.
8:     **if** inserted synset has relations with already inserted synsets **then**
9:         identify the relation and determine their score.
10:     **end if**
11: **end for**
12: **for all** relations **do**
13:     identify the chains compatible with the current relation.
14:     **if** compatible chain is found **then**
15:         insert into chain by looking out for repetition.
16:         update the chain score.
17:     **else**
18:         create a new chain for the relation.
19:     **end if**
20: **end for**
21: sort the chains in descending order based on the chain scores.
22: **for all** chains **do**
23:     **for all** chainmembers **do**
24:         **if** chain member already assigned a sense **then**
25:             **if** assigned sense is not equal to the current chainmember sense **then**
26:                 $FLG \leftarrow FALSE$
27:             **end if**
28:         **else**
29:             assign the chain member the sense temporarily.
30:         **end if**
31:     **end for**
32:     **if** FLG equals to FALSE **then**
33:         discard the chain.
34:     **else**
35:         assign the chain members their respective senses from the temporarily stored values
36:         retain the chain.
37:     **end if**
38: **end for**
39: stop.

Chapter 3

System Design and Implementation

Summarization, as carried out by humans, can be divided into two stages (Jones, 1993):

1. Building of intermediate representation.

2. Synthesis of intermediate representation to generate summary.

Barzilay and Elhadad (1997) and Silber and McCoy (2002) established that lexical chains can be used as an efficient intermediate representation for the source. We describe a system to compute lexical chains as an intermediate representation for the source and extract sentences as summaries to satisfy certain criteria.

The architecture of our summarization system is shown in Figure 3.1. Source documents are divided into smaller segments based on the topical structure, and lexical chains are computed for each segment. Lexical chains computed are then used to extract the sentences from the source considered salient to the user needs.

3.1 Document processing

We parse the XML format source documents to filter the header tags and extract the textual information. Input text, free of the XML header tags, is then *tokenized* to separate each word into individual tokens, using the OAK English analyzer tools.[1]

---

[1]http://nlp.cs.nyu.edu/oak/
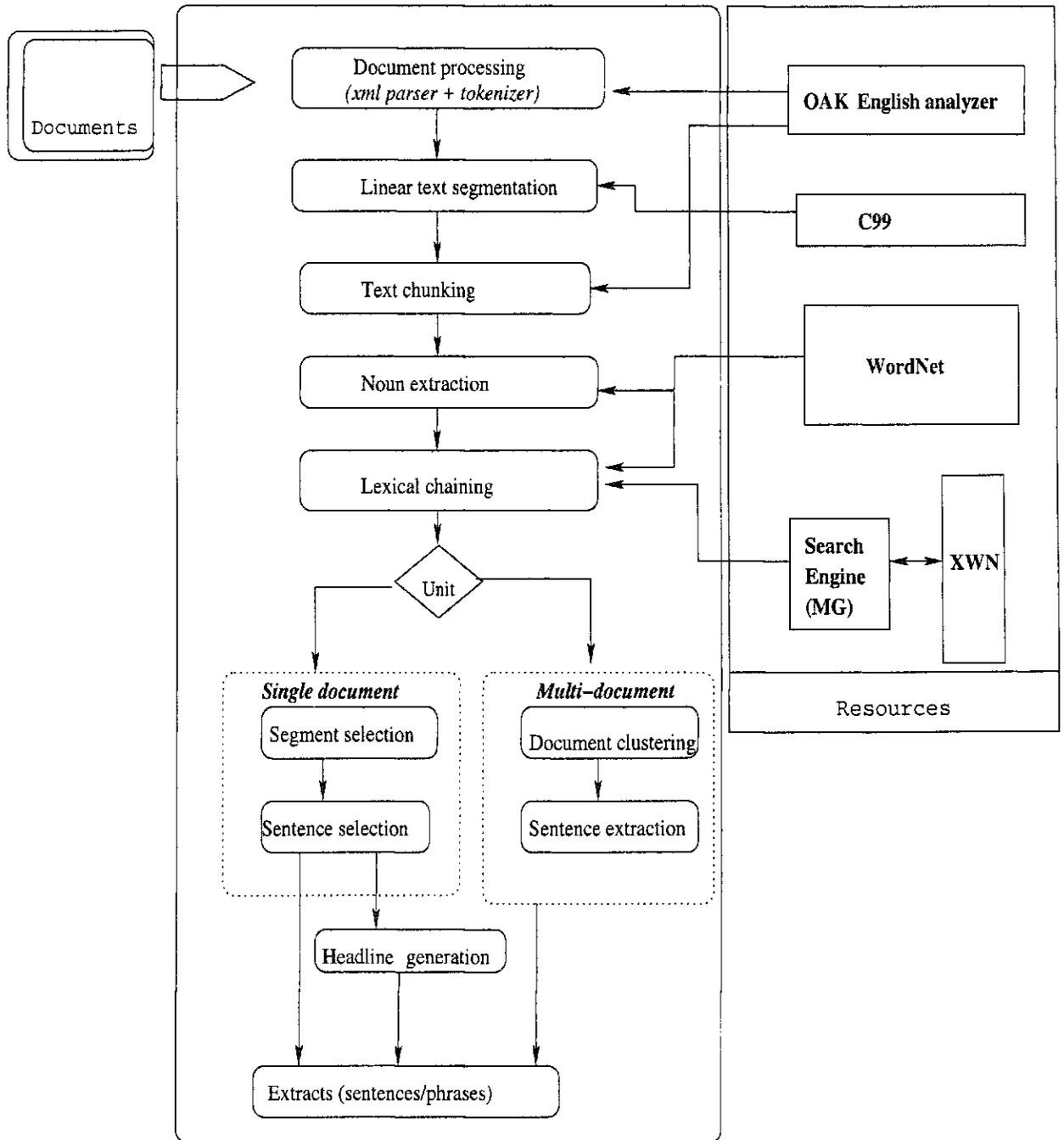
Figure 3.1: Architecture of the Summarizer

For example,

> Input: Pierre Vinken will join the board as a non-executive director. Mr.
> Vinken is the chairman of Elsevier.

> Tokenized text: Pierre Vinken will join the board as a non-executive
> director .
> Mr. Vinken is the chairman of Elsevier .

In the above example, every sentence starts (in the tokenized text) starts on a new line and each word is transformed into separate token (e.g.:- "Elsevier." to "Elsevier .").

3.2   Linear text segmentation

Any medium to large size article contains multiple topics or various events related to a topic. Linear text segmentation is a method of dividing large texts into smaller segments, based on the topical structure.

Segmentation is extremely useful in the areas of information retrieval and summarization. By dividing the document into smaller segments based on topic boundaries, it enables the summarization system to efficiently analyze the discourse structure. In the case of information retrieval, it provides direct access to the relevant portions of the document for a given query word.

Several methods have been proposed to carry out segmentation. Reynar (1999) identified the topical structure of the document based on the presence of cue-phrases, repetition of named entities etc. Hearst (1997) constructed the topical boundaries based on the distribution of lexical chains. Jobbins and Evett (1998) used linguistic features such as word repetition, and collocation to identify the change in the subject of discussion.

In our work, we use C99 (Choi, 2000) to perform linear text segmentation. C99 takes tokenized text as input and computes the similarity measure between each sentence of the text using the cosine similarity measure (Rijsbergen, 1979). A rank matrix is then computed using the similarity matrix to determine the relative ranking of each sentence in the local region. Finally, document is divided into segments at the point of maximum shift in topic boundaries identified using Reynar (1998) maximization algorithm. Experimental evaluation, Choi (2000), shows that the C99 algorithm is more accurate than almost all of the algorithms.

3.3 Text chunking

Text chunking can be defined as the process of dividing the sentence into a set of non-overlapping chunks. Chunks can be identified by observing the application of stress on certain portions and also the pause/duration, followed by humans while reading a particular statement (Abney, 1991) (Ramshaw and Marcus, 1995). For example:

> Sentence: I begin with an intuition: when I read a sentence, I read it a chunk at a time.
>
> Chunk: [I begin] [with an intuition]: [when I read] [a sentence], [I read it] [a chunk] [at a time].

Each chunk consists of one "content word" surrounded by certain "function words" and is categorized based on the syntactic category of the function word. Identification of these chunks can be done using certain hard template rules and finite state methods. As a result, phrasal structures can be identified without the parse representation.

In our system, we use the chunker belonging to OAK tool set to perform text chunking. It uses the well established chunking technique (Ramshaw and Marcus, 1995). Generation of chunk representation completes the pre-processing stage of

the document. Chunk representation can now be processed to extract the candidate words (nouns, proper nouns) for computation of lexical chains.

3.4 Noun extraction

In this module, *noun phrases* are extracted as candidate words from the chunked representation. Intuition behind selection of noun phrases and not just nouns is to identify the compound relations present in the document. Barzilay and Elhadad (1997) shows the disadvantage of not considering the compound form of words in formation of lexical chains. For example, consider the candidate words *"election"*, *"judicial writ"*, and *"writ of election"*. Only by retaining the compound form of the nouns, we are able to identify the *hypernym* relation between the words "judicial writ" and "writ of election". On the contrary, it would result in identification of wrong relation between "election" and "election" (non compound form of "writ of election"). This would result in the creation of wrong lexical chain, and so should be avoided.

3.5 Lexical chaining

Candidate words extracted from each segment are used to compute lexical chains, as explained in chapter 2. Lexical chains computed can be used as an intermediate representation of the source (Silber and McCoy, 2002) to extract coherent sentences as the summary of the source (Barzilay and Elhadad, 1997).

Based on this, we extract the sentences to obtain a cohesive summary for the document. Lexical chain approaches, until now, are used to compute single document summaries (Barzilay and Elhadad, 1997), (Silber and McCoy, 2002). Since lexical chains efficiently identify the theme of the document, we investigate methods to group the topically related units of a multi-document collection into *clusters* based on the overlap of lexical chains. Sentences can then be extracted from each

cluster to generate both generic and user-focused summaries for the given document collection.

In the following sections of this chapter, we detail the various extraction methods to generate the summaries based on the user's request. Our focus is to generate summaries for the single and multiple document source that satisfy certain user's requests.

3.6   Single document summarization

Single document summary can be generated by extracting the relevant sentences from the document. It consists of the following steps:

1. Segment selection.

2. Sentence selection.

3.6.1   Segment selection

Selection of the important segments involves relative ranking of the segments based on their contribution towards the document content. In current work, we performed segment ranking based on the Hoey (1991) principles of lexical cohesion:

1. Relevance of a high informational content word towards the document content or aptness can be determined based on its frequency.

2. Connectedness between two documents can be directly determined by the number of concepts shared between them.

Using these principles, saliency of a segment can be determined by the number of lexical chains it has in common with various segments. For example, consider the number of lexical chains shared by the 6 segments of a document (Table 3.1).

Once we compute the score of each segment with respect to the number of lexical chains shared with the remaining segments, we can sort the segments to determine their relative importance: In above example, the order would be:

#4 > #5 > #3 > #1, #2 > #6.

Hence we deduce that segment #4 is the most salient towards the document content.

	#1	#2	#3	#4	#5	#6	#total
#1	-	1	3	4	2	0	10
#2	1	-	5	2	1	1	10
#3	3	5	-	3	0	0	11
#4	4	2	3	-	5	1	15
#5	2	1	0	5	-	4	12
#6	0	1	0	1	4	-	6

Table 3.1: Relative ranking of segments

3.6.2 Sentence selection

Once segments are ranked based on their saliency towards the document content, we then select important sentences in the segments based on their contribution towards the segment content. Relative importance of each sentence is determined by the number of lexical chains shared in common with the rest of the sentences. This method of sentence ranking is similar to that of segment ranking procedure explained in Section 3.6.1. We then extract the top ranked sentences from the top ranked segments - i.e. top rank sentence from top ranked segment, top rank sentence from second ranked segment and so on - until the desired compression rate is achieved. Finally, we re-arrange the extracted sentences based on their position in the source document.

3.6.3 Headline generation

Very short summaries of the document can be used as indicative summaries to help the user identify the relevant documents in a digital library. In current work, we propose methods to compress the sentence by filtering out the portions of the text that do not contribute towards the meaning of the sentence.

Grefenstette (1998) performed sentence compression by retaining portions based on their syntactic structure. This was mainly proposed to compress the telegraphic text transferred and could easily be read by blind. (Knight and Marcu, 2000) proposed a probabilistic approach to compress a given sentence. They considered the input as a corrupted message which contains some words not contributing towards the meaning of the sentence. Headline, accordingly, can be obtained by eliminating the "noise" from the input sentence. Dorr, Schwartz, and Zajic (2002) used the Hidden Markov Model to retain certain portions of the sentence.

Dorr, Schwartz, and Zajic (2003) proposed a method to generate headlines by iterative elimination of certain content words. They generated a parse structure using the first sentence of the text and eliminated iteratively certain portions of the sentence to obtain an informative headline. Our method is motivated from their approach in that we iteratively eliminate certain phrases/clauses based on the syntactic structure of the sentence.

Input to this module is the parse structure of the top ranked sentences of the text. The given parse representation is then iteratively processed to eliminate certain portions, without loss of the meaning.

Elimination of the Sub-ordinate clauses: Sub-ordinate clauses, generally, are the supportive clauses in the sentence, which do not have any meaning without the main clause. Corston-Oliver and Dolan (1999) found that by not indexing the words present in the sub-ordinate clause, they can achieve the same precision at smaller index size.

Sentence: **The leaders of Malaysia's ruling party met Tuesday to discuss a replacement for ousted deputy prime minister Anwar Ibrahim,** who faces trial next month in a case that will test the country's legal system.

Parse: (S(S The leaders of Malaysia's ruling party met Tuesday to discuss a replacement for ousted deputy prime minister Anwar Ibrahim),(S (SBAR *who faces trial next month in a case that will test the country's legal system.*))

Output: The leaders of Malaysia's ruling party met Tuesday to discuss a replacement for ousted deputy prime minister Anwar Ibrahim.

Elimination of determinants, pronouns:

Sentence: **Wall Street extended** a **global stock selloff Thursday with** the **Dow industrials tumbling more than 200 points for a second straight day.**

Output: Wall Street extended global stock selloff Thursday with Dow industrials tumbling more than 200 points for second straight day.

Eliminate the noun modifiers:

Sentence: **The V-chip will give the parents a** new and potentially revolutionary **device to block out programs they don't want their children to see.**

Output: The V-chip will give the parents a device to block out programs they don't want their children to see.

Eliminate the adverbial phrases:

Sentence: **Dwight C. German said the study by Brzustowicz and colleagues** really **may well be a landmark paper.**

Output: Dwight C. German said the study by Brzustowicz and colleagues may well be a landmark paper.

Eliminate the prepositional phrases:

Sentence: **India's foreign secretary flew to Bangladesh** on Sunday for high-level talks.

Output: India's foreign secretary flew to Bangladesh.

Eliminate specifications in noun phrases:

Sentence: **Schizophrenia patients gained** some **relief after researchers sent magnetic field into** a small area of **their brains**.

Output: Schizophrenia patients gained relief after researchers sent magnetic field into their brains.

3.7 Multi-document summarization

Multi-document summarization involves identification of salient concepts across the collection of closely-related articles, while removing the redundancy and considering the similarities and the differences in the information content (Mani, 2001).

Multi-document summaries are frequently used to summarize news articles detailing with the same events or different phases of an event. Newsblaster (McKeown et al., 2003) gathers the news from various online news resources and groups them into meaningful *clusters* using SIMFINDER (Hatzivassiloglou et al., 2001) tool. Clusters are then used to generate the summary for the related articles. A typical multi-document summarization approach consists of the following tasks:

1. Identification of similar portions of text into a group or cluster.

2. Selection of salient sentences from each cluster.

3. Extraction/Re-generation of summary based on the selected sentences until desired compression rate is achieved.

Our approach to multi-document summarization is to cluster the related text units (segments of the collection) into meaningful clusters and then extract sentences from each cluster to generate a coherent summary.

3.7.1 Document clustering

Document clustering refers to the method of assigning the documents to a finite set of groups, *clusters*, based on associations among features within the documents (Hearst, 1999). Document clustering techniques are used in variety of applications: to organize the retrieved document collection for a user's query in an information retrieval system; to group the various conversations in an electronic meeting (Roussinov, 1999), etc. Clustering is useful in multi-document summarization to identify the various "themes" or events present in the collection. We use clustering techniques to group the segments into clusters based on their similarity.

Document clustering methods can be broadly characterized into the following categories:

1. Hierarchical methods.

2. Non-hierarchical methods or partition based methods.

Hierarchical methods organize the given document set into a tree based structure, depicting the "topic-subtopic" relations as "parent-child" relation of the tree. One major drawback of these methods is that objects once placed into clusters cannot be moved to another cluster (if the later cluster contains more similar documents than that of the former). Non-hierarchical methods, such as K-Means (Kohonen, 1989), group the documents based on some randomly initialized centroid documents for predefined number of clusters. The centroid values are recomputed after

each iteration and documents closer to the centroid points are placed in the respective clusters. This process continues till there is no change in the recomputed values of the centroid. A major disadvantage of these methods is the probable incorrect initialization of the centroid values resulting in inaccurate partitioning of the document collection.

We used the document clustering techniques to cluster the segments of the document collection. It involves the following steps:

1. Computation of similarity measure between the segments.

2. Grouping the documents into clusters using algorithm (Algorithm 2).

Similarity measure:

Good similarity measure is the key to document clustering. Documents are grouped into *clusters* based on the similarity value such that the objects present in one cluster are more similar to each other than the objects present in the other clusters.

Similarity measure is computed based on overlap of certain features (words, phrases, etc). In our approach, we compute the similarity measure using the linguistic features such as nouns (simple and compound), proper nouns. This is based on widely used principle that nouns describe the events in the document.

Nouns can be used in different senses - word "cone" in WordNet has the following senses :

- cone – (any cone-shaped artifact)

- cone, cone cell, retinal cone – (visual receptor cell sensitive to color)

As evident, cone can refer to a shape or a visual receptive cell in body. Distinction is required to be made in the computation of the similarity measure to not consider the overlap of these kinds of relations. We perform Word Sense Disambiguation (WSD) to identify the sense of the word being used in the given context.

Lexical chains, by definition, are sequence of semantically related words and they narrow down the sense of the word being used (Morris and Hirst, 1991). This property of lexical chains can also be used to disambiguate the nouns in the context of a given segment. Based on these principles, the nouns are divided into two categories:

1. Ambiguous nouns: Nouns, whose senses cannot be determined (i.e. not present in the lexical chains). Let f_j^i be frequency of the "ambiguous" word j in segment i, similarity measure $(sim_{am}^{(a,b)})$ between segments a,b can be computed by using the cosine similarity measure (Rijsbergen, 1979) as follows:

$$sim_{am}^{a,b} = \frac{\sum_{j=1}^{n} (\frac{1}{k_j} * (f_j^a) * (f_j^b))}{\sqrt{\sum_{j=1}^{n} ((f_j^a)^2) * \sum_{j=1}^{n} ((f_j^b)^2)}} \quad (3.1)$$

where k_j is the number of possible senses for the word j.

2. Dis-ambiguous nouns: Nouns whose sense in the segment can be determined from lexical chains. Let f_j^i be frequency of the word j in segment i, similarity measure $(sim_d^{(a,b)})$ between segments a,b can be computed by using the cosine similarity measure as follows:

$$sim_d^{a,b} = \frac{\sum_{j=1}^{n} ((f_j^a) * (f_j^b))}{\sqrt{\sum_{j=1}^{n} ((f_j^a)^2) * \sum_{j=1}^{n} ((f_j^b)^2)}} \quad (3.2)$$

Along with these two measures, we compute the third measure based on the number of proper nouns shared between the two segments. Since proper nouns always refer to names of person, place or organization, we do not perform WSD and compute the similarity based on the frequency of proper nouns common to both segments. Let f_j^i be frequency of the "proper noun" j in segment i, similarity measure $(sim_{ppn}^{(a,b)})$ between segments a,b can be computed by using the cosine similarity measure as follows (W can be additional weighting factor):

$$sim_{ppn}^{a,b} = W * \frac{\sum_{j=1}^{n} ((f_j^a) * (f_j^b))}{\sqrt{\sum_{j=1}^{n} ((f_j^a)^2) * \sum_{j=1}^{n} ((f_j^b)^2)}} \quad (3.3)$$

Once these three measure are computed, we take the average of the three measures to compute the similarity $Sim(a,b)$ between two segments a and b.

$$Sim(a,b) = \frac{sim_d^{a,b} + sim_{am}^{a,b} + sim_{ppn}^{a,b}}{3} \tag{3.4}$$

Clustering algorithm:

Our approach to group the segments into clusters (Algorithm 2) consists of the following steps:

. Cluster construction.

. Removal of any overlaps between the clusters.

For each segment (S_i), we include all the segments (S_j) into a cluster, if the similarity (S_i, S_j) is greater than certain threshold value. It should be noted that one segment will be in more than one *cluster* of segments. The next step is to remove the overlap of segments.

Each segment is retained in the cluster in which it contributes the most. This is determined on the basis of cluster score, computed based on the similarity score between the segments contained in it.

3.7.2 Sentence extraction

The main purpose of the clustering process is to organize the segments of the document collection based on their theme. This is important to identify and extract the portions of the documents, relevant to the given user's application. Summaries can thus be generated by extracting sentences from each clusters (Hatzivassiloglou et al., 2001). We extract the sentences from the clusters based on lexical chains of the document collection. We first score the clusters using the TFIDF (term frequency/inverse document frequency) term weighting scheme (Salton, Allan, and Buckley, 1994):

$$score(cluster_i) = \sum_{j=1}^{m} \frac{score(chainMember_j, cluster_i)}{clusters_j} \tag{3.5}$$

where $score(cluster_i)$ is the score of $cluster_i$, $score(chainMember_j, cluster_i)$ is the number of occurrences of $chainMember_j$ in $cluster_i$, $clusters_j$ is the number of clusters having the $chainMember_j$.

Once we select the top ranked clusters, we score the segments in the selected clusters, using the same TFIDF scheme:

$$score(segment_i) = \sum_{j=1}^{m} \frac{score(chainMember_j, segment_i)}{segments_j} \tag{3.6}$$

where $score(segment_i)$ is the score of $segment_i$, $score(chainMember_j, segment_i)$ is the number of occurrence of the $chainMember_j$ in $segment_i$, $segments_j$ is the number of segments having the $chainMember_j$.

Once ranking the segments, we rank the sentences based on the frequency of lexical chains. We then extract the top ranked sentences from the top ranked segments of the top ranked clusters. Summaries can be generated by ordering the sentences based on their position in the source collection (documents in the source collection are sorted based on their time stamps).

3.8 Query based summarization

We extracted sentences from the given document collection with respect to certain key "entity", such as name of a person. Primary objective of this method is to produce an informative summary about the various events related to the person.

Sentences from each cluster are selected based on the following principles:

- Sentences that do not begin with a pronoun.

- Sentences that do not have some quotations;

- Sentences that have the entity name.

Sentences, which satisfy these constraints are then extracted and ordered on basis of their position.

Algorithm 2 Algorithm to cluster segments

Require: Similarity measure $sim(a,b)$ between all the segments.

1: **for** each segment S_i in the document collection **do**
2:     **for** each segment S_j $(j \neq i)$ in the document collection **do**
3:        **if** $sim(S_i, S_j) >= thresholdvalue(th)$ **then**
4:           include S_j into the *related segments list* of S_i
5:        **end if**
6:     **end for**
7: **end for**
8: **for** segment S_i in collection **do**
9:     **for** each cluster C_j (cluster of segments) **do**
10:        **if** S_i has similarity value $>= th$ with all cluster members **then**
11:           include the segment S_i in cluster C_j.
12:           update the cluster score; cluster score = sum of similarity value between segments.
13:        **end if**
14:     **end for**
15:     **if** segment S_i not included in any cluster **then**
16:        create a new cluster
17:     **end if**
18: **end for**
19: **for** each segment S_i of the collection **do**
20:     **for all** clusters that contains the segment S_i **do**
21:        identify the cluster in which the segment contributes the most
22:     **end for**
23:     update the clusters by retaining the segment only in cluster in which contributes the most.
24: **end for**
25: Output the clusters as final clusters of the collection.

Chapter 4

Experimental Evaluation

Evaluation methods can be broadly classified into two categories (Mani and May-bury, 1999): *intrinsic* and *extrinsic*. Extrinsic methods of evaluation, rate the summaries based on their ability to perform certain task (Information retrieval etc). Intrinsic methods of evaluation determine the quality of the summaries based on the overlap with human generated *"ideal summaries"*. In the intrinsic evaluation, precision and recall are the widely used measures computed based on the number of units (sentences, words, etc) common to both system-generated and ideal summaries. Precision (P) is defined as the percentage of system-generated summary in common with the ideal summary. Recall (R) is defined as the ratio of the number of units (sentences/words) of the system-generated summaries in common with the ideal summaries to total number of units in the ideal summaries. Another measure, F-measure is a composite score that uses β factor to weight the relative importance of precision and recall measures:

$$F - measure = \frac{(1 + \beta^2)R * P}{R + \beta^2 P} \tag{4.1}$$

We evaluated our summarization techniques using the test data provided by NIST (National Institute of Standards and Technology). We carried out automatic evaluation of our summaries using ROUGE (Lin, 2004). Manual evaluation was

47

performed by human judges as a result of direct participation in Document Understanding Conference, [1] using the Summary Evaluation Environment (SEE). [2]

4.1 ROUGE

ROUGE (Recall-Oriented Understudy of Gisting Evaluation) is a collection of measures to automatically evaluate the summaries by comparing them with "ideal" summaries, without much of human intervention. Quality of the summaries is determined by the number of *n-gram* (sequence of *n* words) overlaps between the two summaries. ROUGE measures considered in the evaluation are: ROUGE-N (n= 1,2,3,4), ROUGE-L, ROUGE-W.

4.1.1 ROUGE-N

ROUGE-N, a recall based measure, is measured by the number of *n-gram* overlaps ($n = 1,2,3,4$) between the reference and system generated summaries. It is computed as follows:

$$= \frac{\sum_{S\in\{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \qquad (4.2)$$

where *n* stands for the length of the *n-gram*, $gram_n$ and $Count_{match}(gram_n)$ is the maximum number of n-grams common to both summaries.

When multiple references are used for evaluation, pairwise summary-level ROUGE-N score between the candidate summary "*s*" and every reference summary "r_i" is first computed. Final multiple reference ROUGE-N score is then obtained by taking the maximum of the summary-level ROUGE-N scores computed.

$$ROUGE - N_{multi} = argmax_i(ROUGE - N(r_i, s)) \qquad (4.3)$$

---

[1] http://duc.nist.gov
[2] http://www.isi.edu/~cyl/SEE

Given M references, the best score over M sets of M-1 references is computed. The final score is the average of the M ROUGE-N scores using different M-1 references. This method is known as *Jacknifing* procedure and is applied to all ROUGE measures in the ROUGE evaluation package. For example, consider a document having words w1, w2, ..., w25 and having five sentences A1, A2, A3, A4 and A5 as follows:

A1 = w1 w2 w3 w4 w5

A2 = w6 w7 w8 w9 w10

A3 = w11 w12 w13 w14 w15

A4 = w16 w17 w18 w19 w20

A5 = w21 w22 w23 w24 w25

Given three peer summaries S1, S2, S3 and three reference summaries R1, R2,R3:

- R1 consists of A1, A2; R2 consists of A3, A4; R3 consists of A2 and A5.

- S1 contains A1, A2; S2 contains A1, A3; S3 contains A1 and A1.

- $|x|$ refers to the unigram length of x.

- ROUGE(x\R) is ROUGE score of x without reference R.

Using R1, R2, R3 as references, $ROUGE_1$ scores, using *Jacknifing* procedure, for the three peer summaries can be computed as follows:

- ROUGE1(S1\R1) = $|A2| /(|A3| + |A4| + |A2| + |A5|)$ = 1/4,
  ROUGE1(S1\R2) = $(|A2| + |A2| + |A1|) /(|A1| + |A2| + |A2| + |A5|)$ = 3/4,
  ROUGE1(S1\R3) = $(|A2| + |A1|) /(|A3| + |A4| + |A2| + |A5|)$ = 2/4,
  ROUGE1(S1) (Avg) = 0.5.

- ROUGE1(S2\R1) = $|A3|$ /($|A3|$ + $|A4|$ + $|A2|$ + $|A5|$) = 1/4,

  ROUGE1(S2\R2) = ($|A1|$) /($|A1|$ + $|A2|$ + $|A2|$ + $|A5|$) = 1/4,

  ROUGE1(S2\R3) = ($|A3|$ + $|A1|$) /($|A3|$ + $|A4|$ + $|A2|$ + $|A5|$) = 2/4,

  ROUGE1(S2) (Avg) = 0.33.

- ROUGE1(S3\R1) = 0/($|A3|$ + $|A4|$ + $|A2|$ + $|A5|$) = 0,

  ROUGE1(S3\R2) = ($|A1|$) /($|A1|$ + $|A2|$ + $|A2|$ + $|A5|$) = 1/4,

  ROUGE1(S3\R3) = ($|A1|$) /($|A3|$ + $|A4|$ + $|A2|$ + $|A5|$) = 1/4,

  ROUGE1(S3) (Avg) = 0.17.

Based on the average values computed using the multiple references, it can be inferred that S1 is ranked higher than S2 which in turn is ranked higher than S3. From the above example, it is evident that ROUGE-N measure gives more priority to the summaries having more number of overlaps with a pool of summaries.

4.1.2 ROUGE-L

ROUGE-L measure is the value of the "longest common subsequence" in common between the system generated summary and ideal summary. This is based on the intuition that longer the subsequence of words in common, greater the similarity. Given a sequence $Z = [z_1, z_2, \ldots z_n]$ and sequence $X = [x_1, x_2, \ldots, x_n]$, Z is said to be the subsequence of X if there exists a strict increasing sequence $[i_1, i_2, \ldots, i_n]$ on indices of X such that for all j= 1,2,3, ..., k we have $x_{ij} = z_j$. Given two sequences A and B, the Longest Common Subsequence (LCS) is the sequence with the maximum length.

LCS-based F-measure at sentence level between two summaries X and Y of lengths m and n can be computed as follows:

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{4.4}$$

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{4.5}$$

$$F_{lcs} = \frac{(1 + \beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{4.6}$$

Where LCS(X,Y) is the longest subsequence overlap between X and Y, and $\beta = P_{lcs}/R_{lcs}$ when $\partial F_{lcs}/\partial R_{lcs} = \partial F_{lcs}/\partial P_{lcs}$. In DUC evaluation, $\beta- > \infty$ and so $F_{lcs} = R_{lcs}$. F_{lcs} is also known as ROUGE-L measure.

Consider the following example (Lin, 2004):

    S1: police killed the gunman.

    S2: police kill the gunman.

    S3: the gunman kill police.

Using S1 as reference, both S2, S3 have the same ROUGE-2 score even when they differ in meaning (both candidate sentences have just one bi-gram in common with the reference summary, "the gunman"). This can be differentiated using the ROUGE-L measure. Sentence S2 has the ROUGE-L value as 3/4 and sentence S3 1/2, ($\beta = 1$). This distinguishes between the similarity of sentence S1 with S2 and S3.

ROUGE-L has a major drawback that it counts only one main in-sequence words and thereby alternative or shorter sequences of LCS cannot be observed. For example,

    S4: the gunman police killed.

Sentence S4, has two sequences in common the reference sentence S1 ("the gunman" and "police killed"). Since LCS considers only the longest sequence, it gives the sentence S4 the same score as S3. Summary level LCS can be obtained by taking the union LCS matches between a reference summary sentence and every candidate summary sentence. This can be computed as shown below:

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{m} \tag{4.7}$$

$$P_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{n} \tag{4.8}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{4.9}$$

where $LCS_{\cup}(r_i, C)$ is the LCS score of the union of the longest common subsequence between the reference sentence and the candidate summary C. u,m refer to the number of sentences and number of words in reference summary, and v and n refer to the number of sentences and words in candidate summary. ($\beta- > \infty$ in current evaluations.)

4.1.3 ROUGE-W

Consider the following example:

X: [ABCDEFG]

Y1: [ABCDKJL]

Y2: [AHBKCID]

In the above example, both sentences Y1 and Y2 have the same ROUGE-L score of 4/7 ($\beta = 1$) with X as the reference. This would not reward the sentence Y1, which has consecutive sequence of words, as compared with Y2. ROUGE-W, weighted longest common sequence, measure provides an improvement to the basic LCS method of computation by using the function f(n) to credit the sentences having the consecutive matches of words. F-measure based on WLCS can be calculated as follows:

$$R_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right) \tag{4.10}$$

$$P_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right) \tag{4.11}$$

$$F_{wlcs} = \frac{(1+\beta^2)R_{wlcs}P_{wlcs}}{R_{wlcs}+\beta^2 P_{wlcs}} \qquad (4.12)$$

Where f^{-1} is the inverse function of f. F_{wlcs} measure computed above is called as ROUGE-W. By computing the ROUGE-W measure for the two candidate sentences in the above example $(f(k) = k^2)$, we obtains scores of 0.571 and 0.286 for Y1 and Y2 respectively. This enables us to differentiate between the two sentences based on the spatial distance between the sequence of the words.

4.1.4 Correlation with human evaluation

Lin (2004) compared the ROUGE evaluations with the human evaluations obtained from the three DUC evaluation series (DUC 2001, 2002 & 2003). Intention of this comparison was to see if ROUGE assigns a good score to good summaries and bad score to bad summaries. He arrived at the following conclusions:

1. ROUGE-2, ROUGE-L and ROUGE-W correlate strongly with human evaluation for single document summarization.

2. ROUGE-1, ROUGE-L and ROUGE-W achieved closer evaluation results in comparison to the human evaluation for very short summaries (headlines).

3. Correlation of above 90% with human judgment is hard to achieve for multi-document summaries evaluation. ROUGE-1, ROUGE-2 worked better when stop words are eliminated from consideration.

4. In general, correlation improved with the elimination of stop words except for ROUGE-1.

5. Multiple references improves the correlation with human evaluation for shorter samples of summaries.

4.2 Human evaluation using SEE

NIST carries out human evaluation of summaries, with the help of Summary Evaluation Environment (SEE) as follows:

1. Model summaries are divided into content units Elementary Discourse Units (EDU's) and system-generated summaries are divided into sentences.

2. For each model content unit identify the peer units that imply some facts of the model unit.

3. Once the peer units have been marked, determine the extent to which the content in the model unit is covered by the peer unit.

The extent to which the marked units of the peer summaries explain the concept of the model summaries can be *all, most, some, hardly any and none* depending on the extent of the explanation of marked peer units with respect to the content of the model unit. Recall score with respect to the coverage of the model unit content by the peer summary can be computed as follows:

$$C = \frac{(Number of MU's Marked) * E}{Total number of MU's in Model summary} \tag{4.13}$$

Where E is the ratio of completeness ranging between 0 and 1: 0 for *none*, 1/4 for *hardly any*, 1/2 for *some*, 3/4 for *most* and 1 for *all*.

Apart from the content of the summaries, human judges also determine the quality of the summaries generated with respect to various quality factors.

4.3 Experiments

We evaluated the summarization techniques using the data set provided by NIST in context of Document Understanding Conference (DUC) (Over, 2004). NIST provides a task description according to which the summaries are to be generated.

Along with the data, NIST also provides four "model" summaries for each document in each task. We performed evaluations by taking part in the following tasks: headline generation, multi-document summarization and query-based summarization.

4.3.1 Headline generation

In 2004, NIST provided a collection of 500 documents and defined the task as to generate a very short summary (approximately 75 bytes) for each document. Apart from the test data, NIST also provides four human generated *model* summaries for each document.

ROUGE evaluation:

Table 4.1 shows the results of the evaluation with ROUGE parameters set the same as in DUC 2004 evaluation. Our system ($System_B$) that took part in DUC 2004 achieved poor performance as compared to other systems. The reasons for this poor performance are: 1) consideration of only one sentence for headline generation. 2) intended to generate a readable headline causing the loss of content overlap with human summaries.

Another system, $System_A$, extracted the two most relevant sentences from the document. Sentences extracted were then compressed using the methods explained in chapter 3 to generate a headline. In case the generated headline is greater than 75 bytes, we preferred not to discard the extra bytes, as it is automatically performed by ROUGE. As compared to performances of all the other systems participated in DUC 2004, our system ($System_A$) is among the top ranked systems (4/31) with respect to the ROUGE-1, ROUGE-L and ROUGE-W measures.

Additionally, we experimented the effect of "removal of stop words" in ROUGE evaluation. We changed the parameter(s) of the ROUGE [3] such that stop words are

---

[3]http://www.isi.edu/ cyl/ROUGE/

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-W
System$_A$	0.22485	0.04745	0.00993	0.00230	0.18822	0.10189
System$_B$	0.12067	0.02765	0.00799	0.00270	0.10647	0.06537
Best system	0.25302	0.06590	0.02204	0.00766	0.20288	0.12065
Humans (Avg.)	0.29	0.08	0.03	0.01	0.24	0.13

Table 4.1: ROUGE evaluation of headline generation (without stopword removal)

not considered in the evaluation. We observed that there is a significant improve-
ment in the overall performance (Table 4.2).

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-W
System$_A$	0.26254	0.06489	0.01627	0.00321	0.22335	0.12826
Best system	0.29441	0.07500	0.02122	0.00489	0.23748	0.15241
Humans (Avg)	0.32	0.08	0.03	0.01	0.28	0.17

Table 4.2: ROUGE evaluation (with stopword removal)

Human evaluation, using SEE [4], was carried out for the participants in DUC
2003. Human judges evaluated the headlines using only one "ideal" summary for
the coverage. Our system acheived the best possible coverage (40%) among all the
systems.

4.3.2   Multi-document summarization

NIST provided 50 document collections, each containing 10 documents, and de-
fined the task as to generate a summary (max 665 bytes) for the given document
collection. We performed clustering to identify the various themes in the document
collection and extracted sentences from each cluster. Sentences are then ordered
based on the timestamp of the document they are extracted from, to generate an
indicative multi-document summary.

---

[4]http://www.isi.edu/~cyl/SEE

ROUGE evaluation:

Table 4.3 shows the results of ROUGE evaluation.

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-W
Our system	0.30352	0.04745	0.01178	0.00427	0.26164	0.09062
Best system	0.38232	0.09219	0.03363	0.01547	0.33040	0.11528

Table 4.3: ROUGE Evaluation for multi-document summarization

Human evaluation:

Human judges compared the "peer" summaries with one "manual" summary using SEE. Table 4.4 shows the results for the system generated summaries. Apart from the coverage, they also measure the quality of the summaries generated with respect to various quality questions (See Appendix A for a list of the questions).

System	Mean coverage
Our system	0.165
Best system	0.30
Avg of systems	0.21

Table 4.4: SEE evaluation of multi-document summarization

Table 4.5 shows the results for the quality of the summaries with respect to quality questions defined in Appendix A. Our system was ranked 8/17 with respect to the quality of the summaries generated.

System	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Mean
Our system	3.28	2.7	1.36	2.34	1.08	1.22	1.36	1.90
Best system	2.32	2.08	1.56	1.2	1.46	1.22	1.38	1.60

Table 4.5: Quality of the multi-document summaries.

4.3.3 Query-based summarization

NIST provided 50 document collections with each collection containing 10 documents. The description of the task was, given a document collection and a query of form "who is X?", where "X" is the name of a famous person, generate a summary (max 665 bytes) in response to the question.

ROUGE evaluation:

Table 4.6 presents the ROUGE evaluation of our system generated summaries in comparison with the best system.

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-W
Our system	0.30948	0.06957	0.02610	0.01290	0.27060	0.09438
Best system	0.35495	0.08571	0.03281	0.01476	0.31710	0.10970

Table 4.6: ROUGE Evaluation for query-based summarization .

Human evaluation:

Human evaluation was carried out using SEE. Table 4.7 shows the results of the coverage of our summaries. Human judges also evaluate the "responsiveness" of the summaries with respect to the given question (0 = worst, 4 = best). Our system is among the better systems with respect to the responsiveness. Table 4.8 shows the quality of the summaries with respect to various questions framed by NIST for DUC 2004 (Appendix A). Our system shared the top rank (2/15) with another system with respect to the quality of the summaries generated.

System	Mean Coverage	Responsiveness
Our system	0.198 (10/15)	1.42 (7/15)
Best System	0.24144	1.76
Avg. of systems	0.196	1.38

Table 4.7: SEE evaluation for query-based summarization

System	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Our system	2.9	2.42	1.38	2	1.46	1.3	1.3

Table 4.8: Quality of Query based summaries

4.4 Discussion

In this chapter, we presented methods to evaluate the summaries generated by our system. We used an intrinsic method using the test data provided by NIST and ROUGE evaluation measures. We achieved better results with respect to headline generation in both manual and automatic evaluation.

We observed that, much work is required in multi-document summarization to attain better coverage. Our system did well in the quality based evaluation and also among the top in the "responsiveness", with respect to the query-based summarization.

Human evaluation procedure has some disadvantages: Human judges performed the evaluation of the summaries using only one "model" summary. This is in direct contradiction to the well established principle that there does not exist a single "ideal" summary. Also it has been found that humans agree only to 82% of their prior judgments (Lin and Hovy, 2002).

Variability is also found between inter-human judgments, underlying the importance of having a better evaluation method to eliminate the variance. Nenkova and Passonneau (2004) based their evaluation procedure on the *summarization content units* (SCU), which are extracted from the summaries not larger than the clause. They grouped the SCU's obtained from the pool of human summaries into a tier's of a *pyramid* model. Each tier in the pyramid model consists of the SCU's that have the same weightage. So SCU's present in the nth tier level has more importance than those present in the $(n-1)^{th}$ level. With this approach, they can establish the similarity between the summaries and also efficiently determine the differences between the summaries.

Chapter 5

Conclusion and future work

5.1 Conclusion

In this thesis, we presented a method to compute lexical chains as an efficient intermediate representation of the document. Along with normal WordNet relations, our method also included additional relations such as proper noun repetition and gloss relations in the computation of lexical chains. We identified these additional relations using semantically enhanced tool, eXtended WordNet. The method to include gloss relations contributes towards the better understanding of the text and enhances text coherence (Harabagiu and Moldovan, 1998).

We then investigated methods to extract sentences from the document(s) based on the distribution of lexical chains. We proposed a method to generate the headlines, motivated from Dorr, Schwartz, and Zajic (2003), for a given document by filtering the portions not contributing towards the meaning of the sentence. We based our compression techniques on certain linguistically motivated principles.

Lexical chains, until now, were mainly used to generate single document summaries. Lexical chains help identify the *themes*, by clustering the document collection. Indicative multi-document summaries can then be generated by selecting clusters relevant to the user's criteria and extracting sentences from each cluster.

We performed intrinsic evaluation to determine the quality of the summaries generated by our approaches. We found that our system achieved better results in

60

headline generation and in query based summarization in context of DUC (Over, 2004).

5.2 Future work

We wish to pursue further research in the following directions:

Lexical chaining algorithm: Our method to compute lexical chains includes the gloss relations. These relations were based on the presence of gloss concept or synonym of the gloss concept in the text. We would like to pursue further research into the methods to compute the semantic similarity based on the overlap of the gloss concepts as in Banerjee and Pedersen (2003).

Lexical chains are evaluated based on their performance in identifying the sense of the word in given context. It has been proved that concepts present in the gloss of a word play an important role in the determination of the word sense (Lesk, 1986), (Banerjee and Pedersen, 2003). We would like to compare our system performance in this aspect with respect to other lexical chaining methods.

Document clustering: Document clustering is a key step towards the identification of various themes in a multi-document collection. Good similarity measure plays an important role in determining the overall efficiency of the clusters. We compute the similarity measure based on the overlap of nouns (used in same sense) between two segments. Based on the study that verbs play an important role in determining the "action" performed in the text (Klavans and Kan, 1998), we would like to investigate new methods to include the verb relations into the computation of the similarity measure.

WordNet-2.0 contains the relations between the verbs and nouns (e.g. summary − > (Verb) summarize). Also eXtended WordNet identifies the sense of the verbs in the gloss definition. Using these two resources, we wish to pursue further research

into the computation of the similarity measure.

Multi-document summarization: Multi-document summarization is still a complex and challenging task. One problem is to find method to extract sentences to compose a coherent summary. We would like to further investigate into this problem to implement an efficient method to extract sentences from each cluster.

We would like to use our sentence reduction techniques to eliminate certain portions of the extracted sentences, so as to include more content at the given compression rate.

Appendix A

Quality questions (DUC 2004)

1. Does the summary build from sentence to sentence to a coherent body of information about the topic?

   A Very coherently.

   B Somewhat coherently

   C Neutral as to coherence.

   D Not so coherently

   E Incoherent.

2. If you were editing the summary to make it more concise and to the point, how much useless, confusing or repetitive text would you remove from the existing summary?

   A None

   B A little

   C Some

   D A lot

   E Most of the text.

3. To what degree does the summary say the same thing over again?

    A None; the summary has no repeated information.

    B Minor repetitions.

    C Some repetition.

    D More than half of the text is repetitive

    E Quite a lot; most sentences are repetitive.

4. How much trouble did you have in identifying the referents of noun phrases in the summary? Are there nouns, pronouns or personal names that are not well-specified? For example, a person is mentioned and it is not clear what his role in the story is, or any other entity that is referenced but its identity and relation with the story remains unclear.

    A No problems; it is clear who/what is being referred to throughout.

    B Slight problems, mostly cosmetic/stylistic.

    C Somewhat problematic; some minor events/things/people/places are unclear, or very few major ones, but overall the *who* and *what* are clear.

    D Rather problematic; enough events/things/people/places are unclear that parts of the summary are hard to understand.

    E Severe problems; main events, characters or places are not well-specified and/or it's difficult to say how they relate to the topic.

5. To what degree do you think the entities (person/thing/event/place) were re-mentioned in an overly explicit way, so that readability was impaired? For example, a pronoun could have been used instead of a lengthy description, or a shorter description would have been more appropriate?

    A None; references to entities were acceptably explicit.

B A little: once or twice, an entity was over-described.

C Somewhat: to a noticeable but not annoying degree, some entities were over-described.

D Rather problematic: to a degree that became distracting, entities were over-described.

E A lot: re-introduction of characters and entities made reading difficult/caused comprehension problems.

6. Are there any obviously ungrammatical sentences, *e.g.*: missing components, unrelated fragments or any other grammar-related problem that makes the text difficult to read.

A No noticeable grammatical problems.

B Minor grammatical problems.

C Some problems, but overall acceptable.

D A fair amount of grammatical errors.

E Too many problems, the summary is impossible to read.

7. Are there any datelines, system-internal formatting or capitalization errors that can make the reading of the summary difficult?

A No noticeable formatting problems.

B Minor formatting problems.

C Some, but they do not create any major difficulties.

D A fair amount of formatting problems.

E Many, to an extent that reading is difficult.

Appendix B

Lexical Chains

Sample Document:

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. The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph. "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday. Cabral said residents of the province of Barahona should closely follow Gilbert's movement. An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo. Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night. 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. 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 weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday. Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast. There were no reports of casualties. San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night. On Saturday, Hurricane

66

Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast. Residents returned home, happy to find little damage from 80 mph winds and sheets of rain. Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane. The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

Lexical chains computed for the above text are:

weather storm wind rain

hurricane rain wind

month night season watch

mile foot

resident gilbert

movement coast

weather high_wind gust wind

puerto_rico san_juan

puerto_rico province

mile mph

puerto_rico virgin_islands u.s.

city san_juan miami florence

southeast u.s. west

caribbean southeast west

center position

midnight night

area wind

wind sheet

west virgin_islands u.s.

santo_domingo city

people

casualty damage

u.s. republic

cloudiness

television

season weather

area people

defense

civil_defense defense

gulf high_sea

strength

remnant sheet

weather cloudiness

movement flood

Appendix C

Sample System Generated Summaries

Following are the example summaries generated by our system for the document collection from the DUC 2004 (Over, 2004) test set.

- Headlines generation: Following are the sample headlines Figure C.1 generated for the DUC 2004 test set.

NYT19981107.0251
movement Islamic Holy War Saturday suicide bombing Jerusalem market Friday

APW19981118.0276
leader Hun Sen has safety freedom politicians to ease fears rivals be return to
country

APW19981026.0220
Cambodia 's opposition Asian Development Bank Monday to stop loans to government
weeks hope influence parties

APW19981105.1220
information Honduras countryside officials to lower death toll Hurricane Mitch to Thursday
leaders

Figure C.1: Sample headline summaries generated by the system

- Multi-document summarization: The following are the summaries generated for the multi-document summarization task using DUC 2004 test data. Figure C.2 is the summary for the document collection, which achieved poor performance, when evaluated in comparison with the human generated summaries. Figure C.3 is the summary for document set, which is closer to the human generated summaries.

Honduras braced for potential catastrophe Tuesday as Hurricane Mitch roared through the northwest Caribbean, churning up high waves and intense rain that sent coastal residents scurrying for safer ground. Hurricane Mitch cut through the Honduran coast like a ripsaw Thursday, its devastating winds whirling for a third day through resort islands and mainland communities. Agriculture Minister Pedro Arturo Sevilla said crucial grain, citrus and banana crops had been damaged "and the economic future of Honduras is uncertain."

Figure C.2: Multi-document summary for the document collection *d30002t*

- Query-based summarization: The following are the summaries generated for the query-based summarization task in DUC 2004. Figure C.4 is the summary for the document collection, which achieved poor performance, when evaluated in comparison with the human generated summaries. Figure C.5 is the summary for document set,which is closer to the human generated summaries.

King Norodom Sihanouk has declined requests to chair a summit of Cambodia's top political leaders, saying the meeting would not bring any progress in deadlocked negotiations to form a government. In a long-elusive compromise, opposition leader Prince Norodom Ranariddh will become president of the National Assembly resulting from disputed elections in July, even though Hun Sen's party holds a majority of 64 seats in the 122-member chamber. In a letter to King Norodom Sihanouk_ the prince's father and Cambodia's head of state_ that was broadcast on television Tuesday, Hun Sen said that guarantees of safety extended to Ranaiddh applied to all politicians.

Figure C.3: Multi-document summary for the document collection *d30001t*

Venezuelan President Hugo Chavez, who symbolically resigned last week, on Wednesday was sworn in as president before the National Constitutional Assembly (NCA). The National Electoral Council of Venezuela on Friday officially proclaimed Hugo Chavez Frias President elect for the 1999-2004 period. After preliminary results revealed that Chavez, 44, had defeated his rival Henrique Salas Romer, U.S. ambassador John Maisto met with leaders of the Movement To Socialism, one of the parties making up Chavez's leftist Patriotic Pole.

Figure C.4: Query based summary for the document collection *d170*, for the query "Hugo Chavez"

A nurse who was seriously injured in the 1998 bombing of a Birmingham abortion clinic
is suing fugitive suspect Eric Robert Rudolph, partly in an effort to block any profits he
might receive from a book or movie, her attorney said. Bombing suspect Eric Robert Rudolph
may be intruder who broke into up to 12 mountain homes from July to January to steal food
and toilet paper or sometimes just to get a shower and a shave, a federal agent said today.
That and other evidence convinces federal investigators that Eric Robert Rudolph, suspect in
the Olympic Park and Birmingham abortion clinic bombings, remains in the rugged hills
of Western North Carolina.

Figure C.5: Query based summary for the document collection *d188*, for the query
"Eric Robert Rudolph"

References

Abney, S. (1991). Parsing by chunks. In Robert C. Berwick, Steven P. Abney, and Carol Tenny, editors, *Principle-Based Parsing: Computation and Psycholinguistics*. Kluwer Academic Publishers, pages 257 - 278.

Banerjee, S. and T. Pedersen. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805-810, Acapulco, Mexico.

Barzilay, R. and M. Elhadad. (1997). Using lexical chains for text summarization. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th European Chapter Meeting of the Association for Computational Linguistics, Workshop on Intelligent Scalable Text Summarization*, pages 10-17, Madrid.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing, ACL*, pages 152-155, Trento,IT.

Choi, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics*, pages 26 - 33, Seattle, Washington.

Corston-Oliver, S. H. and W. B. Dolan. (1999). Less is more: eliminating index terms from subordinate clauses. In *Proceedings of the 37th conference on Association for Computational Linguistics*, pages 349–356, College Park, Maryland. Association for Computational Linguistics.

Dorr, B., R. Schwartz, and D. Zajic. (2002). Automatic headline generation for newspaper stories. In *Proceedings of the Document Understanding Conference*, pages 78–85, Philadelphia. NIST.

Dorr, B., R. Schwartz, and D. Zajic. (2003). Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the Document Understanding Conference*, pages 1–8, Edmonton. NIST.

Edmundson, H. P. (1969). New methods in automatic extracting. *J. ACM*, 16(2):264–285.

Galley, M. and K. McKeown. (2003). Improving word sense disambiguation in lexical chaining. In *Proceedings of 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1486–1488, Acapulco, Mexico.

Green, S. J. (1999). Building hypertext links by computing semantic similarity. *IEEE Transactions on Knowledge and Data Engineering*, 11(5):713-730.

Grefenstette, G. (1998). Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *AAAI 98 Spring Symposium on Intelligent Text Summarization*, pages 111-117, Stanford University, Stanford, California.

Halliday, M. and R. Hasan. (1976). *Cohesion in English*. Longman, London.

Halliday, M.A.K. (1978). *Language as Social Semiotic: The Social Interpretation of Language and Meaning*. Baltimore: University Park Press; London: Edward Arnold.

Harabagiu, S. and D. Moldovan, (1998). *WordNet: An Electronic Lexical Database*, chapter Knowledge Processing on an Extended WordNet. MIT press.

Hatzivassiloglou, V., J. L. Klavans, M. L. Holcombe, R. Barzilay, Min-Yen Kan, and K. R. McKeown. (2001). Simfinder: A flexible clustering tool for summarization. In *Workshop on Automatic Summarization, NAACL*, Pittsburg (PA), USA.

Hearst, M. A. (1994). Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 9–16, Las Cruces, New Mexico.

Hearst, M. A. (1997). Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33-64.

Hearst, M. A. (1999). The use of categories and clusters for organizing retrieval results. In Strzalkowski, editor, *Natural Language Information Retrieval*. Kluwer Academic Publishers.

Hirst, G., C. DiMarco, E. Hovy, and K. Parsons. (1997). Authoring and generating health-education documents that are tailored to the needs of the individual patient. In *Proceedings of the Sixth International Conference on User Modeling*, pages 107-118, Sardinia Italy.

Hirst, G. and D. St-Onge. (1997). Lexical chains as representation of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press, pages 305-332.

Hobbs, J. (1978). Coherence and coreference. Technical report, Technical Report Technical note 168, SRI International.

Hoey, M. (1991). *Patterns of Lexis in Text*. Oxford University Press.

Jobbins, A. C. and L. J. Evett. (1998). Text segmentation using reiteration and collocation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 614-618, Montreal.

Jones, K. S. (1993). What might be in a summary? In Knorz, Krause, and Womser-Hacker, editors, *Information Retrieval 93: Von der Modellierung zur Anwendung*, pages 9–26, Konstanz, DE. Universitätsverlag Konstanz.

Kan, M-Y, J. L. Klavans, , and K. R. McKeown. (1998). Linear segmentation and segment significance. In *Proceedings of the 6th International Workshop of Very Large Corpora (WVLC-6)*, pages 197–205, Montreal, Quebec.

Klavans, J. and M-Y Kan. (1998). Role of verbs in document analysis. In *COLING-ACL*, pages 680–686, Montreal, Quebec, Canada.

Knight, K. and D. Marcu. (2000). Statistics-based summarization - step one: Sentence compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 703-710. AAAI Press / The MIT Press.

Kohonen, T. (1989). *Self-Organization and Associative Memory.* Springer-Verlag New York, Inc.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, Toronto, Ontario, Canada. ACM Press.

Lin, C-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, pages 74 - 81, Barcelona, Spain.

Lin, C-Y and E. Hovy. (2002). Manual and automatic evaluations of summaries. In *Workshop on Automatic Summarization, post conference workshop of ACL-2002*, Philadelphia.

Lin, C-Y and Eduard Hovy. (1997). Identifying topics by position. In *Proceedings of the ACL Conference on Applied Natural Language Processing*, pages 283-290, Washington, DC.

Luhn, H. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2).

76

Mani, I. (2001). *Automatic Summarization*. John Benjamins Co, Amsterdam/Philadelphia.

Mani, I. and M. Maybury. (1999). *Advances in Automatic Text Summarization*. MIT Press.

Maybury, M. and A. Merlino. (1997). Multimedia summaries of broadcast news. In M. Maybury, editor, *Multimedia Information Retrieval*.

McKeown, K., R. Barzilay, J. Chen, D. Elson, D. Evans, J. Klavans, A. Nenkova, B. Schiffman, and S. Sigelman. (2003). Columbia's newsblaster: New features and future directions (demo). In *Proceedings of NAACL-HLT'03*, Edmonton, Canada.

McKeown, K. R., Desmond A. Jordan, and Vasileios Hatzivassiloglou. (1998). Generating patient-specific summaries of online literature. In *AAAI 98 Spring Symposium on Intelligent Text Summarization*, pages 34-43, Stanford University.

Mihalcea, R. and D. Moldovan. (2001). extended wordnet: progress report. In *NAACL2001 - Workshop on WordNet and Other Lexical Resources*, pages 95–100, Pittsburgh, PA.

Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. (1993). Five papers on wordnet. CSL Report 43, Cognitive Science Laboratory, Princeton University.

Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. (1990). An on-line lexical database. *International Journal of Lexicography*, 3(4):235-312.

Moldovan, D. and A. Novischi. (2002). Lexical chains for question answering. In *Proceedings of COLING 2002*, pages 674-680, Taipei, Taiwan.

Morris, J. and G. Hirst. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21-48.

Nenkova, A. and R Passonneau. (2004). Evaluating content selection in summarization: the Pyramid method. In *Proceedings of the Human Language Technology Research Conference/North American Chapter of the Association of COmputation Linguistics*, pages 145-152, Boston, MA.

Over, P. (2004). Introduction to duc-2004: an intrinsic evaluation of generic news text summarization systems. In *Proceedings of the Document Understanding Conference*, Boston MA.

Radev, D. R., S. Blair-Goldensohn, Z. Zhang, and R. Sundara Raghavan. (2001). Newsinessence: A system for domain-independent, real-time news clustering and multi-document summarization. In *Demo Presentation, Human Language Technology Conference*, San Diego, CA.

Rambow, O., L. Shrestha, J. Chen, and C. Lauridsen. (2004). Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*, Boston, MA.

Ramshaw, L. and M Marcus. (1995). Text chunking using transformation-based learning. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 82-94, Somerset, New Jersey. Association for Computational Linguistics.

Reynar, J. (1998). *Topic Segmentation: Algorithms and applications*. Ph.D. thesis, Computer and Information Science, University of Pennsylvania.

Reynar, J. C. (1999). Statistical models for topic segmentation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 357 - 364, College Park, Maryland.

Rijsbergen, C. J. Van. (1979). *Information Retrieval*. Butterworth-Heinemann.

Rohall, S. L., D. Gruen, P. Moody, M. Wattenberg, M. Stern, B. Kerr, B. Stachel, K. Dave, R. Armes, and E. Wilcox. 2004. Remail: a reinvented email prototype. In *Extended abstracts of the 2004 conference on Human factors and computing systems*, pages 791–792. ACM Press.

Roussinov, D. G.and Chen, H. (1999). Document clustering for electronic meetings: An experimental comparison of two techniques. *Decision Support Systems [Decis Support Syst]*, 27(1):67–79.

Salton, G., J. Allan, and C. Buckley. (1994). Automatic structuring and retrieval of large text file. *Commun. ACM*, 37(2):97–108.

Silber, H. G. and K. F. McCoy. (2002). Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4):487–496.

Sparck-Jones, K. (1999). Automatic summarizing: Factors and directions. In Mani and Maybury, editors, *Advances in Automatic Text Summarization*. MIT press.

Stairmand, M. A. (1996). *A Computational Analysis of Lexical Cohesion with Applications in Information Retrieval*. Ph.D. thesis, Center for Computational Linguistics, UMIST, Manchester.

Stairmand, M. A. 1997. Textual context analysis for information retrieval. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 140–147, Philadelphia, Pennsylvania, United States. ACM Press.

Stokes, N. (2004). *Applications of Lexical Cohesion Analysis in the Topic Detection and Tracking Domain*. Ph.D. thesis, Department of Computer Science, University College Dublin.

Waibel, A., M. Bett, M. Finke, and R. Stiefelhagen. (1998). Meeting browser:tracking and summarization meetings. In *Proceedings of the 1998 DARPA Broadcast News Workshop*, Lansdowne, Virginia.