AUTOMATED TEXT SUMMARIZATION AND KEYPHRASE EXTRACTION

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Gönenç Ercan

September, 2006

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İlyas Çiçekli (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ferda Nur Alpaslan

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

AUTOMATED TEXT SUMMARIZATION AND KEYPHRASE EXTRACTION

Gönenç Ercan
M.S. in Computer Engineering
Supervisor: Asst. Prof. Dr. İlyas Çiçekli
September, 2006

As the number of electronic documents increase rapidly, the need for faster techniques to asses the relevance of documents emerges. A summary can be considered as a concise representation of the underlying text. To form an ideal summary, a full understanding of the document is essential. For computers, full understanding is difficult, if not impossible. Thus, selecting important sentences from the original text and presenting these sentences as a summary is a common technique in automated text summarization research.

The lexical cohesion structure of the text can be exploited to determine the importance of a sentence/phrase. Lexical chains are useful tools to analyze the lexical cohesion structure in a text. This thesis discusses our research on automated text summarization and keyphrase extraction using lexical chains. We investigate the effect of the use of lexical cohesion features in keyphrase extraction, with a supervised machine learning algorithm. Our summarization algorithm constructs the lexical chains, detects topics roughly from lexical chains, segments the text with respect to the topics and selects the most important sentences. Our experiments show that lexical cohesion based features improve keyphrase extraction. Our summarization algorithm has achieved good results, compared to some other lexical cohesion based algorithms.

Keywords: Automated Text Summarization, Keyphrase Extraction, Lexical Chain, Lexical Cohesion, Natural Language Processing.

ÖZET

OTOMATIK ÖZET VE ANAHTAR KELİME ÇIKARMA

Gönenç Ercan
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. İlyas Çiçekli
Eylül, 2006

Elektronik dokümanların sayısı arttıkça, onların bizim ihtiyaçlarımıza olan yakınlığını ölçebileceğimiz otomatik tekniklere ihtiyaç da artmaktadır. Özetler, dokümanın kısa ve öz bir sunumu olarak kabul edilebilir. İdeal bir özet için, dokümanın tamamıyla anlaşılması çok önemlidir. Ancak, bilgisayarların otomatik olarak dokümanı anlaması imkansız değil ise bile çok zordur. Bunun için, dokümandan önemli kelime veya cümleleri seçmek ve bunları özet olarak sunmak, otomatik özet çıkarma araştırmalarında sık kullanılan bir yöntemdir.

Dokümandaki kelime bütünlüğü önemli kelime veya cümleleri belirlemekte kullanılabilir. Kelime zincirleri, kelime bütünlüğünü analiz etmekte kullanılabilecek bir araçtır. Bu tezde kelime zincirleri kullanarak, otomatik özet ve anahtar kelime çıkarma çalışmalarımız anlatılıyor. Bu tezde kelime bütünlüğünün, anahtar kelime bulmadaki etkileri bir öğreticiyle öğrenme programı aracılığıyla araştırılıyor. Özet çıkarma sistemimiz bir dokümanın kelime zincirlerini çıkarıp, konuları kelime zincirlerinden kabaca bulup, yazıyı konuya göre parçalara bölüp, en önemli parçalardan cümle seçiyor. Anahtar kelime bulma deneylerimizde, kelime bütünlüğünün anahtar kelime bulmanın başarısını arttırdığı görülmüştür. Özet çıkarma sistemimiz diğer kelime bütünlüğü kullanan özet sistemleriyle karşılaştırılınca, iyi sonuçlar almıştır.

Anahtar sözcükler: Otomatik Özet Çıkarma, Anahtar Kelime Çıkarma, Kelime Zincirleri, Kelime Bütünlüğü, Doğal Dil İşleme.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Summarization

Summary is the condensed representation of a document's contents. A summary outlines important aspects of the document in a precise way. It should be informative, providing the most important information in the document. A summary should be non-repetitive and as brief as possible. In a text, the same information can be repeated to emphasize its importance, but a summary should give as much precise information as possible. A summary should be indicative, it should indicate the document's relevance to the reader.

Humans write a document to represent an idea, event or an opinion. Text evolves around a general concept, which is coherently partitioned into sub-topics, that supports the main topic. The summary should capture the general idea and should include important topics.

Eduard Hovy [25] formally defines summary as:

> *a summary is a text that is produced from one or more texts, that contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s).*

This definition could be relaxed, so that the target summary could be in any format. While summaries usually are structured informative or indicative short texts, they could also be diagrams or outlines sketching the topics discussed. Even a set of keyphrases could be considered as a summary, as long as it is providing significant information.

Abstracts are special forms of summaries that are generated/paraphrased texts and are formed from the most important topics in the text. A summary formed of sentences extracted from the original text is called an **extract**. In an extract, sentences extracted from the text should be the most important and representative sentences. In the case of keyphrases, extraction of the most important and representative phrases is called **keyphrase extraction** which constrains the output to phrases that appear in the document. If the target keyphrases may contain phrases that do not appear in the original text, it is called **keyphrase generation**.

Content to be summarized can be any data representable in text. Long documents such as journals, novels and books or short documents such as emails, news articles and dialogue scripts are some examples of attacked text genres. Each genre has different structures and its own difficulties. Summarizing longer documents with more topics and weaker co-relations is a harder problem, because it is difficult to determine the main topic in the document. In a short document, repetition of terms is lower and interpreting the document requires more prior knowledge.

The way we share, store, write and publish text data has been revolutionized with the advances in networks, Internet, PC hardware and software tools, which makes it easier to create content. When you have such a vast material at your hands it gets harder to search and find data relevant to you. While search engines can be used for finding data, they don't analyze the semantic structure of the text or understand the text. Matches for queries depend mostly on word repetitions. Thus, you may end up trying to scan large number of documents. Summaries aid users to evaluate the relevance of a document without reading the full text. Since a summary is not cluttered with detail, a user can quickly recognize its relevance.

Summaries could be displayed in search results as an informative tool for the user. For example Google search engine provides summaries in search results. Digital libraries and journals could make use of summaries. Users of the library or readers of the journal could benefit from summaries and find relevant text easily. News portals could provide precise summaries about a news merged from multiple source articles. Columbia University has a system called Newsblaster [30] for this purpose. Web browsing with web site summaries could change our browsing habits and enable us to filter irrelevant web pages.

While most obvious uses of summaries are focused on using them as tools for users, summaries could be used by search engines. Search engines could index summaries instead of the whole document, lowering the resources needed by indexing algorithms. A summary capable of highlighting the most important aspects of the text could improve the performance of search engines, in terms of the relevance of the results.

Unfortunately, most of the data available today does not have summaries. Even for a human, summarizing a text is an exhaustive and difficult work. For this reason, automating the task of creating indicative and informative summaries has been issued by many researchers. **Automated text summarization** is the process of automatically constructing summaries for a text. Systems summarizing a single document are called **single document summarization** systems. Systems summarizing a set of documents to form a single summary are called **multi document summarization** systems. Single document summarization is a difficult task by itself, but multi document summarization has additional difficulties. **Query relevant summarization** systems provide a summary for document(s) based on a query or a question. Query relevant summarization is very similar to question answering. The generated summary is shaped by the user's interest.

An **extract** is a summary formed of sentences taken from the document(s). Forming extracts, involves identification of the important units in text. A system that targets to output extracts is named as **extractive summarization system**. The system should decide what is important and what is not. Text generation is not needed for extracts, since it is formed from text taken from the original text(s). However, extracted content should be presented in a coherent way. After extraction, the formed summary could suffer from dangling references and weak information ordering. An extractive summarization system could benefit from correcting these problems to produce higher quality summaries. Some systems reduce the sentences to form even shorter summaries, this is called **sentence reduction** [28]

An abstract is a summary, that summarizes document(s) using its own words. Abstracts are harder to form. A system targeting abstracts must understand the text, identify important data and fuse this data in a cohesive and grammatically correct manner. Ideally, abstracting a text involves interpreting the text fully. Thus, a human or a computer needs to interpret and understand the text to be summarized. Only constrained, or domain specific solutions are present for forming abstracts. Humans interpret the text using their prior knowledge about the domain, this is difficult if not impossible for a computer. Forming abstracts remains as an important and unresolved challenge.

Extractive summarization systems are usually formed of two phases. The first phase deals with important content selection, and the second phase deals with the presentation of the selected contents.

Important content identification is the first and the most important part of extractive summarization systems. Current techniques for importance identification usually depend on more surface level features. Features like cue phrases, position in text, taking advantage of formatting features like headers and bold text, frequency and more sophisticated features like cohesion and coherence are used in current summarization systems. In most of the techniques, the motivation is to identify topics and evaluate the importance of these topics.

Although the main focus on summarization is on selection procedures, correcting and presenting the extracted data in a more cohesive manner is issued by some researchers. Extractive summarization systems do not have text generation, but the quality of extracts could be improved. Sentence reduction, anaphora resolution, information ordering [8] and reducing repetition could improve summaries. Mani [36] presents a summary revision system. Knight and Marcu [31] outlines a text compression algorithm which uses discourse structure. Paraphrasing sentences of the extract is an impressing technique. Barzilay et al.[7] describes a system, that paraphrases sentences.

Cohesion and coherence are two natural phenomena, seen in sensible texts. Coherence is the semantic structure of the text that gives the feeling that the text is interpretable. The coherence structure is hard to model. Modeling the coherence structure requires prior knowledge and requires some level of understanding. Marcu [38, 40, 39] presents a good summarization system which takes advantage of coherence. Marcu uses discourse structure and more specifically rhetorical parsing to model coherence. His model depends on cue phrases called discourse markers. Research on coherence based summarization systems, is challenged by the difficiulties in modelling coherence.

Cohesion, especially lexical cohesion, is a simpler and more surface level feature which can be modeled computationally. **Cohesion** is defined as sticking together. In text, text units stick to each other with relations. Relations between word meanings in a text form **lexical cohesion**, which is a type of cohesion. **Lexical chains** are computational models for lexical cohesion.

In this thesis, we present an **extractive summarization system** attacking **single document summarization**, **multi document summarization** and **keyphrase extraction** problems. Our focus is on important sentence and keyphrase identification. Our system takes advantage of **lexical cohesion** and **cohesion**. Using **lexical chains**, topics and segments in the text are identified. Our keyphrase extraction algorithm tries to improve existing keyphrase extraction systems (Turney [56] and Witten et al. [58]) by integrating lexical cohesion based features to extract keyphrases.

Extracting sentences and keyphrases are similar problems. In both keyphrase extraction and summarization, all the phrases or sentences are evaluated by their importance. Thus, usually similar features are exploited for these problems. Our system takes advantage of lexical chains.

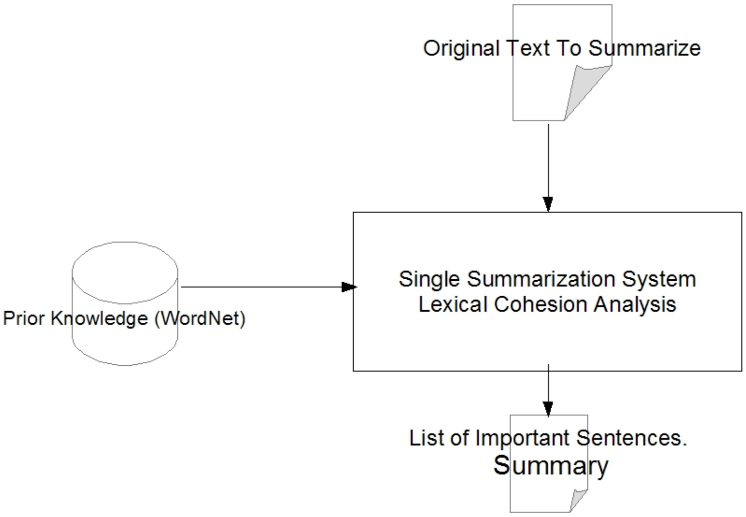

Original Text To Summarize
Single Summarization System
Lexical Cohesion Analysis
Prior Knowledge (WordNet)
List of Important Sentences.
Summary

Figure 1.1: Single Document Summarization System

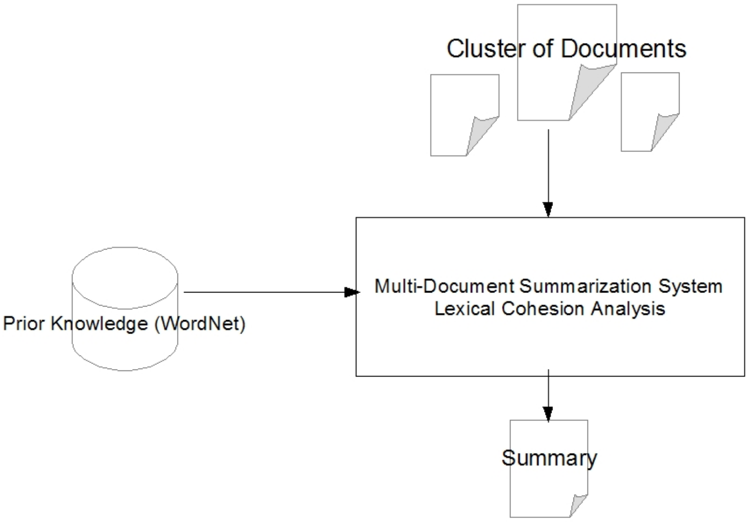

Cluster of Documents
Multi-Document Summarization System
Lexical Cohesion Analysis
Prior Knowledge (WordNet)
Summary

Figure 1.2: Multi Document Sumarization System

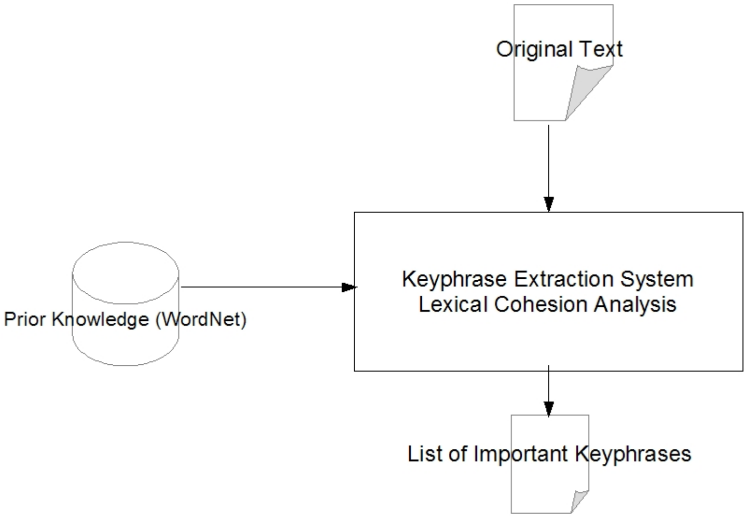

Original Text
Keyphrase Extraction System
Lexical Cohesion Analysis
Prior Knowledge (WordNet)
List of Important Keyphrases

Figure 1.3: Keyphrase Extraction

1.2 Thesis Goals

This thesis addresses three major problems:

1. Selecting n sentences from a single document to form up a summary.
2. Selecting n sentences from multiple documents, somewhat about the same topic, to form up a summary.
3. Selecting n keyphrases for a single document.

For these goals, this thesis focuses on lexical cohesion and tries to model the document's cohesion structure roughly with lexical chains. We have investigated the role of the cohesion structure in texts and how it can be exploited to identify topics and segments in the text. The contribution of this thesis focuses on

Cambodian leader Hun Sen on Friday rejected opposition parties’ demands for talks outside the country, accusing them of trying to ‘‘internationalize’’ the political crisis. Government and opposition parties have asked King Norodom Sihanouk to host a summit meeting after a series of post-election negotiations between the two opposition groups and Hun Sen’s party to form a new government failed. Opposition leaders Prince Norodom Ranariddh and Sam Rainsy, citing Hun Sen’s threats to arrest opposition figures after two alleged attempts on his life, said they could not negotiate freely in Cambodia and called for talks at Sihanouk’s residence in Beijing. Hun Sen, however, rejected that. ‘‘I would like to make it clear that all meetings related to Cambodian affairs must be conducted in the Kingdom of Cambodia,’’ Hun Sen told reporters after a Cabinet meeting on Friday. ‘‘No-one should internationalize Cambodian affairs. It is detrimental to the sovereignty of Cambodia,’’ he said. Hun Sen’s Cambodian People’s Party won 64 of the 122 parliamentary seats in July’s elections, short of the two-thirds majority needed to form a government on its own. Ranariddh and Sam Rainsy have charged that Hun Sen’s victory in the elections was achieved through widespread fraud. They have demanded a thorough investigation into their election complaints as a precondition for their cooperation in getting the national assembly moving and a new government formed. Hun Sen said on Friday that the opposition concerns over their safety in the country was ‘‘just an excuse for them to stay abroad.’’ Both Ranariddh and Sam Rainsy have been outside the country since parliament was ceremonially opened on Sep. 24. Sam Rainsy and a number of opposition figures have been under court investigation for a grenade attack on Hun Sen’s Phnom Penh residence on Sep. 7. Hun Sen was not home at the time of the attack, which was followed by a police crackdown on demonstrators contesting Hun Sen’s election victory. The Sam Rainsy Party, in a statement released Friday, accused Hun Sen of being ‘‘unwilling to make any compromise’’ on negotiations to break the deadlock. ’A meeting outside Cambodia, as suggested by the opposition, could place all parties on more equal footing,’ said the statement. ’But the ruling party refuses to negotiate unless it is able to threaten its negotiating partners with arrest or worse power.’

Figure 1.4: An Example News Article

the sentence/phrase scoring procedure. For keyphrase extraction, we have experimented with a supervised machine learning algorithm. Different features are investigated that can be derived from lexical chains.

For summarization, we treated lexical chains as contributors of topics. We tried to identify the relations between lexical chains from the current context. Using these topics, which are actually sets of lexical chains, we tried to segment the text from the perspective of each topic. With this approach, we identified topic shifts and topic concentration points.

The general system architecture for these problems are sketched in Figure 1.1 and Figure 1.2 for summarization and Figure 1.3 for keyphrase extraction. Our system uses WordNet [42, 20] as prior knowledge. Meaning of words and relations between words are acquired from WordNet.

Figure 1.4 shows an example news article and Figure 1.5 shows the summary generated by our system for this article.

Cambodian leader Hun Sen on Friday rejected opposition parties ' demands for talks outside the country , accusing them of trying to " internationalize " the political crisis . Hun Sen said on Friday that the opposition concerns over their safety in the country was " just an excuse for them to stay abroad . " Hun Sen 's Cambodian People 's Party won 64 of the 122 parliamentary seats in July 's elections , short of the two-thirds majority needed to form a government on its own . Sam Rainsy and a number of opposition figures have been under court investigation for a grenade attack on Hun Sen 's Phnom Penh residence on Sep. 7 .

Figure 1.5: Automatically Extracted Summary of the Text in Figure 1.4

1.3 Thesis Outline

In Chapter 2, necessary background information and related work in summarization research is outlined. As a background information, we believe that a solid understanding of the terms coherence and cohesion is necessary, as our algorithm depends heavily on these concepts. This chapter explains these terms from a computational linguists perspective. WordNet and lexical cohesion are also described in this chapter. Previous research for text summarization and keyphrase extraction are also briefly introduced.

Lexical chains are described in Chapter 3. Lexical chains are the key elements of our research, for this reason we tried to explain lexical chaining algorithms and properties of lexical chains in more detail. Lexical chaining algorithms and challenges in lexical chaining are described.

Chapter 4 describes our algorithm for single document summarization. A unique approach, trying to integrate different cohesion clues is explained in detail. This chapter presents our experiments for summarization and a comparison with existing algorithms.

Chapter 5 defines our multiple document summarization algorithm derived from the single summarization algorithm. This chapter presents our experiments for summarization and comparison with existing algorithms.

Our supervised keyphrase extraction algorithm and the features based on lexical chains are given in detail in Chapter 6. This chapter discusses our solution and compares our results to other algorithms results.

This thesis addresses three similar problems. In our solutions for these problems, there are common components. The implementation details of our components like noun phrase detector, sentence detector and part of speech are given in Chapter 7. Components like noun phrase detection, sentence detection, part of speech tagging are explained in this chapter.

As a conclusion, in Chapter 8, overall performance of our algorithms, and lexical cohesion based techniques are discussed. Also the evaluation of our algorithms, possible improvements and possible applications for the work on this thesis are discussed.

Chapter 2

Background and Related Work

2.1 Linguistic Background

In any sensible text, document is not just a bag of sentences, above grammatical structure, text has a semantic structure. Humans write to present an idea, event, concept or an opinion. Thus, it is more than natural that our document evolves around a general concept. In a good presentation, the main idea to be presented is divided into sub-ideas and concepts. These ideas should be structured and related with each other semantically, so that they can form up the big picture. Topics should drift in a proper way, so that the reader can follow the general idea easily.

2.1.1 Coherence

In linguistics, coherence is used to define the semantic integrity of a document. Coherence is essential in well composed documents, which can be thought as a hidden element which provides the feeling that a document is written intelligently. We can think of coherence as the semantic structure of the text. Modelling coherence requires interpretation of the text. Although there are some patterns for writing a coherent document, it is not possible to define strict rules.

- *[John is living in a neighborhood with a very high crime rate.*$_1$*] [His house was robbed 4 times last year.*$_2$*]*
- *[John is living in a neighborhood with a very high crime rate.*$_1$*] [He likes spinach.*$_2$*]*
- *[John is living in a neighborhood with a very high crime rate.*$_1$*] [I bought a movie about a murderer.*$_2$*]*

Figure 2.1: Example of Coherence

Coherence relationships could be exploited to form a model of the text. Common relationships are elaboration, cause, support, exemplification, contrast and result. Explaining each relation is not in the scope of this thesis. Relationship classification for sentences or clauses is a very hard process. Usually efforts on coherence analysis result in trees where the nodes are the text segments (paragraphs, sentences, phrases) connected by these relationships.

Discourse structure and rhetorical parsing is a good example for representing the coherence structure in text. Marcu [41, 38] presents an effective summarization system, which uses shallow models of coherence. Marcu takes advantage of cue phrases and calls these phrases as discourse markers. Local discourse structure forms a tree like model, which forms the global discourse structure of the text. Coherence structure is a hard feature to deal with as it requires more knowledge than the information that could be acquired from the text.

In Figure 2.1, the first example is coherent, while the others are not. If we take a closer look at the coherence structure of the first example, the second sentence supports the first sentence. These relations can only be determined by the text's meaning as a whole. The second example is not coherent as there is no link between the two sentences, but if; '*Spinach is easy to find in that neighborhood*' is a known fact by the reader then it would be coherent. Readers use their prior knowledge on the domain to interpret the coherence structure. Third example is not coherent also, although '*murder*' occurring in the following sentence is related to *crime*.

2.1.2 Cohesion and Lexical Cohesion

Cohesion is simpler than coherence and it can also help to determine the discourse structure in a text. Cohesion is a more surface level feature. Coherence usually deals with the whole semantic structure of the text, while cohesion deals with the relationships between peer text units. In a meaningful document, cohesion usually forms up a chain of co-related units. In contrast to coherence, cohesion only tries to determine if a text unit is related with another unit in the same text or not.

Halliday and Hasan [22] defines five types of cohesion relationships:

- **Conjunction** - Usage of conjunctive structures like '*and*' to present two facts in a cohesive manner. In the sentence *'I have a cat and his name is Felix'*, two facts are connected with the conjunctive '*and*'.
- **Reference** - Usage of pronouns for entities. In the example *'Dr. Kenny lives in London.* ***He*** *is a doctor.'*, the pronoun '*he*' in the second sentence refers to '*Dr.Kenny*' in the first sentence.
- **Lexical Cohesion**- Usage of related words. In the example sentence '***Prince*** *is the next* ***leader*** *of the kingdom.*', '*leader*' is a more general word for '*prince*'.
- **Substitution** - Using an indefinite article for a noun. In the example '*As soon as John was given a* ***vanilla ice cream cone****, Mary wanted* ***one*** *too.*', the word '*one*' refers to the phrase '*vanilla ice cream cone*'.
- **Ellipsis** - Implying noun without repeating. In the sentence '*Do you have a* ***pencil****? No I don't*', the word '*pencil*' is implied without repeating in the second sentence.

From these cohesion structures, lexical cohesion is the most definite and easiest to find. Thus, like most of the research on cohesion, we focus on lexical cohesion, and use it in our summarization system. Our system uses other cohesion types

It is easy to see that the **dog**’s *family tree* has it’s *roots* from **wolves**. In fact, their connection is so close and recent, the position of **wolves** on the *tree* would be located somewhere on the *branches*. Any breed of **dog** can have fertile offspring with a **wolf** as a mate. The only physical trait found on a **wolf** that is not found on a domesticated **dog** is a scent gland located on the outside base of a **wolf**’s **tail**. Every physical trait on a **dog** can be found on a **wolf**. **Wolves** might not have the **coat** pattern of a **Dalmatian**, but there are **wolves** with black **fur** and there are **wolves** with white **fur**.

Figure 2.2: A Lexical Cohesion Example

with loose assumptions and techniques. Cohesion is based on the relationships between units of the document, in the case of lexical cohesion these units are words and phrases. Phrases in a document should be semantically related and this is called lexical cohesion.

If we reconsider the example in Figure 2.1, the first example, which is coherent, has some lexical cohesion through the words ’*crime*’ and ’*robbed*’. These two words are sticking to the concept of crime. In the third example, we still have lexical cohesion through the words ’*crime*’ and ’*murder*’, but these two sentences are not coherent. This example shows that since lexical cohesion is a more surface level indicator, it may not reflect the coherence structure correctly.

Figure 2.2 shows an extended sample of lexical cohesion with two chains. Two sets of words, {`wolf, dog, Dalmatian, fur, coat, tail`} and {`family tree, tree, branches, roots`} can be considered as the lexical cohesion structure of this text.

Forming the lexical cohesion depends on determining the semantic relationships between words. These semantic relations are known by humans and can be quickly recognized, as long as the vocabulary used is familiar to them. A lexicon is a structured knowledge base storing semantic information about words.

2.1.3 Lexicon

Words can be categorized into two: open-class words and closed-class words. Open-class words are namely nouns, verbs, adverbs and adjectives. Closed-class words are pronouns, articles and prepositions. Open-class words contribute more information to the meaning of a text. Closed-class words contribute more to the grammatical structure.

An open-class word has some attributes like grammatic properties, meaning, relation with other words, spelling and sound. This information for a word, is called as **lexical entry**. Library of lexical entries is called as **lexicon**. In a lexicon, all meanings of a word are stored. A meaning of a word is called as a **sense** of the word. Senses are related with each other through semantic relationships, forming a huge semantic network.

A symbolic representation of a word can have many different meanings. For example, Figure 2.3 shows 4 senses of the word ' *bank*' out of 10 senses defined in WordNet, which is a lexicon. When the word *bank* occurs in a text we may not know which sense it occurs as, but only one of the senses is intended in that sentence.

If two senses of the same word are semantically unrelated, these two senses are called **homonyms**. If they are related, these two senses are called **polysemous**. The sense '$bank_3$' is related with '$bank_1$'. In '*Blood bank*', '*sperm bank*', the word '*bank*' is '$bank_3$'. This sense is related to '$bank_1$'. In fact, '$bank_3$' is actually derived from '$bank_1$', for this reason they are considered as polysemous. However, '$bank_1$' and '$bank_2$' are homonyms, since their meanings are unrelated. Polysemous senses are a challenge in determining the correct sense of a word, as their definition may overlap with each other.

A sense can have the same meaning with a different symbolic representation. For example, words '*plant*' and '$flora$' map to the same sense. This is called **synonymy**. Opposite meaning of a sense is called **antonym**. For example '*good*' and '*evil*' are antonyms of each other.

In a text, nouns contribute to the meaning of the text more than the other word groups. Although verbs and adjectives also contribute to the meaning of the text, they do not influence the meaning as much as nouns do. Adjectives are not key contributors of lexical cohesion. For example, in the sentence '*I bought a red pencil*', the word '*red*' only specifies an attribute of the noun '*pencil*'. Verbs however does provide more information, but semantic relations of verbs are not

1. *a financial institution that accepts deposits and channels the money into lending activities; 'he cashed a check at the bank'; 'that bank holds the mortgage on my home'*
2. *sloping land (especially the slope beside a body of water); 'they pulled the canoe up on the bank'; 'he sat on the bank of the river and watched the currents'*
3. *a supply or stock held in reserve for future use (especially in emergencies)*
4. *a building in which commercial banking is transacted; 'the bank is on the corner of Nassau and Witherspoon'*

Figure 2.3: Senses of the Word *'bank'*

suitable for lexical cohesion analysis. First problem with verbs is, most verbs entail each other. Number of polysemous verbs is very high. Fewer nouns are polysemous. Thus, most of the research on lexical cohesion analysis has been focused on nouns.

Nouns can be structured in a specialization/generalization hierarchy. For example, '*car*' is a specialization of '*vehicle*'. Specializations and generalizations are symmetric so '*vehicle*' is a generalization of '*car*'. A sense can be a part of another sense. For example '*Belgium*' is a member of '*European Union*'. These are called part/whole and whole/part relationships. These relations are symmetric as '*European Union*' has a member called '*Belgium*'.

Siblings in the generalization/specialization hierarchy are called as **coordinate terms**. For example '*wolf*' and '*dog*' are two coordinate terms, as they are both '*canines*'.

All the relationships defined in this section are considered as **classical relationships** between senses. They are easy to classify and identify, as a clear and structured definition exists for them. However, there are relations between senses that are hard to define. For example '*cop*' and '*donut*' are related. These kind of relations can be described as senses in the same domain. These words will tend to co-occur in many text. These kind of relations are usually identified statistically and there is no general rule for defining these kinds of relations, as they do not fit into any of the classical relationships.

2.1.4 WordNet as a Thesaurus

In previous sections, we have defined lexical cohesion and relationships between words that form the lexical cohesion. For a computer to access this information, a thesaurus has to be used. WordNet, Roget's Thesaurus [2] and some more have been designed for this purpose. WordNet has a higher coverage then all other Thesaurus. WordNet is a semantic network formed from synsets, that are connected through semantic relations. Synsets are the senses in WordNet, we will prefer to use the word sense in following discussions. Synonymy is considered as the strongest semantic relation. Two words are synonyms if they map to the same synset in WordNet. Meaning of each sense is called glosses.

The relation from a sense to a more specialized sense is called **hyponym** and to a more general sense is called **hypernym**. WordNet's hypernym/hyponym hierarchy starts of from some root nodes and is organized in levels of abstraction. Root nodes are the most general and abstract terms. For example '*entity*' is the most dominant root node for nouns in WordNet. The longest path from a node to a root node is 16 in the hierarchy. Length of paths in hypernym/hyponym hierarchies is a somewhat questionable metric as hierarchies can have varying depths.

Whole/part relation in WordNet is named as **Meronymy/Holonymy**. **Holonymy** stands for the whole, ie. '*tree*' has part '*branch*', '*tree*' is a holonym of '*branch*'. **Meronymy** is the part, ie. '*branch*' is a part of '*tree*', '*branch*' is meronym of '*tree*'. Meronyms/holonyms in WordNet can be classified into sub groups. '**Has/Is part**', '**is (made from) substance**' and '**has/is member**'. Meronyms/holonyms are inherited from hypernyms of a sense. While a '*dog*' has '*flag*' as a part of '*dog*', '*hair*' is inherited from '*mammal*' which is a hypernym of '*dog*'.

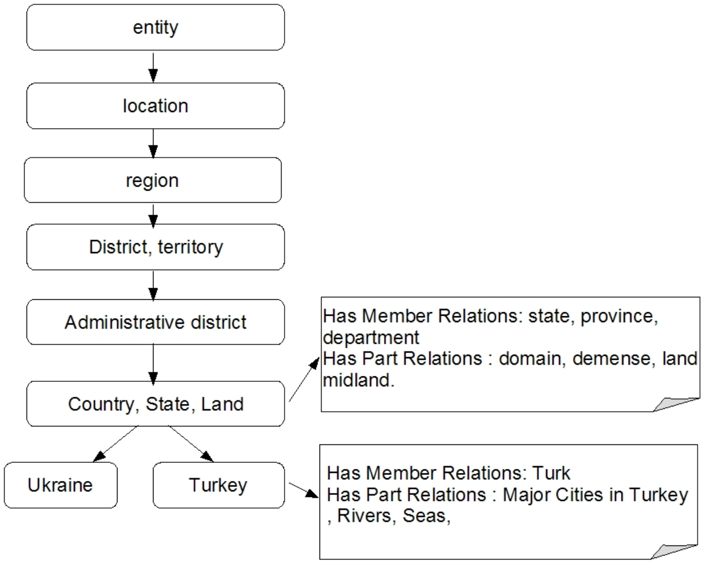

entity
location
region
District, territory
Administrative district
Country, State, Land
Ukraine
Turkey
Has Member Relations: state, province, department
Has Part Relations : domain, demense, land midland.
Has Member Relations: Turk
Has Part Relations : Major Cities in Turkey , Rivers, Seas,

Figure 2.4: Wordnet Hierarchy Example For the word Turkey, Republic of Turkey

Figure 2.4 shows full hypernym/hyponym hierarchy for the word '*Turkey*'. The word '*Turkey*' is a leaf node and it is connected to the top level word '*entity*' by 5 intermediate nodes. Some meronyms of '*Turkey*' are shown in the figure. While each node can have meronyms of its own, it can also inherit the meronyms of its ancestor. In this case, '*Turkey*' has inherited meronyms like '*department*' from its ancestor '*country*'. Also the word '*Ukraine*' is shown in the graph. '*Ukraine*' is a hyponym of '*country*'. '*Turkey*' and '*Ukraine*' are siblings in the hypernym/hyponym hierarchy, so they are **coordinate terms**. Note that '*Ukraine*' also inherits the meronyms of '*country*'.

We have talked about non-classical relationships in the previous section. There are some efforts to identify these relations through combinations of classical relationships between two senses in WordNet. Similarity between two senses is measured by paths connecting these senses. WordNet is connected, that is any two sense has a path connecting them. Budanitsky and Hirst [12] evaluates five different measures for similarity between two senses. Some of these techniques merge the data in WordNet with co-occurrence statistics obtained from corpora.

A sense can contain another sense's word in its gloss, but there might be no classical relation between these two senses. Extended WordNet [23] project tries to increase the connectivity between senses using the information in glosses. In their word, glosses are parsed and all the words are disambiguated to find the correct sense and semantic relations are formed. This is another technique to determine non-classical relations.

Currently English WordNet's coverage is very high and WordNet is the most complete thesaurus. WordNet for other languages is under development. Global WordNet Association is organizing these efforts to provide WordNets for different languages. EuroWordNet [1] project is organizing the efforts on European languages. BalkaNet is a branch of EuroWordNet, researching to form a database for Balkan languages. Construction of Turkish WordNet is still under process organized by Sabanci University [44].

WordNet's completeness and the efforts on other language versions of WordNet, encouraged us to use WordNet as a thesaurus.

2.2 Related Work in Summarization

Summarization has been an active research area since 1950's. Summarization task could be thought as a two level process, content selection/importance identification and text generation/smoothing extracts. We will present the previous work for these two phases separately.

2.2.1 Content Selection and Importance Identification

A summarization system tries to identify significant information that is important enough to be in the summary. The way we write documents, how we form the content model, and how we emphasize certain content is a phenomenon. There are no strict rules, but there are clues that could be exploited to identify important topics and ideas. Summarization research investigates different clues. It is not possible to claim that any of the features that are used in summarization yields the best results for all text genre.

2.2.1.1 Methods Using Position in Text

Document creators tend to follow some patterns on positioning the important content. Although this depends on the genre and domain of the text, a general belief is that important content is usually positioned in the first sentences. In fact, a very simple and surprisingly successful method for summarization is selection of the first sentences in text. Brandow, Mitze and Rau [10] has achieved very good results in news articles, by selecting the first sentences as summaries. Edmundson [17], Kupiec, Pederson and Chen [32], Teufel and Moens [54] all experimented with similar algorithms. They report that this simple technique gives the best results in news articles and scientific reports. As a matter of fact in Document Understanding Conference 2004 [4], baseline algorithm simply extracts the first sentences, and has been one of the best scoring algorithms when the target summary is limited to 75 characters.

Lin and Hovy [33] presents an extensive research for deriving the optimum position policy for different domains. They report that different text genres have different focus positions.

2.2.1.2 Methods Using Cue-Phrases and Formatting

Some phrases are used to emphasize their importance in text, and these phrases are called **bonus phrases**. Some clue phrases reflect that the sentence is not important, and these phrases are called **stigma phrases**. *'significantly'*, *'in conclusion'* and *'last but not the least'* are few examples of bonus phrases while *'hardly'* and *'impossible'* are examples of stigma phrases. Teufel et al.[54] uses cue phrases on science articles while Kupiec et al.[32] and Edmundson[17] uses cue-phrases to improve existing summarization systems. Exploiting the formatting features like bold words, headers could also improve the summarization performance. Edmundson[17] and Teufel et al.[54] have shown that simple heuristics taking advantage of format features improves the success of the summarizer. Overlap between the sentences and the titles, bold phrases could be used as a clue for importance.

[With its distant orbit $_1$] [50 percent farther from the sun than Earth $_2$] [and slim atmospheric blanket, $_3$] [Mars experiences frigid weather conditions. $_4$] [Surface temperatures typically average about -60 degrees Celsius (-76 degrees Fahrenheit) at the equator$_5$] [and can dip to -123 degrees C near the poles. $_6$]

Figure 2.5: Text Fragment to Demonstrate Coherence Based Techniques

2.2.1.3 Methods Using Word Frequency

Luhn [35] claims that important sentences contain unusually frequent words in the text. This has not been proven in any research. In fact, word frequency decreased the performance of some summarization systems. Edmundson [17] and Kupiec et al.'s [32] experiments indicate that integrating word frequency to their summarization systems decreased the accuracy of the summarization system. However, word repetition by itself is a lexical cohesion type and there are lexical cohesion based summarization systems that reported successful results. Using word frequency by itself is not proven to be a powerful clue. Some systems takes advantage of word repetitions with information retrieval techniques, but the theory behind these algorithms is more sophisticated and we preferred to classify them as lexical cohesion based summarization systems.

2.2.1.4 Methods Using Coherence

Much of the research on Coherence based summarization is focused on Rhetorical theory. Marcu's method [41, 39] is an example of coherence based summarization. Marcu uses rhetorical parsing to model the discourse structure in the text. He models the discourse structure of the text using a tree like structure. From local structures to whole text, all relationships between clauses are determined. Forming this tree like structure takes advantage of cue phrases. Figure 2.5 shows an example from Marcu's Phd. Thesis [39] and Figure 2.6 shows the discourse structure for this text.

From the discourse structure Marcu derived a scoring function for each unit depending on relation types and depth of the tree below each node. Marcu's work has achieved good results and is considered as one of the best summarization algorithms available. However, building discourse trees is a difficult problem.

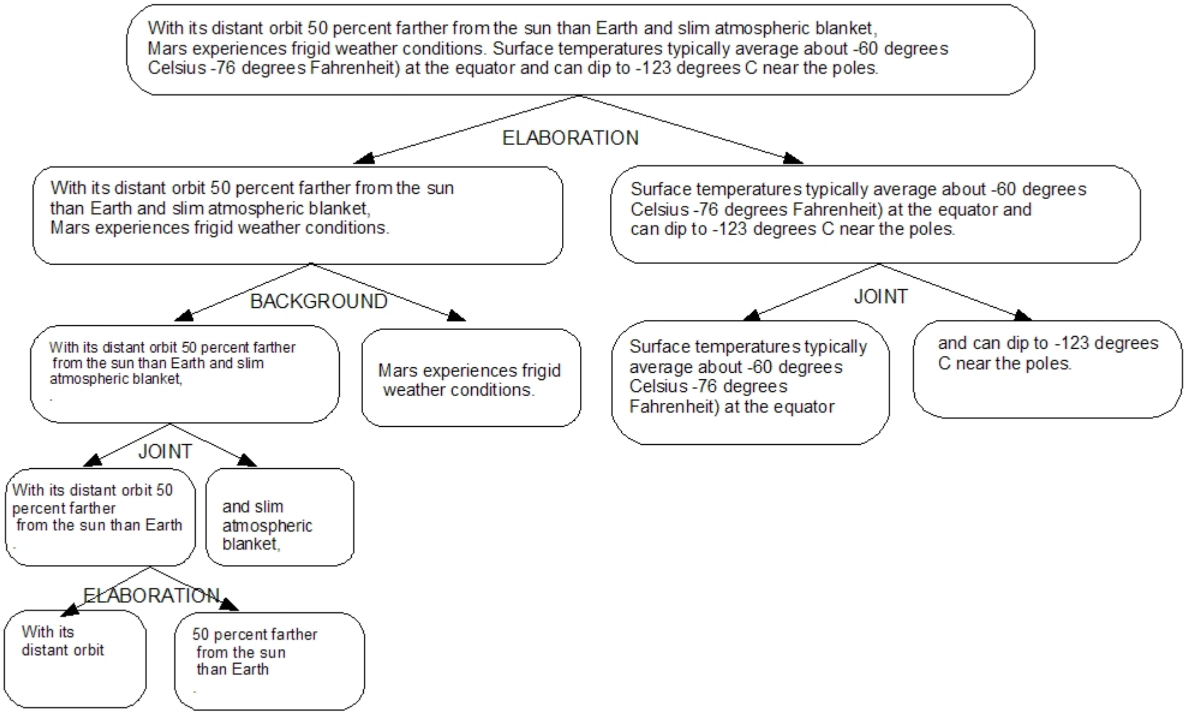

With its distant orbit 50 percent farther from the sun than Earth and slim atmospheric blanket, Mars experiences frigid weather conditions. Surface temperatures typically average about -60 degrees Celsius -76 degrees Fahrenheit) at the equator and can dip to -123 degrees C near the poles.
ELABORATION
With its distant orbit 50 percent farther from the sun than Earth and slim atmospheric blanket, Mars experiences frigid weather conditions.
Surface temperatures typically average about -60 degrees Celsius -76 degrees Fahrenheit) at the equator and can dip to -123 degrees C near the poles.
BACKGROUND
JOINT
With its distant orbit 50 percent farther from the sun than Earth and slim atmospheric blanket,
Mars experiences frigid weather conditions.
Surface temperatures typically average about -60 degrees Celsius -76 degrees Fahrenheit) at the equator
and can dip to -123 degrees C near the poles.
JOINT
With its distant orbit 50 percent farther from the sun than Earth
and slim atmospheric blanket,
ELABORATION
With its distant orbit
50 percent farther from the sun than Earth

Figure 2.6: Example Discourse Structure for the Text in Figure 2.5

Performance of building the discourse tree structure is questionable. This method is blocked by the difficulties in modeling the coherence structure.

2.2.1.5 Methods Using Lexical Cohesion

Radev et al. [48] attacks automated summarization problem using information retrieval techniques. Radev et al. uses vector space model and clustering to find the central and salient sentences. They are using weighted vectors of TFx-IDF values to represent sentences. **TF** is **term frequency** and IDF is **inverse document frequency**. IDF is the frequency of the word in all documents in the corpus. Note that this approach depends on word frequencies, so they are only taking advantage of word repetition. Word repetition is one of the lexical cohesion types. Erkan [18] improved the performance of the summarizer by introducing a Google's Pagerank [45] like algorithm for the selection procedure. This summarization system is a part of MEAD summarization toolkit [47] and is an important algorithm in automated summarization research literature.

Lexical chains are structures for modeling lexical cohesion computationally. Lexical chains are sets of related words. Halliday [22] presents one of the first work on lexical cohesion. Morris and Hirst [43] discusses an algorithm for building lexical chains. St.Onge et al. presents [53] the first algorithm where, lexical chains are built using WordNet. They used lexical chains to detect and correct malapropisms[1]. Barzilay [6] presented her lexical chaining algorithm and used lexical chains to extract summaries. Barzilay's algorithm has achieved good results in evaluations. Usually, in algorithms using lexical chains, text units that are traversed by the strongest lexical chains are selected. Following Barzilay's algorithm there have been many lexical cohesion based summarization techniques. Silber and McCoy [52] presented an efficient summarizer based on lexical chains. Their algorithm is focused on improving the running time of lexical chaining algorithm. Brunn et al. [11, 13] proposed a different sentence selection procedure using lexical chains. Doran et al. [16, 57] experimented with different scoring functions for detecting the most important sentences. Le Sun and Nie [46] integrates document index graphics and lexical chains for summarization. Ye et al. [51] presents a unique approach using WordNet and WordNet glosses for summarization.

Alemany and Fort tried to incorporate cohesion and coherence for summarization [5]. They tried to improve a lexical chain based algorithm by enhancing it using discourse markers. They report that their algorithm did not provide significant performance gain.

[1] Malapropism is the unintentional misuse of a word by confusion with one that sounds/spells similar

2.2.2 Text Generation, Text Compression and Smoothing

Ideally, a summarization system should interpret the text, transform it into a semantic representation and generate the summary from the semantic representation. Interpreting the text is a hard problem. Extensive domain knowledge is required for interpretation.

Some researchers tried to fill some predefined templates to create summaries, by treating summarization as information extraction problem [49]. However, this approach is too domain specific and it is not possible to generalize this approach.

Paraphrasing or reducing the sentences extracted by extractive summarization systems could provide more coherent and shorter summaries. Knight and Marcu [31] presents a text compression algorithm. Their work uses probabilistic models and describes a EM (expectation maximization) algorithm to reduce sentences to shorter ones using syntactic parse trees. Their algorithm is also able to fusion multiple sentences into one.

Mani et al. [36] defines a summary revision system which takes in an extract and produces a shorter and more readable version for it. Their system tries to resolve dangling references. Carbonell and Goldstein [3] describes a system called Maximal Marginal Relevance(MMR). Their metric identifies similarity between sentences and represents the repetition in the summary.

Barzilay and McKeown [9] describes a sentence fusion algorithm, which is a text-to-text generation algorithm. This algorithm is very important in the sense that, it can paraphrase sentences. With such a tool, it is possible to convert extracts into abstracts, without understanding the text. Their algorithm takes in similar sentences and outputs a fusion of these sentences.

2.2.3 Evaluating Summarization Systems

Evaluation of summaries is a hard task, as summaries are subjective. Different people will write different summaries for the same document. Evaluation of summaries is a research area by itself. Evaluation methodologies are divided into two main categories. Intrinsic evaluations try to measure the quality of the summary, by defining quality metrics for the summary text. For example, an intrinsic evaluation of selected content's importance is usually done by comparing system generated output summaries to model summaries written by humans. Evaluation is done by measuring the overlap between model and the automatically extracted summary. ROUGE [34] is such an algorithm. Coherence of the summary is usually evaluated by human judges as there were no automatic evaluation methods for coherence. Recently Barzilay and Lapata [50] proposed an automatic evaluation methodology for evaluating coherence of summaries.

Extrinsic evaluations are done by using the summaries in different tasks. For example human annotators use the output summaries to categorize documents. Accuracy of the humans gives the quality of the summaries. Mani et al. [37] describes an extrinsic evaluation methodology based on usefulness of the system summaries.

2.3 Related Work in Keyphrase Extraction

Connection between summarization and keyphrase extraction is clear, but the connection between lexical cohesion and keyphrases has not been issued in summarization literature extensively. Attempts on keyphrase extraction can be classified into two main streams, which are supervised machine learning algorithms and unsupervised machine learning algorithms. Most of the research has been on using supervised machine learning algorithms for keyphrase extraction.

2.3.1 Supervised Machine Learning Techniques

In contrast to summarization, keyphrase extraction is suitable for supervised machine learning algorithms, since assigning class attributes to instances is ambiguous in summarization. For keyprase extraction labeling instances is definite, the author assigned keyphrases are considered as the ground truth.

Turney [56] and KEA algorithm by Witten et al.[58] attacked keyphrase extraction. These two algorithms used first occurrence position in text and frequency based features incorporated with machine learning algorithms. Later Hulth [26] have extended their work by integrating more linguistic features like part of speech tags.

2.3.2 Unsupervised Machine Learning Techniques

There are some unsupervised filtered indexing algorithms, which filter the words in the text using scoring functions based on frequency and TFxIDF.[19, 14, 29] A common belief is that, uncommon phrases for the domain used frequently in a text are keyphrases. To our knowledge there are no unsupervised algorithms for keyphrase extraction, which evaluates their algorithms using author assigned keyphrases. However, keyphrase extraction is very similar to other problems like text categorization and filtered indexing.

Chapter 3

Lexical Chains

3.1 What are Lexical chains?

Lexical chains can be used to model lexical cohesion in documents. A topic can be expressed within a representation formed of words contributing to the topic presentation. When we read a document, we immediately interpret the correct senses of words in that document. Meaning of each word seen in the document contributes to a topic.

Lexical chains are sets of word senses that are related with each other. Let a document D be formed of word occurrences $\{w_1 ... w_i ...\ w_n\}$. These n words are only symbolic representations, meaning of the word can only be determined from the text with prior knowledge. Each word can have more than one sense. For example, word '*bank*' has 10 different senses defined in WordNet. A lexical chain in D is a set of word senses $\{ws_{3_2},\ ws_{6_1},\ ws_{4_{10}},\ ws_{10_2}\}$, where ws_{i_j} is the j'th sense of the word w_i.

Most of lexical cohesion research, concentrated on nouns as they provide more and accurate information. For the example text in Figure 3.1, some of its lexical chains are shown in Figure 3.2.

King Norodom Sihanouk on **Tuesday** praised **agreements** by **Cambodia** 's top two **political parties** previously **bitter rivals** to form a **coalition government** led by **strongman Hun Sen. In** a **short letter** sent to **news agencies** , the **king** said he had received **copies** of **cooperation agreements** signed **Monday** that will place **Hun Sen** and his **Cambodian People** 's **Party** in **firm control** of fiscal and **administrative functions** in the **government** . , " The **protocol** on **cooperation** between the **CPP** and **FUNCINPEC** will certainly bring **peace** and **progress** to our **nation** and **people** , " **Sihanouk** wrote . , Uncompromising **enemies** just a **few months** ago , **Hun Sen** and **FUNCINPEC President Prince Norodom Ranariddh** agreed **Nov.** 13 to form a **government** at a **summit** convened by **Sihanouk** . , The **deal** , which will make **Hun Sen prime minister** and **Ranariddh president** of the **National Assembly** , ended more than three **months** of **political deadlock** that followed a **July election** narrowly won by **Hun Sen. Key** to the **agreement** was the **formation** of a **Senate** as the **upper house** of **Parliament** , to be led by **CPP President Chea Sim** , the **outgoing head** of the **National Assembly** . , **Sihanouk** , recalling **procedures** used in a **past government** , suggested **Tuesday** that he should appoint the first two **members** of the **upper house** . , The remaining **senators** , he said , should be selected by a **method** agreed upon by the **new government** and the **National Assembly** . , **Hun Sen** said **Monday** that the **CPP** and **FUNCINPEC** had agreed that the **Senate** would be half as large as the **122-seat National Assembly** . , **Other details** of the **Senate** , including how **much power** it will be given in the **promulgation** of **legislation** , have yet to be ironed out by the two **parties** .

Figure 3.1: Example Text

LC_1 = {house, government, Senate, house, Senate, assembly, assembly, government, parliament, assembly, assembly, Senate, government, government, government}
LC_2 = {nation, President, people, Key, minister, head, people}
LC_3 = {deal, promulgation, agreement, peace, agreement, agreement}
LC_4 = {Tuesday, Monday, Monday, Tuesday}

Figure 3.2: Lexical Chains for the Text in Figure 3.1

When we look at the sets, we can quickly recognize that these words are related with each other. Note that lexical chains are formed of senses of word occurrences, not senses of unique words in the text. Each word in the lexical chains in Figure 3.2 represents its intended sense of that word.

Figure 3.3 shows the lexical chain graph of LC_3 in detail. Edges are labeled, they represent the semantic relations between senses. Lexical chaining algorithms usually depend on classical relations that can be acquired from WordNet. Meronym/Holonym, Synonym/Repetition, Antonym, Hypernym/Hyponym and Sibling relations are used in our lexical chaining algorithm. Note that these edges are all bi-directional. For example edges e_8, e_3 and e_9 are between senses of '*deal*' and '*agreement*'. '*agreement*' is a hypernym of '*deal*' and '*deal*' is a hyponym of '*agreement*'.

Lexical chain graph edges have weights and edge weights represent the strength of semantic relations between senses. Semantic relations can be weighted to reflect their semantic similarity between word senses, Table 3.1 shows the edge

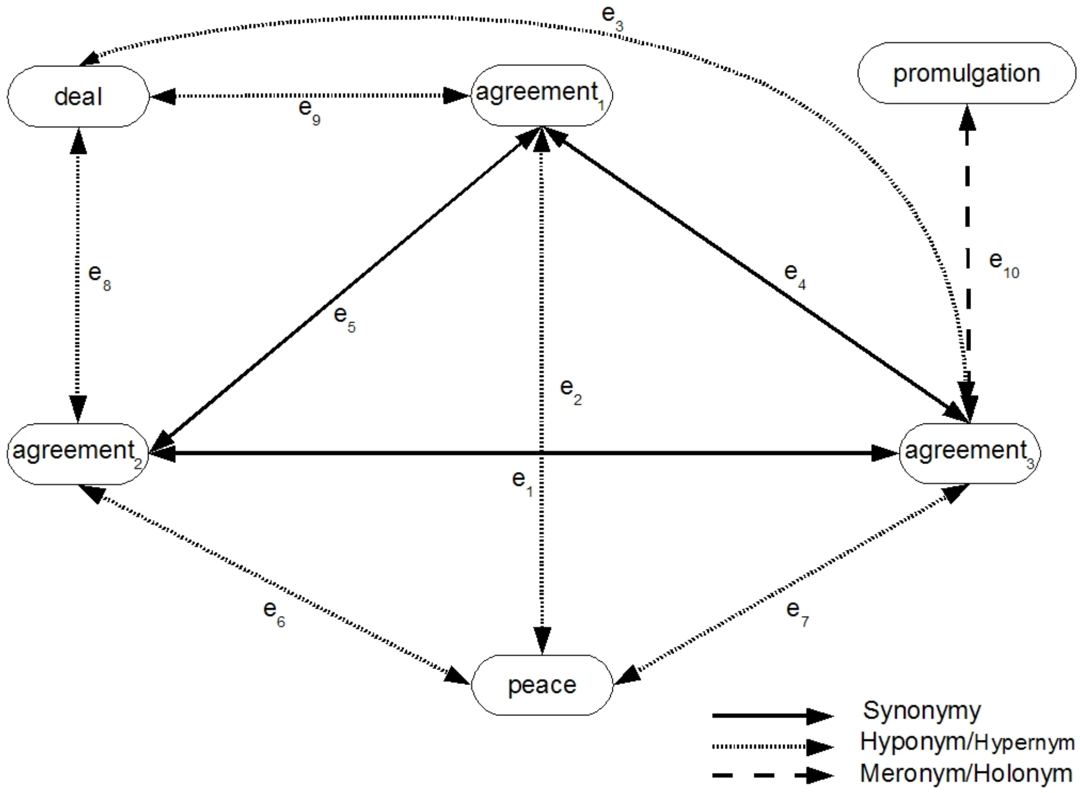


Figure 3.3: Lexical Chain Graph for LC_3 extracted from text in Figure 3.1

weights that we have used for semantic relations. These weights depend on two factors, distance between two words and type of relation. Lexical cohesion is a local feature in text. Since lexical chains reflect topic segments, and topics in a text can change quickly, distance between two words should contribute to the strength measure of the relation. In our example, there is an interesting case for the effect of distance, '*promulgation*' is a part meronym of '*agreement*', but '*promulgation*' is not connected to all instances of '*agreement*'. $agreement_3$ is connected to '*promulgation*' because the distance between them is 5 sentences. Using the Table 3.1 we can see that e_{10} has a weight of 0.3. All other instances of '*agreement*' are at least 7 sentences apart from '*promulgation*', which is our segment boundary. This is another reason for using word occurrence's senses instead of unique word's senses in lexical chains.

Although between '*deal*' and '*promulgation*' there is no relation that could be

	1 Sentence	3 Sentence	1 Paragraph/Segment	Other
Iteration/Synonym	1	1	1	1
Antonym	1	0.3	0.2	0
Hypernym/Hyponym	1	0.5	0.5	0.5
Meronym/Holonym	1	0.3	0.3	0
Sibling	1	0.3	0.2	0

Table 3.1: Edge Weights For Semantic Relations

acquired from WordNet's classical relations, they appear in the same lexical chain in our example. These two senses are connected to each other through the word sense '*agreement*'. Lexical chains are connected graphs. This fact introduces a constraint for lexical chains, no two lexical chains can share a sense of a word occurence. Synonym's have an edge weight of 1 for any distances, this enforces that a sense can be a member of one and only one lexical chain. However this constraint does not imply that a word should be a member of single lexical chain. Remember that words can have multiple senses, and in a document a word can be used in different senses. For example, in a document that describes a '*village*', '*financial institutions*' and '*banks*' of the village could be described in the first paragraphs. Later the same document could talk about its '*river*' and the '*river bank*'. In this case word '*bank*' is used in two different senses. These two senses will be member of different lexical chains. This is an extreme case and usually one sense per word is used in articles. For this reason some lexical chaining algorithms, including ours, impose one sense per word constraint.

3.2 Lexical Chaining Algorithms

For extracting lexical chains in a document, all words and correct senses of these words should be known. Humans disambiguate words by the current context. Lexical chaining algorithms depend on an assumption, and this assumption is that correct sense of words has stronger relations with other word senses. Using this assumption, lexical chaining algorithms first try to disambiguate all word occurrences. For this reason, word sense disambiguation (WSD) is an immediate application of lexical chains and an extrinsic evaluation methodology.

The **US congress** has started an **inquiry** about the **country** 's **ability** in **intelligence** . The **intelligence agencies** in the **country**, **CIA** and **NSA** are investigated after the **terrorist attack** on **September 11**.

Figure 3.4: Lexical Chain Example Text

Before going into more detail, some definitions are required to explain the algorithms more easily.

Definition 1 *Interpretation of a document is one of possible word sense combinations in the word sense space. For example using the first senses for all words in a document is an interpretation of the document.*

Definition 2 *Interpretation Space is the set of all possible interpretations that can be formed for a document. Interpretation space is formed of all combinations of word senses. For a document with n words where each word has 2 senses there are 2^n interpretations in the interpretation space.*

Definition 3 *Word Sense Graph is the graph of an interpretation where nodes are senses and the edges are relations between senses.*

In a text, if the correct interpretation is known, then lexical chains are the connected subgraphs in the word sense graph.

In Figure 3.4, an example text fragment is shown, where noun phrases are in bold. Table 3.2 shows these nouns, number of senses in WordNet and intended sense's gloss. The word sense graph in Figure 3.5 shows the relations between these senses. Forming the lexical chains from a word sense graph is an easy task. The connected subgraphs are the lexical chains for the text. Lexical chains for this text are given in Figure 3.6

In Table 3.2 number of word senses are given. Even in such a small text the interpretation space contains 750 interpretations [1].

[1] Note that the word '*country*' appears twice in the given text, for this reason when calculating the interpretation space, country's 5 senses are considered twice

Word	**Number of senses in WordNet**	**Correct Sense**
Us Congress	1	The legislature of the United States government
inquiry	3	a systematic investigation of a matter of public interest
country	5	a politically organized body of people under a single government; "the state has elected a new president"; "African nations"; "students who had come to the nation's capitol"; "the country's largest manufacturer"; "an industrialized land
ability	2	possession of the qualities (especially mental qualities) required to do something or get something done; "danger heightened his powers of discrimination"
intelligence	5	a unit responsible for gathering and interpreting information about an enemy
intelligence agencies	1	a unit responsible for gathering and interpreting information about an enemy
CIA	1	an independent agency of the United States government responsible for collecting and coordinating intelligence and counterintelligence activities abroad in the national interest; headed by the Director of Central Intelligence under the supervision of the President and National Security Council
NSA	1	the United States cryptologic organization that coordinates and directs highly specialized activities to protect United States information systems and to produce foreign intelligence information
terrorist attack	1	a surprise attack involving the deliberate use of violence against civilians in the hope of attaining political or religious aims
September 11	1	the day in 2001 when Arab suicide bombers hijacked United States airliners and used them as bombs

Table 3.2: Word Senses for the words in Figure 3.4

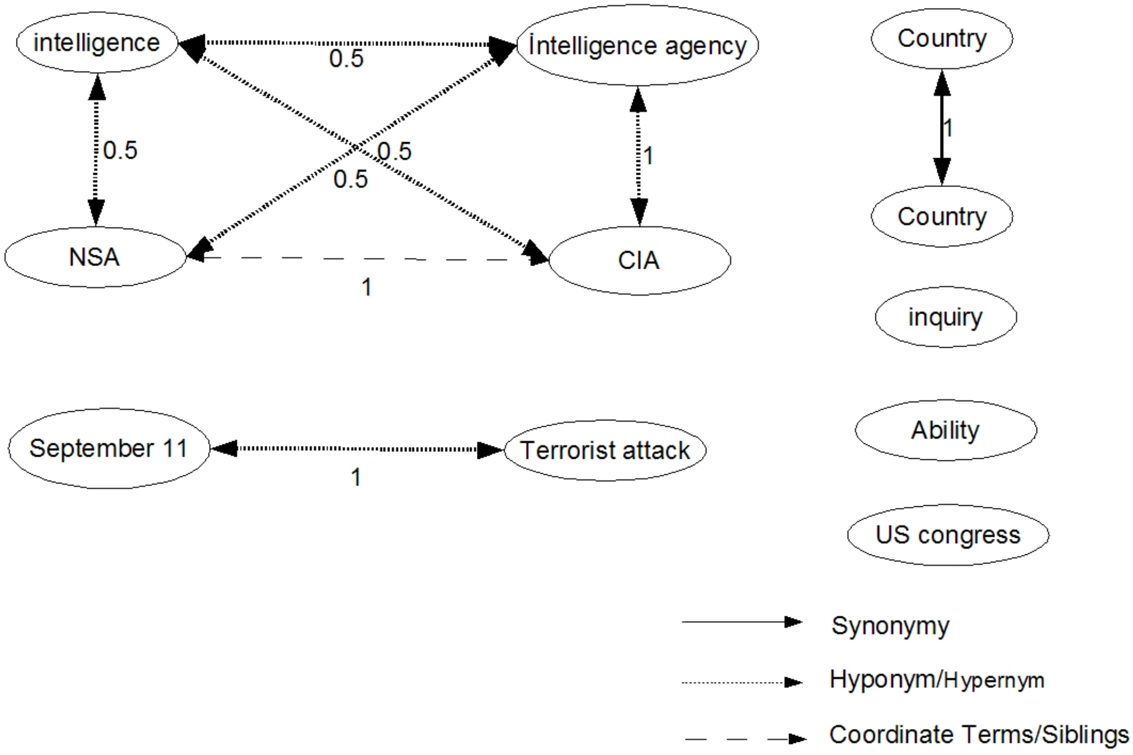


Figure 3.5: Word relation graph

LC_1 ={intelligence, intelligence agency, CIA, NSA}
LC_2 ={September 11, terrorist attack}
LC_3 ={country, country}
LC_4 ={inquiry}
LC_5 ={ability}
LC_6 ={US congress}

Figure 3.6: List of Lexical Chains

If we assume that the correct sense of a word is the sense that has stronger relations with surrounding words, then the correct interpretation for a document is the interpretation that has the word sense graph with the highest sum of edge weights. With these definitions, we can claim that finding lexical chains is equal to finding the connected subgraphs in word sense graph of the best interpretation. However, size of the interpretation space increases rapidly with number of words making it infeasible to search the whole interpretation space.

3.2.1 Greedy Approaches

First algorithms for building lexical chains were greedy algorithms. These algorithms do not build all possible interpretations but disambiguate words using the words on its left. Morris and Hirst [43] has outlined a general algorithm for automatic construction of lexical chains. As WordNet was not available at that time, they have used Roget's Thesaurus as a knowledge base. This algorithm is a greedy approach to lexical chaining. Every candidate word sense's semantic relations are looked up and its membership to lexical chains are searched. Disambiguation is done as soon as there is a relation with any of the word's senses.

Let w_i be a word in the document, and w_i have n senses $\{w_{i_1}...w_{i_j}...w_{i_n}\}$. If any one sense w_{i_j} of w_i has a relation with w_{x_y}, any member of previously created lexical chain LC_f, w_i is included in this chain and w_{i_j} is selected as its correct sense. If more than one sense of w_i is related with lexical chain(s), then the w_{i_j} which has stronger relation(s) is selected. If two senses of w_i are connected with equal strengths, then sense which is more common is chosen. If none of the senses of w_i can be associated with any current lexical chains then a new lexical chain is created, disambiguation is postponed until a related sense is encountered.

St Onge and Hirst [53] has adopted Morris and Hirst's algorithm, and they used WordNet. Their algorithm uses 3 classes of relationships.

- **Extra Strong Relations** are all the repetitions of the same word.
- **Strong Relations** include synonyms, hypernyms/hyponyms, holonyms/meronyms

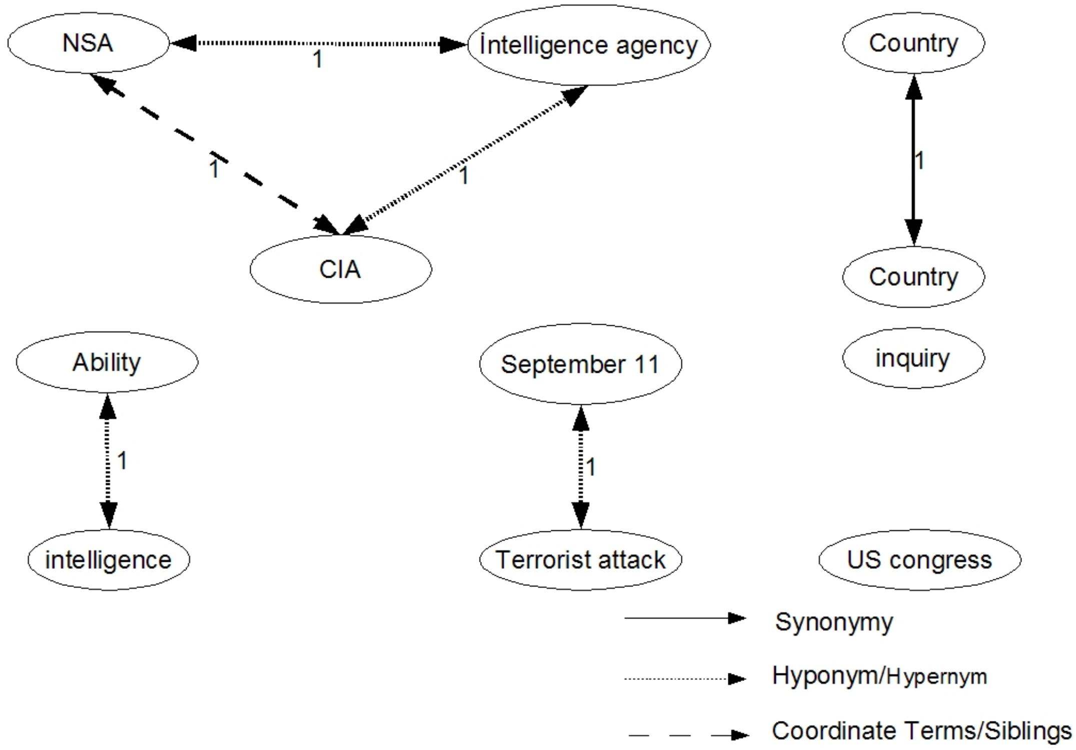


Figure 3.7: Greedy Word Sense Graph

and antonyms.

- **Medium Strong Relations** includes all relations in WordNet with allowable paths of length 5.

Relation types and edge weights used in this algorithm differs from our algorithm. This work is very important in the sense that, it is the first automated lexical chaining algorithm which uses WordNet.

Greedy approaches try to disambiguate the word's sense using only the context on the left hand side of the word. This may result in errors in disambiguation, as the word's correct sense could have relations with words that appear after its occurrence.

We try to demonstrate this algorithm for the text in Figure 3.4 to provide

Step Number	Word Processed	Action	Lexical Chains before the step
1	US congress	not related to any Lexical Chain create a new lexical chain	
2	inquiry	not related to any Lexical Chain create a new lexical chain	LC1={US congress}
3	country	not related to any Lexical Chain create a new lexical chain	LC1={US congress}, LC2={inquiry}
4	ability	not related to any Lexical Chain create a new lexical chain	LC1={US congress}, LC2={inquiry }, LC3={country}
5	intelligence	related to LC4 intelligence is a kind of ability hypernym	LC1={US congress}, LC2={inquiry}, LC3={country}, LC4={ability}
6	intelligence agencies	not related to any Lexical Chain create a new lexical chain	LC1={US congress}, LC2={inquiry}, LC3={country}, LC4={ability, intelligence}
7	country	reiteration of country in LC3	LC1={US congress}, LC2={inquiry}, LC3={country}, LC4={ability}, LC5={intelligence agencies}
8	CIA	CIA is a kind of intelligence agency hypernym, add to LC6	LC1={US congress}, LC2={inquiry}, LC3={country, country}, LC4={ability}, LC5={intelligence agencies}
9	NSA	NSA is a kind of intelligence agency hypernym add to LC6	LC1={US congress}, LC2={inquiry}, LC3={country, country}, LC4={ability}, LC5={intelligence agencies, CIA}
10	terrorist attack	not related to any Lexical Chain create a new lexical chain	LC1={US congress}, LC2={inquiry}, LC3={country, country}, LC4={ability}, LC5={intelligence agencies, CIA, NSA}
11	September 11	September 11 is a kind of terrorist attack hypernym add to LC7	LC1={US congress}, LC2={inquiry}, LC3={country, country}, LC4={ability}, LC5={intelligence agencies, CIA, NSA}, LC6={terrorist attack}
Result			LC1={US congress}, LC2={inquiry}, LC3={country, country}, LC4={ability}, LC5={intelligence agencies, CIA, NSA}, LC6={terrorist attack, September 11}

Table 3.3: Greedy Lexical Chaining Algorithm Execution

a better understanding of this problem with greedy lexical chaining algorithms. Table 3.3 shows the whole algorithm execution. In step 5, '*intelligence*' which is used in the sense *intelligence agency* has been falsely disambiguated as *intelligence* as a *person's ability*. This error is due to the fact that the word '*intelligence*' is disambiguated using the context before its occurence.

3.2.2 Non-Greedy Approaches

To achieve better disambiguation in lexical chaining algorithms, all of the word senses and word relations should take part in the word sense disambiguation process. Differing from greedy approaches, non-greedy approaches disambiguate words in a second pass after all words are processed.

3.2.2.1 A Naive Approach

A naive approach for lexical chaining could be to construct all possible interpretations for the text and score each interpretation using their relation strengths. This approach forms and searches the whole interpretation space.

For example, for the text in Figure 3.4, when the word 'US Congress' is processed, 1 interpretation is created as this word has only one sense. When the word 'inquiry' is processed number of interpretations are multiplied by 3 since the word 'inquiry' has 3 senses in WordNet. When two sentences are processed, there will be 750 interpretations in consideration. The correct word sense graph is given in Figure 3.5 and word sense graph of the interpretation that is the output of the greedy algorithm is given in Figure 3.7.

Using edge weights, each interpretation is scored. Summing the edge weights, scores for two interpretations in Figure 3.7 and Figure 3.5 are calculated. If we use the edge weights in Table 3.1, the score of the interpretation in Figure 3.7 will be 12 and the score of the interpretation in Figure 3.5 will be 13. Edges are bidirectional, while '*intelligence*' has a hypernym relation to '*ability*', '*ability*' has a hyponym relation to '*intelligence*'. Thus, correct interpretation can be in Figure 3.5.

This naive approach is not feasible in the sense of computational time. It is not feasible to form all interpretations, since size of the interpretation space grows rapidly with the number of nouns in the text. Instead of searching the whole space, some approximations are required.

3.2.2.2 Barzilay et al.'s algorithm

First non-greedy algorithm for lexical chaining is the work of Barzilay and Elhadad [6]. Their algorithm tries to form all strongly connected interpretations of text and chooses the best one in terms of semantic connectedness.

Barzilay has applied some simple techniques to improve the naive approach.

Instead of disambiguating the whole document, Barzilay's algorithm keeps track of exclusive components in the text. Components in Barzilay's algorithm partitions the interpretation space for a subset of words that are co-related in any sense. Components decreases the number of interpretations built. For the sake of efficiency in computation time and memory space, Barzilay prunes the interpretations that have low scores, when the number of interpretations in the system exceeds some threshold.

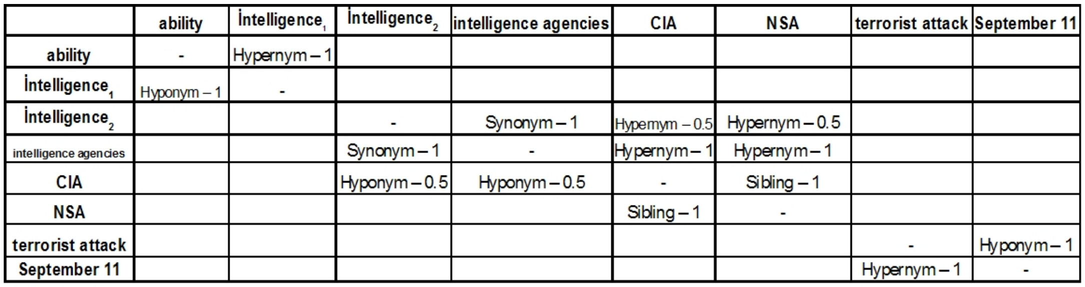
Their algorithm divides the text into segments. Text segmentation is done explicitly using the algorithm defined by Hearst [24], which uses word repetitions to detect segment boundaries. Barzilay et al. processes the lexical relations between two words only if they are in the same segment. The effects of using segments in lexical chaining algorithms have been investigated by both Barzilay and Silber [52]. Segments are used for both efficiency purposes and to account for the distance factor in relations. In our algorithm, we use edge weights depending on distance, Barzilay et al.'s algorithm uses fixed weights for relations: 10 for synonymy, 7 to antonyms, 4 to hypernymy/hyponymy and meronymy/holonymy. Intra-segment synonym relations are merged after processing all the words. Note that in our edge weights synonym has the weight 1 for any distances, which provides a similar result. We believe that distance sensitive edge weights provides a more natural way for lexical chaining and word sense disambiguation then using strict segments. Our summarization algorithm implicitly defines segments, using all the lexical cohesion clues in the text.

3.2.2.3 Silber et al.'s Algorithm

Silber and Mccoy [52] describes an efficient algorithm for lexical chaining. Their algorithm differs from Barzilay's work in two important aspects. In Silber et al.'s algorithm, the lexical chains are not built for each interpretation explicitly, instead, sense and relations are gathered in a graph. In order to disambiguate a word, the amount of semantic relation score that the word sense contributes to the graph is calculated for each word sense, and the sense that contributes the most is the correct sense. In order to determine the contribution strength of a sense, they sum the weights of all the edges reachable from the sense node. The other senses are removed from the graph. All of the words are disambiguated using this technique in order to obtain the final interpretation and set of lexical chains. In contrast to Barzilay's work, this algorithm does not use an external text segmentation algorithm. The effect of distance is provided by edge weights that depend on the distance between two senses. Edge weights used in their algorithm are very similar to our weights in Table 3.1. Silber and Mccoy also report that indexing the nouns in WordNet significantly improved running time of their algorithm.

3.2.2.4 One Sense Per-Word Constraint

Evaluation methodology for lexical chaining algorithms is discussed later in this chapter, however Galley and McKeown [21] report that the word sense disambiguation accuracy of Silber and McCoy's algorithm is lower than Barzilay's algorithm. They provide an algorithm that they report to perform better in word sense disambiguation(WSD). Their algorithm separates the lexical chaining and WSD phases. They impose one sense per word constraint for a document. Generally a word is used in the same sense in a document, but it is possible that a word could be used in different senses in a document. With this constraint, all the clues and relations obtained from multiple occurrences of the word, contribute to a single decision for the word's sense. The most important difference between Silber's algorithm and this algorithm is that, in Silber's algorithm each word occurrence is disambiguated separately, but in this algorithm a word with its all occurrences is disambiguated once. Their algorithm is very similar to Silber's algorithm, all of the nouns are processed, all senses and relations are found forming a disambiguation graph. From all word senses, the one with the strongest relations is selected. Relation scores are gathered from all occurrences of the word.



	ability	İntelligence$_1$	İntelligence$_2$	intelligence agencies	CIA	NSA	terrorist attack	September 11
ability	-	Hypernym – 1						
İntelligence$_1$	Hyponym – 1	-						
İntelligence$_2$			-	Synonym – 1	Hypemym – 0.5	Hypernym – 0.5		
intelligence agencies			Synonym – 1	-	Hypernym – 1	Hypernym – 1		
CIA			Hyponym – 0.5	Hyponym – 0.5	-	Sibling – 1		
NSA					Sibling – 1	-		
terrorist attack							-	Hyponym – 1
September 11							Hypernym – 1	-

Table 3.4: Sense Relation Table

3.2.2.5 Our algorithm

We implemented Galley's algorithm with two simple modifications. First, we integrated meronym relations to their algorithm to increase connectivity. Second, in order to choose the correct sense, we used all the edges that can be reached from the words nodes instead of just immediate edges. Table 3.4 shows the sense relation matrix after processing all senses and relations. In this table, we have ignored senses that does not have any relations. Two different senses of the word '*intelligence*' is shown in the table, *intelligence*$_1$ is related with *ability* and *intelligence*$_2$ is a synonym of *intelligence agency*. Relation types and relation weights are given in the table. When our algorithm tries to disambiguate word *intelligence*, for both senses total semantic scores are calculated. The sense *intelligence*$_1$ has a score 2 that is equal to sum of all incoming and outgoing edges. The sense *intelligence*$_2$ has a score of 4 that is equal to its incoming and outgoing edges. Sum of all edges that can be reached from *intelligence*$_2$ is 8. Differing from Galley's algorithm we used 8 as the score of intelligence, instead of 4. The sense which contributes more to the graph, *intelligence*$_2$ is selected as the sense of the word '*intelligence*' for this text.

3.3 Evaluation of Lexical Chaining Algorithms

There are two important questions for lexical chains: what is the accuracy of a lexical chain, and how important is the lexical chain. A basic approach for evaluating the accuracy of lexical chaining algorithms is the usage of word sense disambiguation corpora. The found word senses in a lexical chain are compared with the correct senses in the corpora to find the accuracy of a lexical chain. Barzilay has done experiments, taking advantage of human evaluations to reflect the importance of lexical chains in text.

3.3.1 Correctness of Lexical Chains

Lexical chains are very hard to evaluate. Even for a human, it is hard and subjective to extract lexical chains of a document. There are no corpora to evaluate the lexical chains directly. For this reason, there are no intrinsic evaluation methods for lexical chains. Some extrinsic evaluation methods are available. Lexical chains are used in different applications like topic segmentation and summarization. However, it is not possible to asses the accuracy of the lexical chains using the outputs obtained from these applications as the effect of lexical chains in summarization itself is questionable. Usually, lexical chains are evaluated using word sense disambiguation corpora. Comparison of lexical chaining algorithms are discussed in detail in Silber et al.[52] and Galley et al.[21]. Galley experimented with a word sense disambiguation corpus and found out that Barzilay's algorithm has an accuracy of %57. Silber's algorithm has performed a lower score, %53. Algorithm described by Galley has an accuracy of %63. Galley has the advantage of using all occurences of the word to decide the correct sense. Our algorithm is very similar to Galley's algorithm, for this reason we expect it to have a similar accuracy in word sense disambiguation.

3.3.2 Strength of Lexical Chains

Lexical chain score defined by Barzilay et al. [6] has been used in different applications. Barzilay et al. reports that they have experimented with different properties of lexical chains. They report that Equation 3.1 is the best indicator of importance for a lexical chain. They also report that lexical chains satisfying the criteria in Equation 3.2 are important lexical chains.

$$Score(Chain) = Length * Homogeneity \tag{3.1}$$

$$Score(Chain) > Average(Scores) + 2 * StandartDeviation(Scores) \tag{3.2}$$

$$Homogeneity = 1 - \frac{\#DistinctMembers}{Length} \tag{3.3}$$

Length is the number of members of the lexical chain. Homogeneity is shown on Equation 3.3. This scoring function has been used by many researchers. We used Equation 3.1 to rank our lexical chains, and incorporated this score with sequence score which will be described in the following chapters.

Chapter 4

Single Document Summarization

An application for lexical chains is summarization. Any intelligently written text is divided into topics. Lexical chains are a good tool for topic identification. Senses in a lexical chain contribute to a topic. Our algorithm uses lexical chains to identify topics and topic segments, and tries to extract most significant topics and sentences representing these topics to form a summary.

In this chapter, we describe the main parts of our summarization algorithm which uses lexical chains. The implementation details and other components of our algorithm are discussed in Chapter 7. Here we describe the usage of lexical chains in text segmentation, and extraction of the important sentences. We also compare our algorithm with other single document summarization systems.

4.1 Summarization Algorithm

For summarization, our algorithm segments the text, according to topics, using the lexical chains created for the text. Topics in the text is roughly determined using lexical chains. Through clustering of lexical chains, our algorithm produces more granular segments. In each segment, it is assumed that first sentence is a general description of the topic, first sentence of the segment is selected to be included in the summary.

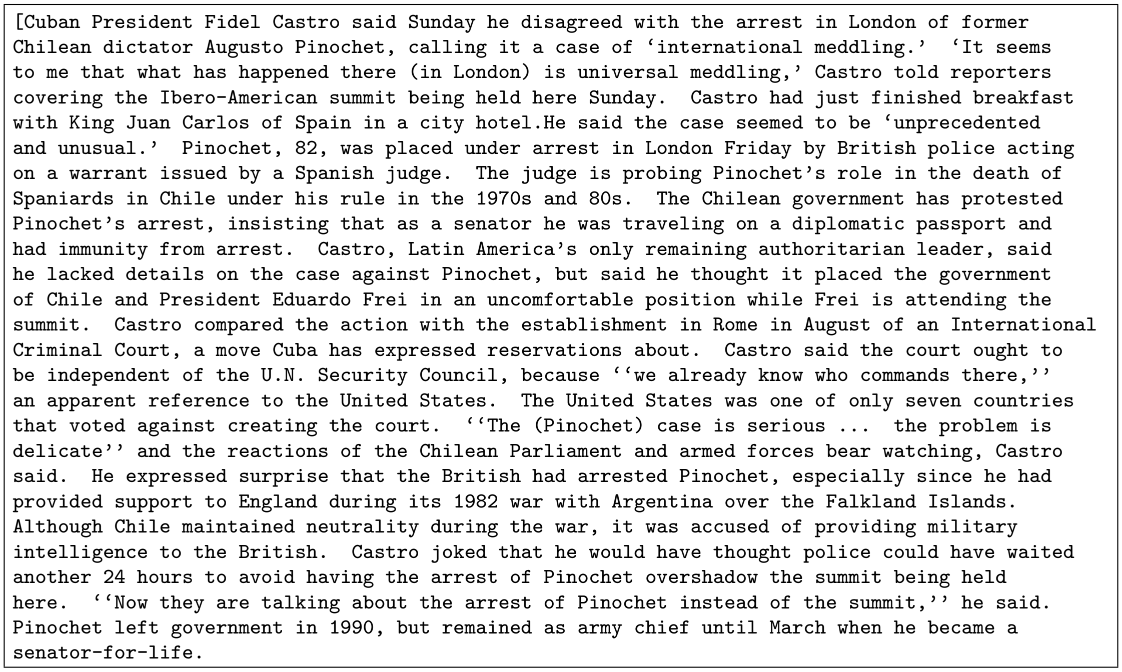
Our algorithm is based on lexical chains, for this reason, our system requires deeper analysis of the text. An outline of our algorithm could be given as:

1. Sentence Detection
2. Part of Speech Tagging
3. Noun Phrase Detection
4. Lexical Chaining
5. Filtering Weak Lexical Chains
6. Clustering Lexical Chains Based on Co-occurrence
7. Extracting Sequences / Segmenting the Text Regard to Clusters.

Sentence detection, part of speech tagging and Noun Phrase Detection are detailed in Chapter 7. Part of speech tagging is necessary to identify nouns in the text. Noun phrase detection is required, since noun phrases are treated differently from words. Although WordNet contains some compound nouns, most of domain specific noun phrases are not contained in WordNet. The semantic contribution of these phrases are calculated using contribution of their head nouns. For example; the phrase '*quantum computing*', will be related to '*computing*' rather than '*quantum*'.

Our lexical chaining procedure is explained in Chapter 3. We are using one sense per word constraint in our lexical chaining algorithm. Our lexical chaining algorithm is very similar to Galley et al.'s algorithm [21] and its details are given in Chapter 3.

After lexical chains are constructed for the text, there will be some weak lexical chains formed of single word senses. These lexical chains can cause complications in topic identification and segmentation. Barzilay et al. [6] presents the formula in Equation 3.1. This formula has been formulated to reflect the strength of



[Cuban President Fidel Castro said Sunday he disagreed with the arrest in London of former Chilean dictator Augusto Pinochet, calling it a case of ‘international meddling.’ ‘It seems to me that what has happened there (in London) is universal meddling,’ Castro told reporters covering the Ibero-American summit being held here Sunday. Castro had just finished breakfast with King Juan Carlos of Spain in a city hotel.He said the case seemed to be ‘unprecedented and unusual.’ Pinochet, 82, was placed under arrest in London Friday by British police acting on a warrant issued by a Spanish judge. The judge is probing Pinochet’s role in the death of Spaniards in Chile under his rule in the 1970s and 80s. The Chilean government has protested Pinochet’s arrest, insisting that as a senator he was traveling on a diplomatic passport and had immunity from arrest. Castro, Latin America’s only remaining authoritarian leader, said he lacked details on the case against Pinochet, but said he thought it placed the government of Chile and President Eduardo Frei in an uncomfortable position while Frei is attending the summit. Castro compared the action with the establishment in Rome in August of an International Criminal Court, a move Cuba has expressed reservations about. Castro said the court ought to be independent of the U.N. Security Council, because ‘‘we already know who commands there,’’ an apparent reference to the United States. The United States was one of only seven countries that voted against creating the court. ‘‘The (Pinochet) case is serious ... the problem is delicate’’ and the reactions of the Chilean Parliament and armed forces bear watching, Castro said. He expressed surprise that the British had arrested Pinochet, especially since he had provided support to England during its 1982 war with Argentina over the Falkland Islands. Although Chile maintained neutrality during the war, it was accused of providing military intelligence to the British. Castro joked that he would have thought police could have waited another 24 hours to avoid having the arrest of Pinochet overshadow the summit being held here. ‘‘Now they are talking about the arrest of Pinochet instead of the summit,’’ he said. Pinochet left government in 1990, but remained as army chief until March when he became a senator-for-life.

Figure 4.1: An Example News Article

lexical chains. Barzilay et al reports that this is the best formula that correlates with the human judges. After lexical chain construction, Barzilay suggests that lexical chains below a strength criterion should be filtered. Strength criterion is defined as Equation 3.2. We have filtered weak lexical chains before clustering lexical chains.

4.1.1 Clustering Lexical Chains

All strong lexical chains in the document are clustered using co-occurrence statistics. A single lexical chain may not be sufficient to represent a single topic. Figure 4.2 gives the important clusters of lexical chains constructed for the document in Figure 4.1[1].

A topic could be formed of words that are not necessarily co-related. For example, in Figure 4.2 cluster 3 is a good example. This cluster talks about

[1] Proper names in the text, ’Pinochet’ and ’Frei’ are not present in WordNet. We have ignored nouns that are not in WordNet. Thus, ’Pinochet’ and ’Frei’ are not considered in our algorithm

Cluster1 :
LC_1= {Castro, Castro, chief, Castro, Castro, Castro, Castro, Castro, Castro, leader}
V_1 = {1,1,1,0,0,0,0,2,1,1,0,1,0,0,1,0,1}
LC_2 ={establishment, United States, parliament, United States, government, government, government}
V_2 = {0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,1}
Cluster2 :
LC_1= {action, march, meddling, arrest, arrest, arrest, surprise, arrest, meddling, arrest, arrest}
V_1 = {2,1,0,0,1,0,2,0,1,0,0,0,1,0,1,1,1}
LC_2 = {London, London}
V_2 = {1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
LC_3 = {Sunday, Sunday}
V_3 = {1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
Cluster 3 :
LC_1 = {summit, summit, summit, summit}
V_1 = {1,1,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0}
Cluster 4 :
LC_1 = {Chile, Argentina, Chile, Chile}
V_1 = {0,0,0,0,0,1,0,1,0,0,0,0,1,1,0,0,0}
LC_2 = {war, war}
V_2 = {0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0}
Cluster 5 :
LC_1 = {court, court}
V_1 = {0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0}

Figure 4.2: Lexical Chain Clusters for the Example in figure 4.1

an '*arrest*' in '*London*' on '*Sunday*'. These three sets and their relations with each other can only be determined by the current context. We believe that through clustering, we are forming a relation between these lexical chains. In cluster 3, lexical chains in the cluster are forming up the relations 'what', 'where' and 'when' respectively. Our clustering algorithm depends on a very simple assumption, if two lexical chains tends to appear in same sentences, then there may be a relation between two sets in the given context. It is clear that, this will not hold in all cases. There will be falsely related lexical chains, however, a more accurate algorithm requires deeper semantic analysis. Our approach is just accurate enough for our segmentation algorithm.

In cohesion relations, like reference, substitution and ellipsis, word is not repeated in each sentence but replaced or omitted. Through clustering, we can be able to account for other cohesion clues other than lexical cohesion, for example ellipsis. By forming the link between two or more lexical chains by co-occurrence, it is possible to consider all lexical cohesion relations while segmenting the text. It might enable us to detect topic segments more accurately.

For each lexical chain LC_i, a sentence occurrence vector V_i is formed. $V_i = \{s_{1_i}, ...s_{k_i}...s_{n_i}\}$ where n is the number of sentences in the document. Each s_{k_i}

Sequence 1:
[The Chilean government has protested Pinochet 's arrest , insisting that as a senator he was traveling on a diplomatic passport and had immunity from arrest . , Castro , Latin America 's only remaining authoritarian leader , said he lacked details on the case against Pinochet , but said he thought it placed the government of Chile and President Eduardo Frei in an uncomfortable position while Frei is attending the summit . , Castro compared the action with the establishment in Rome in August of an International Criminal Court , a move Cuba has expressed reservations about . , Castro said the court ought to be independent of the U.N. Security Council , because " we already know who commands there , " an apparent reference to the United States . , The United States was one of only seven countries that voted against creating the court . , " The -LRB- Pinochet -RRB- case is serious ... the problem is delicate " and the reactions of the Chilean Parliament and armed forces bear watching , Castro said .]
clusterLexicalChainScore= 17.0 noOfChainsInCluster= 2
Cluster 1
Starting lexical chains : LC2
Participating Lexical chains : LC1, LC2
Sequence 2 :
[Cuban President Fidel Castro said Sunday he disagreed with the arrest in London of former Chilean dictator Augusto Pinochet , calling it a case of " international meddling . " , " It seems to me that what has happened there -LRB- in London -RRB- is universal meddling , " Castro told reporters covering the Ibero-American summit being held here Sunday .]
clusterLexicalChainScore= 15.0 noOfChainsInCluster= 3
Cluster 2
Starting lexical chains: LC1, LC2, LC3 Participating Lexical chains: LC1, LC2, LC3
Sequence 3 :
[Castro compared the action with the establishment in Rome in August of an International Criminal Court , a move Cuba has expressed reservations about . , Castro said the court ought to be independent of the U.N. Security Council , because " we already know who commands there , " an apparent reference to the United States . , The United States was one of only seven countries that voted against creating the court .]
clusterLexicalChainScore= 3.0 noOfChainsInCluster= 1 noOfSentences= 17
Cluster 5
Participating Lexical chains : LC1 Starting Lexical chains : LC1
Sequence 4 :
[He expressed surprise that the British had arrested Pinochet , especially since he had provided support to England during its 1982 war with Argentina over the Falkland Islands . , Although Chile maintained neutrality during the war , it was accused of providing military intelligence to the British .]
clusterLexicalChainScore= 6.0 noOfChainsInCluster= 2 noOfSentences= 17
Cluster 4
Participating Lexical chains : LC1, LC2 Starting Lexical chains : LC2

Figure 4.3: Sequences For Example in 4.1

is the number of LC_i members in the sentence k. If sentence k has 3 members of LC_i then s_{k_i} is 3. Two lexical chains LC_i and LC_j goes into the same cluster if their sentence occurrence vectors V_i and V_j are similar. Clustering of lexical chains will yield in clusters with two properties;

- Lexical chains that co-occur will be on the same clusters. These lexical chains form a set of topics that talk about a single topic.
- Lexical chains that span different sentences will be on different clusters. Two lexical chains that are on different clusters are considered to be unrelated.

Our clustering algorithm, starts from an initial cluster distribution, where each lexical chain is in its own cluster. Thus, our clustering algorithm starts with n clusters, where n is the number of lexical chains. Iteratively the most similar cluster pair is found and they are merged to form a single cluster. Clustering stops when the similarity between the most similar clusters is lower than a threshold value τ.

The similarity between two clusters is measured by finding the similarity between the least similar members of the cluster. This is called **complete link clustering**. Since cluster members are lexical chains in our algorithm, a similarity function measuring the co-occurrence between two lexical chains is needed. We have used cosine similarity for this purpose. Lexical chain occurrence vector V_i is a vector in a m dimensional space, where m is the number of sentences. The angle between two vectors, could be used to find the similarity of two vectors. Between two vectors that are in the same direction, there will be an angle of 0 degrees. Cosine of two vectors can be calculated by Equation 4.1. This value is between 0 to 1, where 1 means most similar.

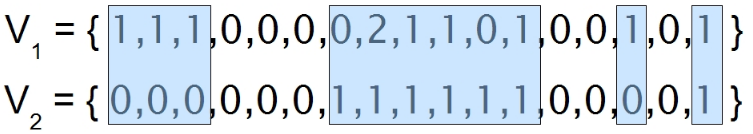
$$\cos(\theta) = \frac{V_i \cdot V_j}{\|V_i\| \, \|V_j\|} \tag{4.1}$$

Equation 4.1 is a well known formula from Linear Algebra, to find the cosine of the angle between two vectors. In the equation $\|V_i\|$ represents the euclidean length for the vector, that is the square root of the sum of squares of vector's dimension values .

4.1.2 Sequence Extraction or Text Segmentation

Some previous algorithms for lexical chain based summarization, depend on explicit segmentation algorithms, such as Brunn et al.[11], and Barzilay [6]. In our algorithm, the text is segmented from the perspective of each lexical chain cluster, finding the hot spots for each topic. For each cluster, connected sequences of sentences are extracted as segments. Sentences that are cohesively connected are usually talking about the same topic.

Figure 4.3 is an example of sequences extracted from the text in Figure 4.1. For each lexical chain cluster Cl_j, we form sequences separately. For each sentence S_k, if sentence S_k has a lexical chain member in Cl_j, a new sequence is started or the sentence is added to the sequence. If there is no cluster member in S_k, then the sequence is ended. By using this procedure, text is segmented with respect to a cluster, identifying topic concentration points.



V_1 = { 1,1,1,0,0,0,0,2,1,1,0,1,0,0,1,0,1 }

V_2 = { 0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,1 }

Figure 4.4: Text Segmentation for Cluster 1 in Figure 4.2

An example of text segmentation for the text in Figure 4.1, with respect to cluster 1 shown in Figure 4.2, is given in Figure 4.4. In cluster 1, there are two lexical chains. The sentence occurrence vectors for these lexical chains are plotted in the figure. Highlighted areas correspond to the sequences in the text. This topic seems to be concentrated on the second sequence extracted as this segment has contributions from both of the lexical chains and spans more than the other segments.

Each sequence is scored using the formula in Equation 4.2.

$$Score(sequence_i) = score(Cl_i) * L_i * \frac{(1 + SLC_i) * PLC_i}{f^2} \tag{4.2}$$

Where L_i is the number of sentences in the $sequence_i$. SLC_i is the number of lexical chains that starts in $sequence_i$. PLC_i is the number of lexical chains having a member in $sequence_i$ and f is the number of lexical chains in cluster. Score of the cluster $score(Cl_i)$, is the average score of the lexical chains in the cluster. Our scoring function tries to model the connectedness of the segment using this cluster score. Note that this score is calculated with the formula in Equation 3.1. Number of sentences in the segment, reflects for how long topic is discussed locally. Our algorithm tries to select the segments that lexical chains are starting in, this will encourage selection of segments that the topic is first introduced in.

4.1.3 Sentence Selection

Humans tend to first explain the topic more generally and then give details in the following sentences. With this motivation, our algorithm extracts the first sentence of each sequence. So, if the sequences are truly topic segments for the text, then our algorithm will extract the first sentence of the new topic. This technique depends on the assumption that, first sentences are general descriptions of the topic and this general description does contain sufficient information to represent the text segment in the summary.

For a summary of length n sentences, n best scoring sequence's first sentences are included in the summary. However, two different sequences found from different lexical chain clusters can start with the same sentences. A problem with this approach may be that n could be higher than the number of sequences starting with a unique sentence, so the number of sentences to be included in the summary is limited by the number of sequences starting with unique sentences. It is possible for two sequences extracted from different lexical chain clusters to overlap in text area.

We will try to demonstrate our algorithm using the news article in Figure 4.1. After lexical chaining and clustering with a τ equal to 0.5, top ranking clusters are given in Figure 4.2. In cluster 4, the connection between 'Chile' and 'Argentina' is 'war'. This is discovered from the given context using co-occurrence in the given text. Clustering increases the connectedness of sentences, resulting in granular text segments.

Figure 4.3 shows the top ranking sequences that contribute to the summary. These sequences correspond to the most significant topic segments in the document. As a result of this process summary in Figure 4.5 is extracted.

The Chilean government has protested Pinochet 's arrest , insisting that as a senator he was traveling on a diplomatic passport and had immunity from arrest . Cuban President Fidel Castro said Sunday he disagreed with the arrest in London of former Chilean dictator Augusto Pinochet , calling it a case of " international meddling . " Castro compared the action with the establishment in Rome in August of an International Criminal Court , a move Cuba has expressed reservations about . He expressed surprise that the British had arrested Pinochet , especially since he had provided support to England during its 1982 war with Argentina over the Falkland Islands.

Figure 4.5: Summary Extracted for the Document in Figure 4.1

4.2 Experiments

4.2.1 Evaluation Metrics

Evaluating summarization algorithms is a difficult task and it is a separate research area in Natural Language Processing. A summary's quality can be evaluated in different aspects: selected contents importance, and presentation quality. Presentation quality itself is composed of two aspects: grammatical correctness and coherence. Since we are extracting sentences from the original text, the grammatical correctness in sentences is guaranteed to be as good as the source document's grammatical correctness. Coherence in our solution is a problem as our algorithm does not consider anaphora resolution and information ordering. However, since we extract the first sentences of topic segments, anaphoric references in our summaries are low.

Since deciding what is more important in a document is a subjective task, an evaluation method is evaluation of summaries by human judges. However, comparing the contents of automaticly built summaries with human extracted summaries is a more fair methodology. Automatic evaluation is done using distributed similarity techniques. The similarity between the model summary and the system output reflects the summary quality. Recall between the system output and the model summaries is used for this reason. In the evaluation procedure, it is more appropriate to use multiple model summaries by different summarizer, since summarization is a subjective task.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [34] is the latest and most popular summarization evaluation methodology. ROUGE calculates the recall of text units using N-grams, LCS (Longest Common Subsequences) and

Weighted Longest Common Subsequences. All of these metrics are aimed to find the percentage of overlap between the system output and the model summaries. ROUGE-N score is the percentage of overlap calculated using N-grams. ROUGE-L score is calculated using LCS and ROUGE-W score is calculated using Weighted LCS.

4.2.2 Experiment Setting and Results

We have experimented with the news article corpus used in DUC2004 [4]. To properly evaluate our algorithm, and compare with existing algorithms we have attempted task 1 of DUC2004. In this task, all summarization systems provide a 75 character summary for each of the 500 articles. Each summary is automatically evaluated against 4 model summaries extracted by professional humans. While calculating ROUGE scores, words in both the model and the system output are stemmed using Porter Stemmer. Weight for calculating the WLCS is assigned as 1.2. These are the values used in DUC2004 and we have used the same values to be compatible with their evaluation. We used a newer version of ROUGE for our evaluations, thus official scores of DUC2004 and our scores may differ in small quantities.

Table 4.1 shows the scores for our system, the best system and the worst system of the 40 systems participated in DUC2004. The average score of the participants of DUC2004 is also given in this table. We also included the scores of two systems, which are also participants of DUC2004 and they also use lexical cohesion methods for summarization. Lethbridge University's [13] summarization system also attacks automated summarization problem using lexical chains. Their algorithm uses an explicit text segmenter, and after building lexical chains they score each segment using the lexical chains. From the best segments, they select sentences. This algorithm is derived from Brunn et. al.'s algorithm [11]. Another algorithm using lexical cohesion in DUC2004 is the system developed in Dublin University [57]. This system extracts phrases instead of sentences. System ranks each phrase using TFxIDF, position of word, lexical cohesion score and POS tags. They use C5.0 machine learning algorithm to classify these phrases. Their work

	ROUGE-1	**ROUGE-2**	**ROUGE-3**	**ROUGE-4**	**ROUGE-L**	**ROUGE-W**
Barzilay	0.17861	0.04381	0.01389	0.00768	0.15577	0.09508
Lethbridge	0.12135	0.02504	0.00626	0.00115	0.10852	0.06604
Dublin	0.22192	0.02543	0.00337	0.00034	0.1766	0.10169
Our System	0.19549	0.05247	0.01697	0.00531	0.17078	0.1034
Average	0.1858	0.04082	0.0111	0.00316	0.15803	0.09470
Best System	0.2511	0.06528	0.02202	0.00768	0.20109	0.11953
Worst System	0.12088	0.00731	0.0017	0.00007	0.10678	0.06564

Table 4.1: ROUGE Scores of our System and Other Participants of DUC2004

reminds our keyphrase extraction efforts.

Our implementation of Barzilay et al.'s algorithm uses our lexical chaining procedure, but uses their selection procedure. Their algorithm selects the first sentence a lexical chain member occurs in. Their sentence selection depends only on lexical chains. In their algorithm, a strong lexical chain contributes to the summary with only one sentence. They assume that a lexical chain is a topic and the first sentence is the most important sentence.

Since a lexical chaining algorithm's word sense disambiguation accuracy is as low as %63, it is possible that the first member of a lexical chain is an error. In our algorithm, lexical chains are used as an intermediate tool to find topic segments. Segments are identified by combining the cues obtained from co-occuring lexical chains. Co-occuring lexical chains may capture context specific relations and other cohesion patterns. Our segments reflect the lexical cohesion hot spots, while the whole lexical chain reflects a set of related terms that may be scattered to the whole document. We select the first sentences of the most lexically cohesive segments. We believe that our sentence selection procedure is more prone to errors in lexical chaining than Barzilay's algorithm.

4.2.3 Results

Scores of our system is promising as it is above Barzilay's algorithm. Also Lethbridge University's algorithm has obtained results below our system. System by Dublin university is above our algorithm in ROUGE-1 scores. However they have lower scores in other scores, this is mainly because their algorithm outputs phrases. In DUC2004 evaluation, stop words are not removed when calculating

	ROUGE-1	**ROUGE-2**	**ROUGE-3**	**ROUGE-4**	**ROUGE-L**	**ROUGE-W**
Barzilay	28	15	13	12	26	22
Lethbridge	41	38	35	33	41	41
Dublin	5	36	39	39	7	9
Our System	17	8	9	10	9	7

Table 4.2: ROUGE Ranks of our System and Other Participants of DUC2004

τ	**ROUGE-1**	**ROUGE-2**	**ROUGE-3**	**ROUGE-4**	**ROUGE-L**	**ROUGE-W**
0,1	0,195490	0,052470	0,016970	0,005310	0,170780	0,103400
0,2	0,177840	0,046000	0,015350	0,005290	0,156440	0,095450
0,3	0,176250	0,045260	0,015050	0,004960	0,154500	0,094350
0,4	0,170500	0,043730	0,015340	0,005150	0,150470	0,092120
0,5	0,168120	0,041470	0,014040	0,004470	0,148190	0,090630
0,6	0,164330	0,03998	0,01329	0,00426	0,14476	0,09
0,7	0,161900	0,039050	0,012870	0,004180	0,142620	0,087150
0,8	0,162320	0,039160	0,012900	0,004240	0,142720	0,087220
0,9	0,160660	0,038520	0,012570	0,004110	0,141250	0,086340
1	0,160660	0,385200	0,012570	0,004110	0,141250	0,086340

Table 4.3: ROUGE Scores for different τ values

recall. The model summaries for evaluation are formed of sentences containing stop words, for this reason their system have lower matches of sequences of words.

Table 4.2 shows the rank of each system when compared to participants of DUC2004 single document summarization task. Our system ranked in the first 10 in all of the scores except ROUGE-1 score, which is calculated using uni-grams. In overall, our system achieved very good results. These results reflect that our system has obtained competing results for the algorithms in DUC2004. Since our algorithm outperforms lexical cohesion based algorithms, such as Barzilay's algorithm, Dublin University's algorithm and Lethbridge Universities algorithm, we can consider it as a successful attempt.

4.2.3.1 Effect of τ Value and Size of the Target Summary

We have experimented with different values of τ, which is used as the lexical chain clustering stop criteria, to understand its effect in summarization. Table 4.3 shows the ROUGE scores for different values of τ. Figure 4.6 shows the graph for ROUGE-1 score for different values of τ. X-axis shows the τ values from 0.1 to 1, y-axis shows the ROUGE-1 score obtained. When τ is 1 only the lexical chains having exactly the same sentence occurrence vectors will be on the same cluster, otherwise they will be on their own clusters formed of one lexical chain.

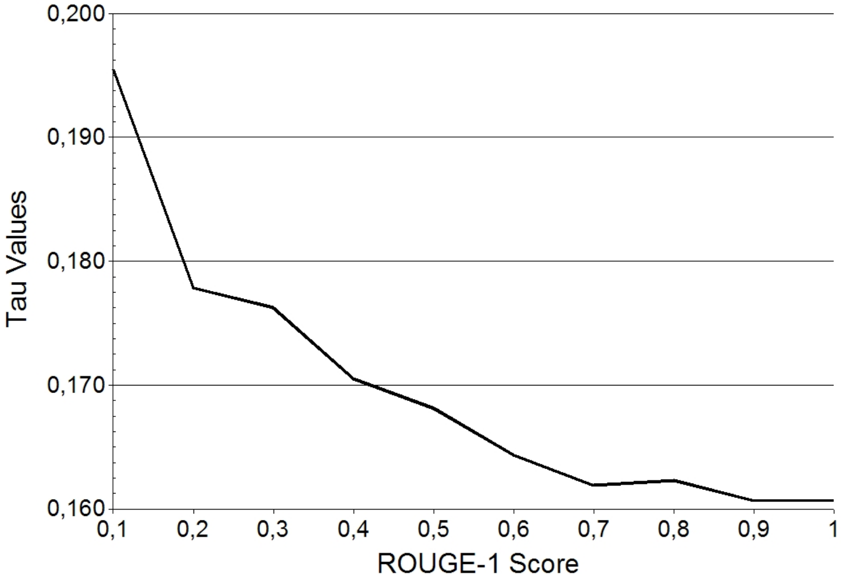


Figure 4.6: τ ROUGE-1 score graph

Experimentally we have seen that with a very small τ value our system performs better. We confirmed with all of our experiments, that lower τ values gives better results.

We also experimented with the corpus of DUC2002, to test our algorithm when the target summaries are longer. In DUC2002, single summarization task involves extracting 665 character long summaries. There were 13 systems competing for single summarization in DUC2002. Official evaluation methodology in DUC2002 was SEE score, which is a semi-automatic evaluation, requiring human evaluation. For this reason we have evaluated all the systems in DUC2002, Barzilay's selection procedure and our system using ROUGE scores. We have confirmed with this corpus that the clustering stop criteria τ must be as low as possible to achieve the best results.

	ROUGE-1	**ROUGE-2**	**ROUGE-3**	**ROUGE-4**	**ROUGE-L**	**ROUGE-W**
Barzilay	0.33579	0.19875	0.14615	0.11313	0.30856	0.17902
Our System	0.31411	0.18019	0.13247	0.10406	0.285	0.16605
Average	0.33462	0.19838	0.1454	0.1133	0.30620	0.1778
Best System	0.41156	0.26522	0.19911	0.15731	0.38062	0.22129
Worst System	0.12321	0.03064	0.01629	0.01098	0.10613	0.06083

Table 4.4: ROUGE Scores of our System and Other Participants of DUC2002

Table 4.4 presents the ROUGE scores for our system and the other participants. For the corpus used in DUC2002, our system ranked 11'th from 15 algorithms. Our algorithm performed just below Barzilay's selection procedure, which ranked 10th on DUC2002 data.

4.3 Discussion

We have experimented with two different corpora and compared our algorithm with DUC participants in 2002 and 2004. We have seen that in DUC2004, our system achieves very good results, ranking in the first 10. Our system is purely extractive, some other competing algorithms are using techniques such as: sentence reduction, anaphora resolution and elimination of repetition. In other competing algorithms, there are some systems that focus on news article domain, tracking events. Reduction of sentences could improve ROUGE score as summaries extracted are limited in size, some systems does have similar approaches. Resolving anaphora, improves the performance as model summaries does not usually contain anaphora.

We have seen that our algorithm, partitioned the document into topic segments with acceptable quality. It is possible to say that our algorithm is successful in summarization at least for the domain of news articles. We have seen two contradicting results when our algorithm is compared with Barzilay's selection procedure. In DUC2004, our algorithm has achieved better results than Barzilay's algorithm. In DUC2002, Barzilay's selection procedure has achieved better results. Our algorithm performed better than other lexical cohesion based algorithms which participated in DUC2004. This result encourages our efforts for clustering and implicit text segmentation.

Summarization evaluation is a very hard task, performance of an algorithm can change drammatically in different summarization settings. It is not possible to say that single method is best for every corpus. There are systems that incorporate different features and techniques into a single algorithm and decide which to use depending on the corpus with machine learning algorithms. Our system did perform worse than Barzilay et al.'s algorithm in DUC2002 data, a reason for this could be the size of the target summaries, but this can also depend purely on the corpus used. Length of the source texts in the corpus can effect our systems performance. With longer lexical chains, our assumption that co-occurring lexical chains are related in the given context, will fail more often as weaker semantic relations among chain members will be more often.

Chapter 5

Multiple Document Summarization

In multiple document summarization, the system has n documents about the same subject/topic as input. System output is a single summary formed of content that is extracted/generated from all input documents. This is considered as a harder problem, when compared to single document summarization. Presentation and selection procedures are more difficult. In lexical chains based summarization algorithms, merging the lexical cohesion structure of different documents is a difficult task.

5.1 Multi Document Summarization Algorithm

Each document's cohesion structure is local, so the lexical chains and sequences are also local. For this reason, we extracted lexical chains for each document separately. Merging lexical chains is not possible as the distance between two related words effect the lexical chaining algorithm. Since distance is a factor for lexical chains, information ordering in the original document is important. Combining the source documents to form a cohesive single document is a difficult task. Content planing and information ordering is essential but hard if not impossible. For this reason, we analyzed the lexical cohesion structure in documents separately.

For single document summarization, if a sequence spans more text in the document, this sequence will be more important. However, in multi document summarization, the importance of sequences and sentences occurring in different documents should be determined. After lexical chains are formed, clustering and sequencing described in the previous chapter is done. The sequences formed from different documents are gathered in a pool. These sequences are ranked with a modified scoring function, which is modified to provide a more general score for sequences.

$$Score(sequence_i) = \frac{score(Cl_i)}{max_{LC_j \in LC}(score(LC_j))} * \frac{L_i}{m} * \frac{(1 + SLC_i) * PLC_i}{f^2} \tag{5.1}$$

Equation 5.1 is derived from Equation 4.2 defined in the previous chapter. Where LC is the set of lexical chains in the document, and $max(score(LC_j))$ is the maximum lexical chain score in the document. Lexical chain score for the cluster is normalized by the maximum lexical chain score in the document. In the equation, m is the number of sentences, L_i is the length of the sequence and $score(Cl_i)$ is the average of the lexical chain scores for the cluster i. Length of the sequence L_i is normalized by the total number of sentences in the document m. As in Equation 4.2, SLC_i is the number of cluster lexical chains starting the sequence, PLC_i is the number of cluster lexical chains appearing in sequence and f is the number of lexical chains for the cluster.

All sequences extracted from all documents in the document set are ranked with the scoring function in Equation 4.2. For the top scoring sequences, their first sentences are included in the summary. We believe that, it is possible to criticise this approach, since it does not take advantage of the repetition of concepts in different documents. Importance of sequences in different documents is determined by the local lexical cohesion. This algorithm extracts sentences from the most lexical cohesive segments in each document. This approach selects the topic concentrated in each document, whether this topic is already included in the summary from another text or not.

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-W
Our System	0.14493	0.02857	0.00753	0.00344	0.1158	0.06671
Lethbridge	0.11835	0.01121	0.0020	0.0	0.09387	0.05417
Average	0.14772	0.03001	0.01105	0.0047	0.1171	0.067408
Best System	0.19297	0.05502	0.02515	0.01472	0.15964	0.09101
Worst System	0.06668	0.00227	0.0	0.0	0.05663	0.03358

Table 5.1: ROUGE Scores of our System and Other Participants of DUC2004

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-W
Our System	21	18	25	22	20	20
Lethbridge	31	34	34	34	32	33

Table 5.2: ROUGE Ranks of our System and Other Participants of DUC2004

5.2 Experiments and Evaluation

DUC2004 conference's task 2 was multi document summarization. Corpus consists of 50 set of topics consisting of 10 documents per topic. These are the same documents used in single document summarization. For each document set, 4 model summaries are given. Both model summaries and the system summaries does not exceed 665 characters. Using ROUGE evaluation we have evaluated our system's output, by comparing with other systems that participated in DUC2004. There were 35 systems in DUC2004 participating task 2 which is multi document summarization.

In Table 5.1 ROUGE scores, for the DUC2004 corpus is given. Our system has obtained average scores for ROUGE evaluation. Lethbridge University's multi document summarizer [13] also uses lexical cohesion based algorithm. For this reason, we believe that comparison between our system's results and their results is very important. We have obtained significantly better results than Lethbridge University's system.

Table 5.2 shows the obtained rank for the ROUGE evaluation among 36 systems. There are some major drawbacks for our system. Clues that can be obtained from repetition of content/topic in different documents are not exploited. Another drawback of our system is that, since our system does not have a sentence reduction module, our summaries are often cluttered with extra text which are usually eliminated in other systems. Some of the algorithms in DUC2004 take advantage of date of the news article. Model summaries tend to use more recent

> Cambodian politicians expressed hope Monday that a new partnership between the parties of strongman Hun Sen and his rival , Prince Norodom Ranariddh , in a coalition government would not end in more violence . Cambodia 's ruling party responded Tuesday to criticisms of its leader in the U.S. Congress with a lengthy defense of strongman Hun Sen 's human rights record . 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 . Cambodian leader Hun Sen has guaranteed the safety and political freedom of all politicians

Figure 5.1: Our system output for a document set from the corpus of DUC2004

> Prospects were dim for resolution of the political crisis in Cambodia in October 1998. Prime Minister Hun Sen insisted that talks take place in Cambodia while opposition leaders Ranariddh and Sam Rainsy, fearing arrest at home, wanted them abroad. King Sihanouk declined to chair talks in either place. A U.S. House resolution criticized Hun Sen's regime while the opposition tried to cut off his access to loans. But in November the King announced a coalition government with Hun Sen heading the executive and Ranariddh leading the parliament. Left out, Sam Rainsy sought the King's assurance of Hun Sen's promise of safety and freedom for all politicians.

Figure 5.2: Model Summary for a document set, summarized by a professional human for DUC2004

information. We have not taken advantage of this feature, providing a more general algorithm. Even with these disadvantages, our system obtains promising results.

Figure 5.1 presents a summary extracted by our system for a document set and Figure 5.2 presents one of the model summaries extracted by a professional human summarizer. This model is one of the four model summaries used for evaluation. ROUGE scores are measured, using the units matching with these four summaries using 6 different metrics.

Chapter 6

Keyphrase Extraction

Keyphrases are the important words/phrases that reflect the subject of the text. The term keyphrase is used in the literature to emphasize that selected terms could be phrases. Research on keyphrases has been focused on supervised algorithms and unsupervised algorithms. Turney [56], Witten et al. [58] with their algorithm called KEA and Hulth [26] describes supervised machine learning algorithms for keyphrase extraction. Turney's system and KEA uses surface level features like term frequency and position in text. The algorithm by Hulth [26] improves these algorithms by using more deeper natural language processing techniques and features, like part of speech tags.

Our algorithm tries to improve these algorithms using semantic relations and lexical cohesion. We have experimented with features derived from lexical chains and word relations.

6.1 Keyphrase Extraction Algorithm

Turney [56] introduces two supervised algorithms, a decision tree algorithm and a genetic algorithm. Both of these algorithms are using first position in text and term frequency. Frequency and term repetition are also considered as clues of lexical cohesion, but valuable information like related words and synonyms are not considered.

KEA uses term frequency times inverse document frequency (TFxIDF) attribute. TFxIDF provides a prior knowledge, familiarity of the word in the domain. KEA outperforms the results of Turney's algorithms when trained for a domain. When using TFxIDF, the corpus is used as prior knowledge to determine the familiarity of the term on the domain. We will try to provide this information using WordNet, and lexical cohesion.

Using lexical chains in summarization is very different from using it for keyphrase extraction. Scoring functions defined for summarization usually focus on the whole chain or text segments. For keyphrase extraction, we used some features that focus on members of the lexical chain rather than the whole lexical chain.

Our keyphrase extraction algorithm, considers only nouns as candidate phrases. For each noun, some set of features are extracted. We experimented with different feature sets that are acquired from lexical chains or WordNet. We used the lexical chaining algorithm described in Chapter 3 for keyphrase extraction.

In lexical chaining algorithm, each word occurrence in the text is disambiguated, and the sense of the word is guessed. After lexical chaining, our keyphrase extraction algorithm trains a keyphrase classifier. Keyphrase classifier is trained with instances of candidate keyphrases. Candidate keyphrases are indexed with their stemmed form. Stemming is done with an aggressive stemmer, called iterated Lovins stemmer. All the word/phrase occurrences in the text are grouped and folded by their stemmed forms. For example, 'conjunction' and 'conjunctive' both have the stem 'conjun'. These two different occurrences are folded to a single candidate keyphrase. Our algorithm tries to gather data from all occurrences of a word. Since we are applying one sense per word constraint, each occurence will map to the same sense. From this sense, semantic properties and relations of the word are determined. If the stem of the candidate keyphrase is equal to an author assigned keyphrase's stemmed form, the candidate keyphrase is labeled as a keyphrase. Training and testing instances are decided on the document level, if a document is reserved for training all candidate phrases in that document are training instances. Trained classifier guesses the class attribute for each candidate keyphrase in the test documents.

6.1.1 Features Used in Keyphrase Extraction

With machine learning techniques, we have investigated the statistical properties of keyphrases especially their lexical cohesion properties. We think that, keyphrases are words that have more relations with other words and text segments. For a folded candidate keyphrase, features that we have investigated are extracted and used to train our keyphrase classifier.

For a candidate keyphrase, we have a set of occurrences in the text. Senses are assigned to all occurrences of the word. In Chapter 3, we have defined word sense graph for texts. Lexical chaining algorithm finds the best interpretation for the text, and the word sense graph for the text can be built. For any candidate keyphrase occurrence, it is possible to find related words, its lexical chain and the assigned sense by consulting the word sense graph.

Given a word sense graph, it is possible to extract our lexical chain based features. We will be explaining our features, using the word sense graph and subgraphs of the word sense graph. We will refer to the word sense graph, found after lexical chaining algorithm, as G . For a candidate keyphrase, if the sense is known, there will be occurrences of that sense. Each occurrence is a node in G. If ws_i is one of those occurrences, then the graph composed of nodes that are connected to ws_i, is the lexical chain that ws_i is a member of. We will refer to the graph of the lexical chain ws_i is member of as G_{LC_i}.

In G_{LC_i}, there will be senses that are not directly related to ws_i, but connected to ws_i by other sense occurrences with a path greater than 1. For a word sense in G_{LC_i}, all member word senses will have the same properties. We believe that features extracted from G_{LC_i} are importance clues for the topic in general. For

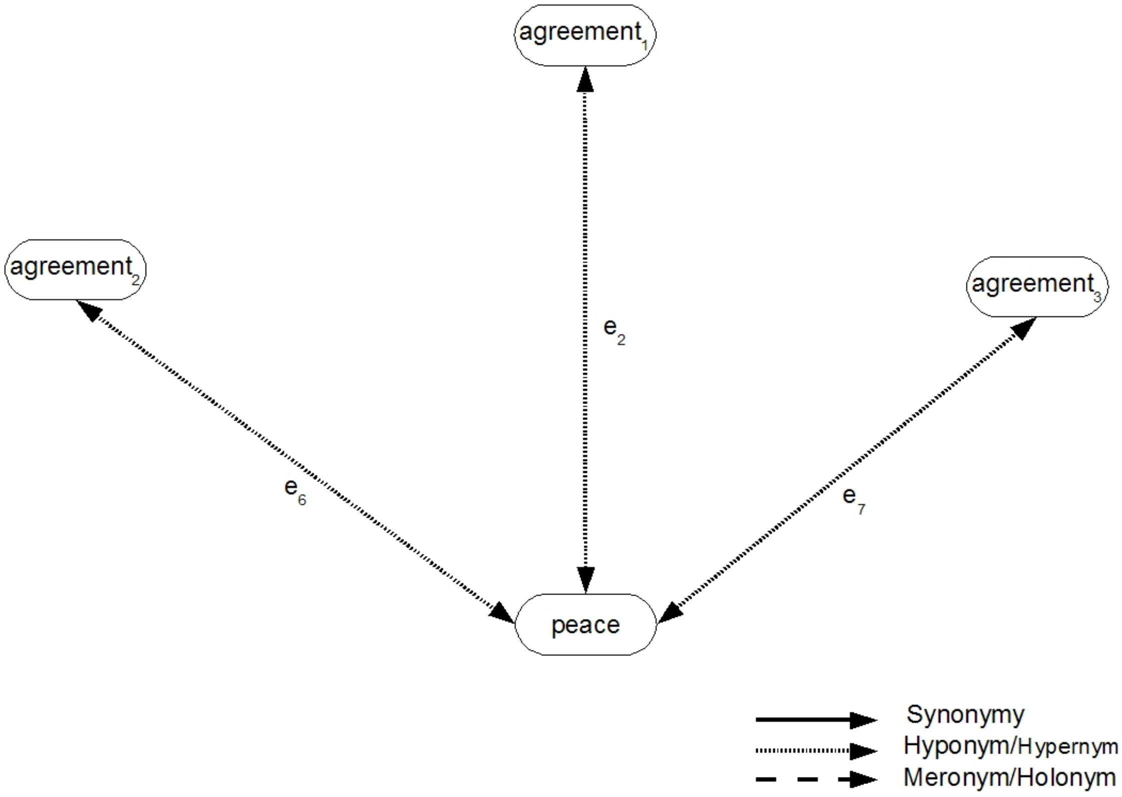


Figure 6.1: Graph Gw_i for word 'peace'

the word sense occurrence ws_i, we have investigated the properties of directly related senses to ws_i. All related words to ws_i will be in a graph composed of nodes that are connected to ws_i by a path length of 1, we will refer to this graph as Gw_i. We believe that features extracted from Gw_i are more local features, they reflect the importance of the phrase within the topic. Note that for a candidate keyphrase, there are multiple occurrences of the word. Different occurrences will have different lexical cohesion properties. In our experiments we have used the maximum scores obtained from these occurrences.

For the sake of clarity, our algorithm is demonstrated using the example text given in Figure 3.1. The graph G_{LC_i} for word 'peace'is given in Figure 3.3. The Graph Gw_i for 'peace' is given in Figure 6.1. G_{LC_i} graph for all senses in the same graph/lexical chain is the same, but Gw_i is different for each occurence.

We have experimented with different features, that could be acquired from

WordNet or lexical chains. The features described below yields the best results for keyphrase extraction.

- Term Frequency : Frequency is the number of occurences of the candidate phrase. Frequency is calculated using an aggressive stemming algorithm (iterated Lovins stemmer). Frequency is normalized by the number of noun occurrences in the document. This feature has been used in other keyphrase extraction algorithms too. We discuss its contribution to the accuracy of the algorithm. Since repetition itself is a lexical cohesion type, theoretically this feature is part of other features that we use.
- First Occurrence in Text : This feature is the first occurrence of the word in text. We use sentence indexes for this feature. More specifically, it is the index of the sentence, where the term first occurs. This feature is normalized by the total number of sentences in the document. It is 1 for 'agreement' in the example in Figure 3.1.
- Last Occurrence in Text : This feature is the last occurrence of the word in text. We use sentence indexes for this feature. More specifically, it is the index of the sentence, where the term last occurs. This feature is normalized by the total number of sentences in the document. It is 6 for 'agreement' in the example in Figure 3.1.
- Semantic Relation Score : Semantic relation score is the total strength of all relations of a sense occurrence. Basically it is the sum of all edge weights in G_{LC_i}, using the edge weights in Table 3.1. This score will be the same for two senses that are members of the same lexical chain. Since we have an one sense per word constraint in our algorithm, a candidate phrase will have the same score for different occurrences. We tried normalizing this feature by the maximum lexical chain score in that document, but best results are obtained without using normalization. In our example, this score is the sum of all edge weights in Figure 3.3. For all of the nodes in the graph this score is the same.
- Direct Semantic Relation Score : This feature is calculated by summing all

the edges in Gw_i. A word can have more than one sense node, these values are very similar in quantities. We selected the maximum scoring sense node, in case of multiple sense nodes. We tried normalizing this feature, but best results are obtained without using normalization. In our example, sum of edges e_2, e_6 and e_7 is the score for 'peace'. Note that 'agreement' has 3 occurrences. In this case, 'agreement$_3$' has a score calculated by using edges, e_{10}, e_7, e_4, e_1 and e_3. However 'agreement$_2$' has a score calculated by using edges, e_8, e_1, e_6 and e_5. Remember that edge weights are dependent on the distance between two related words. Even though the relation types are same, they have different scores. 'agreement$_3$' has an extra edge e_{10} since 'promulgation' is too far away from other instances of 'agreement'. The maximum score among these different occurences is used.

- Lexical chain span : The span score of a lexical chain depends on the portion of the text that is covered by the lexical chain. Let our word w be a member of lexical chain LC_i. This score is the lexical chain's span in text. It is simply; sentence index of the last occurrence of LC_i member - sentence index of the first occurrence of LC_i member. While this score reflects for how long this topic has been discussed, connectivity in the span area is not considered. We have seen that keyphrases mostly have high percentage of text coverage. This feature is normalized by the number of sentences in the text.

- Direct Lexical chain span : maximum sentence index - minimum sentence index of a node in Gw_i. This is the number of sentences spanned by related words to ws_i. Maximum span value among different occurences of the candidate phrase is used. This feature is normalized by total number of sentences in the document.

- Hyponym/Hypernym Hierarchy Level: This feature reflects the hierarchy level of a sense in Hypernym/Hyponym hierarchy. Let P_r be the longest path from the sense node in WordNet to a top element in WordNet's hypernym/hyponym hierarchy. Let P_l be the longest path from the sense node in WordNet to a leaf element in hypernym/hyponym hierarchy. It is calculated by Equation 6.1

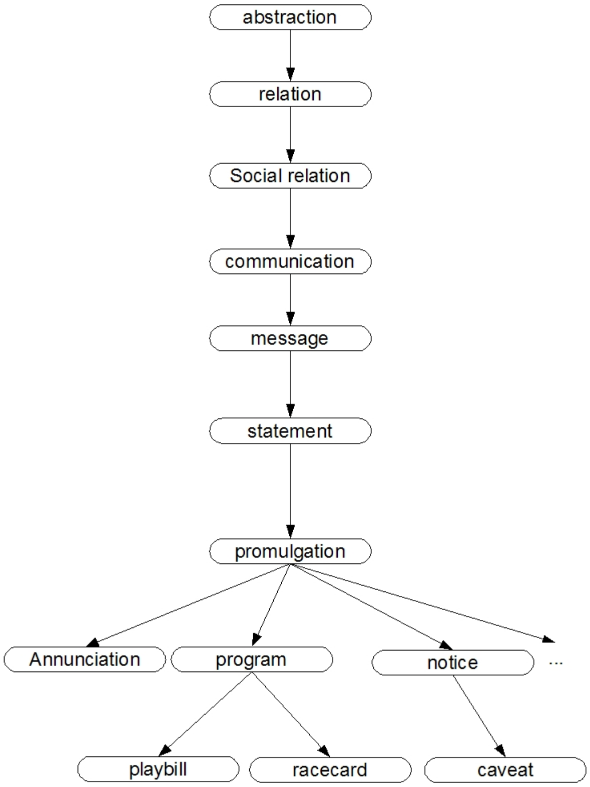


Figure 6.2: Hypernym/Hyponym Hierarchy for ”promulgation”

$$Level = P_r/(P_l + P_r) \tag{6.1}$$

Figure 6.2 shows the Hypernym/Hyponym hierarchy for promulgation. P_r for ’*promulgation*’ is 6 and P_l is 2, so level of ’promulgation’ is 0.75.

- Sentence Count : This is the total number of sentences where LC_i has members in. This differs from the span, since in a span there might be discontinuity in sentence occurrences, lexical chain may not have a member in all sentences between its first occurence and the last. This feature is basically the total number of sentences where the lexical chain LC_i has a member in. We normalized this feature using the total number of sentences in the document.

- Direct Sentence Count : This is the total number of sentences nodes of

Gw_i occurs in. This differs from the direct semantic span feature since in a span there might be discontinuity of occurrences of members. This feature is the total number of sentences the word ws_i or a related word occurs. If ws_i has multiple occurrences, the maximum sentence count value is used. We normalized this feature using the total number of sentences in the document.

6.1.2 Learning to extract keyphrases

Machine learning algorithms are used in Keyphrase Extraction by Turney [56] and Witten et al. (KEA algorithm) ([58]). KEA algorithm uses Naive Bayes learning. Turney experimented with a genetic algorithm and a decision tree algorithm. We have experimented with both decision tree and Naive Bayes algorithm, but obtained better results with Naive Bayes algorithm.

We used Naive Bayes algorithm to extract keyphrases. For comparison, as a baseline algorithm we trained a Naive Bayes classifier with features similar to Turney's decision tree experiment. We were able to compare our results with KEA, as the implementation for KEA is publicly available.

Class attribute for keyphrase extraction has two values. Phrase is a keyphrase or it is not a keyphrase. When classifying a keyphrase, the candidate phrase's stem is compared to the author assigned keyphrases stem. For example 'bird' and 'birds' are considered as a match.

Distribution of classes in keyphrase extraction is not balanced. Number of author assigned keyphrases are relatively very small compared to number of candidate phrases in the document. In machine learning algorithms, oversampling or under sampling can give better results. Oversampling is the process of cloning instances of the less populated class. Under sampling is just the opposite, ignoring instances of the more populated class. Turney has shown experimentally that either of these techniques improve the performance of the algorithm. Our experiments confirm with Turney's experiments.

Turney also reports that bagging improves the performance of the decision tree algorithm for keyphrase extraction. Bagging is the process of classifying with multiple classifiers. In bagging, an instance is classified with multiple classifiers trained with different data, and the average classification probability is used to classify the instance. Bagging decreases the variance, increasing the accuracy. For keyphrase extraction, soft thresholds are used. Instead of classifying each instance as true or false, probability of being a keyphrase is used. With soft-thresholds it is possible to have as many keyphrases as possible. In our experiments we have seen that keyphrases are not classified with high probabilities. Thus, for extracting larger number of keyphrases for a document using soft thresholds is a must.

6.2 Experiments and Evaluation

We compared our algorithm with KEA algorithm and a baseline algorithm which is very similar to the decision tree algorithm used by Turney. KEA and Turney's algorithm considers all possible noun phrase combinations, our implementation depends on a noun phrase skimmer and considers only nouns classified by a POS tagger. Base algorithm, which uses the same POS tagger and Noun phrase skimmer, but uses only first occurrence and frequency count features. This base algorithm differs from Turney's algorithm, since it relies on a POS tagger to find the noun phrases and Turney tries all combinations. Accuracy of the base algorithm and our algorithm reflects the performance of lexical chain and WordNet based features.

In our corpus consisting of 75 journal papers, average number of words per document is 11008.75. About %27.11 of these words are detected as nouns, resulting an average of 2991.98 nouns per document. For this corpus using iterated Lovins stemmer, noun phrase skimmer and POS (Part Of Speech) tagging, an average of 1292.06 unique candidate phrases are found. So, we have 1292.06 instances used in our machine learning algorithm. A very small percentage of these instances are keyphrases, %0.0042 of the candidate phrases are keyphrases.

culture
evolution
group selection
kin selection
inclusive fitness
natural selection
reciprocity
social organization
units of selection

(a) Author Assigned

groups
group selection
selection
evolutionists
individuals
natural selection
Wilson
group level
genes
biology
units
populations
example
Williams
kin selection

(b) Automatically Extracted

Figure 6.3: Author Assigned and Extracted Keyphrases

There are 5.43 keyphrase instances per document. About 450.5 of the nouns in each document could not be found in WordNet that is %16 of total candidates. For these instances, the features except first position, last position and frequency are left as missing attributes.

Figure 6.3 shows the author assigned keyphrases and output of our system for a document from our corpus. The output contains 15 phrases, these are the most probable keyphrases classified by our keyphrase classifier. The bold keyphrases are correct guesses, there are 4 correctly classified keyphrases for this document.

Table 6.1 reflects, the number of correctly classified keywords per document by the algorithms, when 5, 10, 15 keyphrases are extracted. We have processed our corpus with KEA algortithm for comparison. To select the best features we have experimented with all combinations of the features. We will present some interesting results from these combinations. Our base algorithm uses only frequency and first position in text. All without frequency, uses all the features explained above except the frequency, we believe that since word repetition is a lexical cohesion type, this combination should behave as good as the base algorithm. To compare the effect of hyponym level we experimented without using this feature. To compare relation score and span features, we experimented with two sets of features. The first set uses all features except two span features and sentence counts, we will refer to this algorithm as Score Features. The second set uses all features except relation scores and sentence counts, we will refer to

Cutoff KP/Doc	**5**	**10**	**15**
KEA Algorithm	1.08	1.82	2.25
Base Algorithm	0.78	1.33	1.63
All Without Frequency	0.62	0.92	1.19
All without Hyponym Level	0.74	1.33	1.62
Score Feature	0.90	1.29	1.76
Span Features	0.80	1.29	1.62
All features	0.94	1.392	1.74

Table 6.1: Keyphrase Results in correct keyphrases per document

this algorithm as Span Features. When the base algorithm and our algorithm is compared, we can see that lexical cohesion based features improve the accuracy. Also it is possible to see that lexical cohesion based features along with first position performs as good as the base algorithm. We experimented with different combinations of these and more features, and have seen that the best accuracy is obtained using all the features described. Through our experiments we have seen that relation scores, are better features than span features for keyphrase extraction.

6.2.1 Learning to Extract Words in Keyphrases

In keyphrase extraction, lexical cohesion features did not provide significant improvement to the results. One reason for these results is the handling of noun phrases. Noun phrases with the same head noun will have very similar lexical cohesion values. To investigate the effect of this problem, we have experimented with an algorithm which extracts words that appear in keyphrases. In this algorithm, we try to extract words that appear in keyphrases. For example if ’conjoint analysis’ is a keyphrase, both ’conjoint’ and ’analysis’ are classified as keyword.

Feature calculations are the same for keyphrase extraction. We used the same feature sets for this algorithm. Only difference in this algorithm is the class of instances. Class of a word is ’true’ if the stemmed word occurs in a keyphrase, otherwise ’false’. We have used the same machine learning algorithm, which is

Cutoff KW/Doc	**5**	**10**	**15**
Base Algorithm	1.86	2.73	3.21
All features	2.21	2.73	3.26

Table 6.2: Words that appear in Keyphrases Accuracy

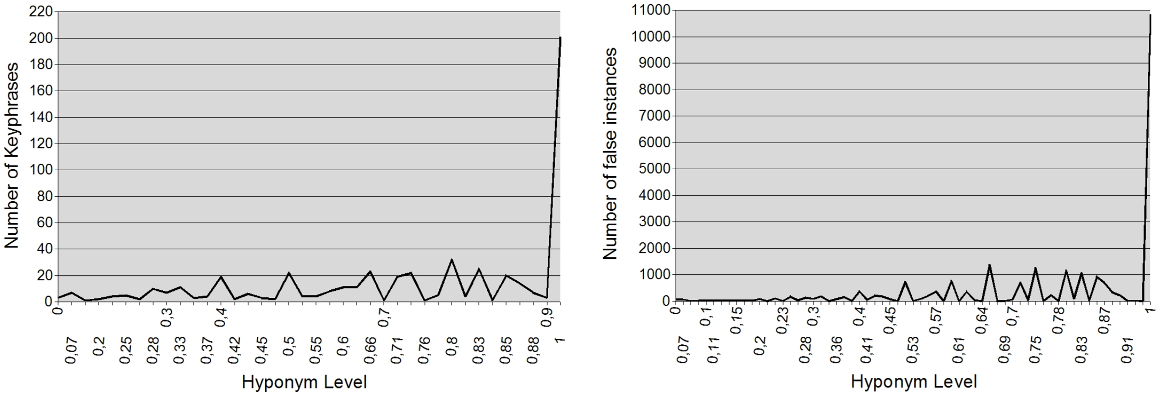
Naive Bayes algorithm.

Table 6.2 presents number of correctly classified keyphrases for a document, when 5, 10, 15 words are extracted. Since KEA and Turney attempts to extract full phrases, it is not possible to compare these results with their results. We have compared our algorithm with a baseline algorithm which uses only frequency and first position features, which are used by Turney. Lexical cohesion features have improved our results slightly.

6.3 Discussion

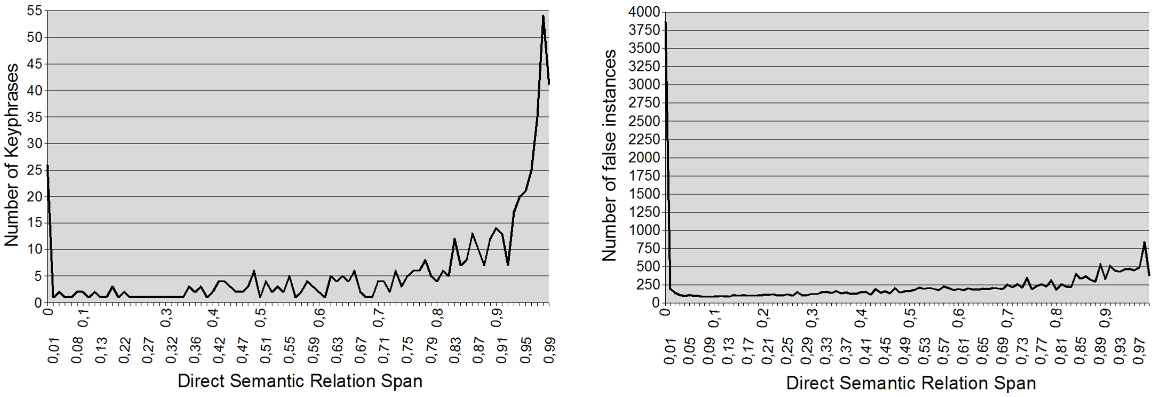
For two of our features, hyponym/hpyernym level and direct semantic score, distribution of their values are plotted. It is seen from Figure 6.4(a) that keyphrases tend to have a hyponym level of 1, which means that keyphrases are mostly more specific words. However, it is seen from Figure 6.4(b) that false instances tend to have a similar histogram. We were hoping that hyponym level would provide extra knowledge that is acquired by using TFxIDF in KEA algorithm, the familarity of the word in the domain. It is seen that in documents usually more specific words are used.

Lexical semantic span feature is a more distinguishing feature. The keyphrase histogram for the feature direct semantic span is shown in Figure 6.5(a). It can be seen that keyphrases tend to cover more semantic space than other words. When we compare the histograms in Figure 6.5(a) and Figure 6.5(b) we can see that direct lexical span feature for keyphrases are usually above 0.7. That is a keyphrase has semantic relations with more than %70 of the document.



(a) Hyponym Level Keyphrase Distribution (b) Hyponym Level False instances Distribution

Figure 6.4: Hyponym Level Distribution



(a) Keyphrases Direct Semantic Span Distribution (b) Non-Keyphrases Direct Semantic Span Distribution

Figure 6.5: Direct Semantical Span Distribution

These results are below our expectations, lexical chains have provided significant accuracy gains in summarization and we were expecting significant improvements in keyphrase extraction. Lexical chains provide usually more information about a segment or sentences. This work differs from previous research in the sense that we try to exploit lexical chains for detecting the importance for phrases. The first problem with lexical chains is that the semantic lexical chain score for a word will be same for any word in the same lexical chain. Direct semantic lexical chain score will be very similar.

Most of the keyphrases are noun phrases. English is a very productive language for noun phrases and most of noun phrases do not appear in WordNet. In lexical chaining process when a phrase does not appear in WordNet, its head noun's relations are used. For this reason lexical chain features for two distinct phrases 'conjoint analysis' and 'geographic analysis' are treated as the same and all the features will be much or less same.

When feature value distributions are observed it is seen that, false instances has a similar feature value distribution. A lexical chain can have thousands of nodes, so the discriminative properties of lexical cohesion are lower. Two different senses belonging to the same lexical chain will have very similar values. Some other features that can be extracted from lexical chains, that focus on word instances are needed.

We have seen that lexical cohesion features, improve our baseline algorithm. However, KEA algorithm still performs better than our algorithm. Our motivation for implementing this algorithm, was to provide the clue obtained from corpus using TFxIDF in KEA using WordNet. Using WordNet as a knowledge base is a more general approach then using TFxIDF. We expect our algorithm to be more domain independent than KEA. Unfortunately we were not able to prove this using different corpora from different domains, as we were unable to prepare enough documents with keyphrases from different domains to prove our point.

We were able to improve the baseline algorithm using lexical cohesion and WordNet based features. With these results, we have shown that lexical cohesion features can improve the accuracy of keyphrase extraction.

Chapter 7

Implementation Details

This chapter describes the essential elements and components used in our algorithms in detail. We implemented the algorithm using Java. Figure 7.1 shows the components of our algorithms. Details of these components are described in this chapter.

7.1 Sentence Detector and Part of Speech Tagger

Sentence detector used in our algorithms uses two heuristics to identify sentences. Punctuations in the text are exploited to detect sentence boundaries. {., !, ?} are used to determine sentence boundaries. However a naive sentence boundary detector can find wrong boundaries with the use of abbreviations. For example 'Dr.Kenny specializes in neurosurgery.'can be detected as two sentences 'Dr.' and 'Kenny specializes in neurosurgery.'. To overcome this problem our sentence detector uses the length of sentence as a second heuristic. If the number of words in a sentence is below some threshold value, the punctuation is ignored and the sentence boundary is detected. The threshold value we have used is 4 words per sentence.

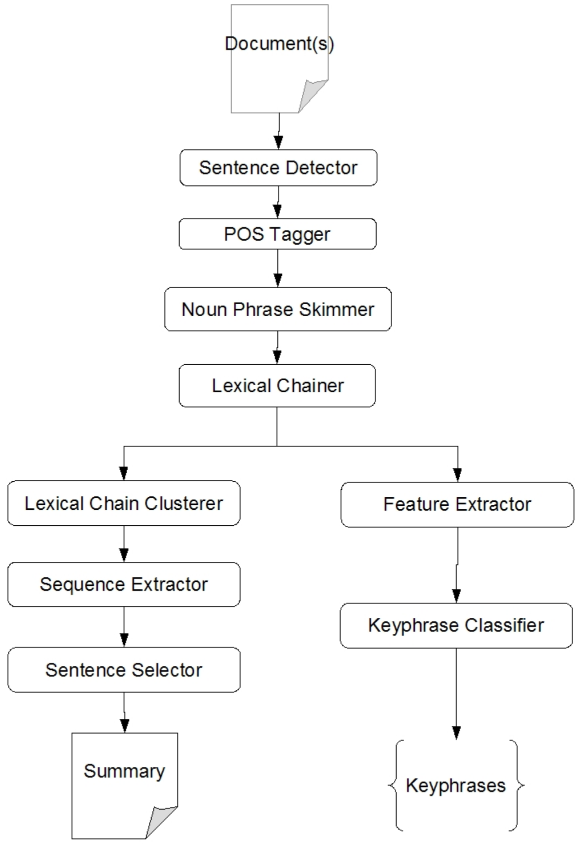


Figure 7.1: General System Architecture and Components

Part of speech (POS) tagging is the process of assigning part of speech tags to each word in a sentence. A POS tagger should guess the correct POS of the seen word with a high accuracy. POS tagging is an important component of our system. Lexical chains are built using the nouns in the document. In keyphrase extraction, nouns are our candidate phrases. Noun phrase skimmer which we describe in the next section, depends highly on part of speech tags. WordNet stores words with their POS tags. The verb 'screening' is different from the noun 'screening', so to determine the correct semantic relations in a document correct POS tags should be identified.

We used Stanford University's Maxent POS Tagger for these purposes [55]. This tagger annotates the given text using maximum entropy models. Details of the POS tagger is out of the scope of this thesis. The accuracy of the tagger is reported to be %96.76 and we have chosen to use this tagger for its availability and accuracy.

7.2 Noun Phrase Skimmer

Accurate detection of noun phrases is important. English is a very productive language for compound nouns. WordNet covers some phrases in English but especially domain specific compound nouns are missing in WordNet. To build lexical chains more accurately, compound nouns that exist in WordNet should be identified. The output of the keyphrase extractor are phrases. The phrases assigned by the author should be identified.

Normally to detect noun phrases, a parser is used in NLP applications. However, we believe that using a full parser for our task is not appropriate. A full parser's efficiency and accuracy does not overlap with our interests. Instead, we decided to build a simple noun phrase skimmer, which uses the POS tags. Noun phrases usually ends with a head noun. This head noun is accompanied by zero or more pre-modifiers, which usually are nouns or adjectives. For this reason, we built a noun phrase skimmer which parses the POS tags for the grammar given below.

$NP \rightarrow (PM) * N$

$PM \rightarrow$ (N|J)

where NP is noun phrase, N is noun, PM is pre-modifier and J is adjective.

7.3 WordNet issues

WordNet is formed up of hierarchies. The generalization/specialization hierarchy in WordNet is a graph consisting of many nodes. Checking for hypernym/hyponym relations for a sense is done through building the whole hierarchy for the word. Part/Whole relations depend on this hierarchy also. The meronyms are inherited by hyponyms of a word. The same applies to holonyms of the word. For example the word ’dog’ inherits its meronym ‘paw’ from its hypernym ’canine’

Hyponym/hypernym hierarchy of a word is a subgraph of WordNet. This subgraph can be huge for some words. For lexical chaining, all of the relations of the sense with other senses are querried for each sense. Most of the execution time in building lexical chains is spent on looking up relations between senses. The word ’entity’ is one of the root words in WordNet. This word has a huge hyponym/hypernym hierarchy. A great portion of the words in WordNet is a hyponym of ’entity’. Gathering all the relationships for top level words takes much more time.

For this reason, we implemented our own WordNet API. First we transfered the data in WordNet to a Relational Database Management System (RDBMS). We indexed all of the relations that we are interested in, as a flat relation table. This table allowed us to build lexical chains in linear time. Checking the relation between two senses is a matter of single database lookup. This reduced the running time significantly.

In our WordNet API, we have implemented a caching strategy to speed up lookups. In lexical chaining algorithm, all combinations of senses in the document are queried. For this reason we cached the relations when processing long documents and in multi document summarization.

Nouns in WordNet are stored in their base forms. For this reason a stemming algorithm is required. WordNet contains an exception list for nouns that do not obey the general suffix and prefix structures. Using this exception list, we have implemented a noun stemmer. Our noun stemmer uses simple heuristics to find singular forms of words. A given word is first checked if it is contained as it is. If the word is not found in WordNet, then exception list is consulted. If word is not in exception list then regular stems are removed from the word. We used simple rules to stem the words. These rules are in Table 7.1.

Plural Suffix	Singular Suffix
ies	y
ses	s
xes	x
zes	z
ches	ch
shes	sh
men	man
sis	
s	

Table 7.1: Stemming Rules

7.4 Lexical Chaining Algorithm

We have implemented the algorithm described by Galley et al. [21]. Algorithm is composed of two phases. In the first phase all senses for a word is found and all relations for the senses are found, pseudocode for this phase is given in Figure 7.2. After this phase, a relation matrix formed of all senses in the document, and relation between these senses are found. Each word knows occurences and its senses. Each sense knows its words. Since a sense can be mapped to different words (synonyms), senses maintain a list of words.

Second phase involves disambiguating the words. This is done by finding each word's most strongly related sense. When relation scores are calculated, occurences of the senses are needed. While the edge weight for a hypernym relation with a distance of 1 is 1, edge weight for a hypernym relation with a distance of 10 sentences is 0.5. For each sense, all the relations and all the occurences are evaluated to find the total score of the sense. Maximum scoring sense is selected as the correct sense. After disambiguating all the words, finding the lexical chains is easy. Graphs formed of word occurences connected with semantic relations are the lexical chains. Lexical chains are connected graphs. For this reason we are keeping track of word senses.

```
Algorithm 7.4.1: buildWordDisambugiationGraph($words$)

global $wordsList, senseRelationMatrix$
for each $noun \in words$
  if $noun \in wordsList$
    then
      $wordNode \leftarrow$ findWordNode($noun$)
      addOccurenceToWordNode($noun$)
    else
      addNounToWordsList($noun$)
      $wordNode \leftarrow$ findWordNode($noun$)
      addOccurencesToSenses($wordNode, noun$)
procedure addOccurencesToSense($wordNode, noun$)
  for each $sense \in noun.senses$
    if $wordNode \notin sense.wordNodes$
      then
    addOccurence($wordNode, sense$)

procedure addNounToWordsList($noun$)
  for each $sense \in noun.senses$
    if $sense \notin senseRelationMatrix$
      then
    addSense($sense, senseRelationMatrix$)
    for each $otherSense \in senseRelationMatrix$
      $relation \leftarrow$ findRelation($sense, otherSense$)
      addRelation($relation, senseRelationMatrix$)
      $symetricOfRelation \leftarrow$ getSymettricOfRelation($relation$)
      addRelation($symetricOfRelation, senseRelationMatrix$)
```

Figure 7.2: Pseudocode for Building the Word Sense Disambiguation Graph

7.5 Keyphrase Extraction

In keyphrase extraction, when calculating the frequency of words and classifying instances of words we used an aggressive stemming algorithm. Lovins algorithm is based on simple heuristics for English [27]. Iterated Lovins algorithm, iteratively stems the given word with Lovins algorithm until there is no change in the output. In our algorithm through WordNet and using heuristics it is possible to find the correct stem of words. However in GENEX [56] and KEA algorithms the classification procedure is done through this algorithm. It would be a disadvantage to use the correct stems of words, for this reason we used this stemmer in our matching processes.

For the machine learning algorithms, we used WEKA machine learning Java library. WEKA [15] contains many machine learning algorithms. We have experimented in keyphrase extraction with C4.5 implementation and Naive Bayes algorithm.

Chapter 8

Conclusion and Future Work

We have attacked single document summarization, multi document summarization and keyphrase extraction problems. These are very similar problems. For single document summarization, our algorithm is able to select sentences that human summarizers prefer to add to their summaries. Our multi document summarizer derived from the single document summarizer with small modifications is able to achieve good results. It is able to select most cohesive sentences from different documents about the same topic. Our keyphrase extraction algorithm is below current state of the art keyphrase extraction algorithms, but we experimentally proved that lexical cohesion based features improve classification. Instead of using TFxIDF which is a domain dependent feature, our algorithm relies on WordNet which is theoretically domain independent. Unfortunately, we could not find corpora to claim that our algorithm is more domain independent. This is left as a future work.

For summarization, we aimed to use more cohesion clues than other lexical chain based summarization algorithms. Our results were competitive with other summarization algorithms and achieved good results. Using co-occurrence of lexical chain members, our algorithm tries to build the bond between subject terms and the object terms in the text. With implicit segmentation, we tried to take advantage of lexical chains for text segmentation. It might be possible to use our algorithm as a text segmenter.

In keyphrase extraction, we expected important phrases to have more relations with other words in the text. Lexical chains somehow reflect topics in the text. We have seen that lexical chains that the keyphrases are a member of, span more text. Although it depends on the domain, keyphrases are expected to be more specific words. We tried to find how specific a word is using WordNet. We have seen that large number of non-keyphrases had similar distributions. This is mostly because lexical chains are not focused on words but on topics. All words in a lexical chain have similar lexical cohesion values.

In overall, our system obtained promising results. Lexical chains are easy to identify structures that can capture the relevance of the text. However lexical chains are far away from full understanding of the text. They can only detect the used words that are related with each other and they do not provide the link between actors, places and other objects. We tried to capture these links through co-occurence analysis.

8.1 Future Work

To prove that our features are more domain independent than TFxIDF, experiments involving different corpora must be conducted. In summarization, different genre documents formed of longer documents, could yield interesting results. Gathering corpora for these experiments is very hard, as keyphrases should be assigned to documents or documents should be summarized.

Available lexical chaining algorithms are not very efficient with respect to running time. Word sense disambiguation accuracy is low. An alternative approach could be separating the word sense disambiguation phase from lexical chains. There are some word sense disambiguation algorithms with different techniques. Using these algorithms to form a word sense graph for documents and then finding the lexical chains could result in better performance both in terms of accuracy and computation time.

Our summarization algorithm depends on text segmentation, for this reason our algorithm has a text segmenter. This segmenter should be evaluated for its performance in text segmentation.

Keyphrases extracted by our algorithm could be used in search engines. Using our algorithm, words that will be indexed could be filtered, lowering resources needed by indexing algorithms. Lexical cohesion features could be used in different problems such as text categorization.

Our algorithm is currently for English, but it is possible to convert this algorithm to different languages. The language dependency of this algorithm is mainly caused by WordNet. When Turkish WordNet is available it will be possible to use this algorithm for Turkish texts. Only WordNet, Part of speech tagger and stemming rules are language dependent.

Appendix A

Example Summaries

A.1 News Article 1

A.1.1 Article

Bulent Ecevit, who was asked to form a new government Wednesday, is a former prime minister best remembered for ordering an invasion of Cyprus in 1974 that made him an overnight hero at home. The invasion, after a short-lived coup by supporters of union with Greece, has led to the division of the island. Throughout the years, Ecevit, 73, has remained a strong defender of the cause of the Turkish Cypriots "As long as Turkey lives, we won't allow the oppression and subordination of Turkish Cypriots at the hands of Greek Cypriots," he said in July 1997 during the 23rd anniversary celebrations of the invasion. Ecevit, who was prime minister three times since 1974, has over the years shed some of the socialist idealism he was known for in the 70s. During his tenure as deputy prime minister in a 17-month government that was toppled last week over a corruption scandal, he gave his backing to the liberal policies of the center-right-led coalition. He often said he was carrying out a duty to bring a stable government and spare Turkey from crisis - a reference to tensions between a previous Islamic-led government and the secular military. Though never a Marxist, Ecevit was in his early years viewed with suspicion by big business for espousing socialism based on heavy government social benefits and a strong role for the state sector in the economy. Recently, however, he has helped the government keep on good terms with the IMF, which ordered a strict curb on public spending, and approved a number of state sell-offs. Under his leadership in the 70's, ties with the United States were tense. He has also expressed concern over a U.S.-led multinational force based in Turkey that monitors a no-fly zone over Kurdish-controlled northern Iraq. He argues it it is helping create a Kurdish state. His frequent visits to Iraq to meet with President Saddam Hussein have in turn raised suspicion in Washington. Despite a short alliance with an Islamic party in 1974, he is a staunch defender of Turkey's secular traditions and pushed for a crackdown on Islamic radicalism. Ecevit was born in Istanbul in 1925, to an intellectual family and studied literature at a prestigious American-run high school. He has taken some courses at Harvard University. A former journalist, he entered politics in 1957, rising to the leadership of the Republican People's Party in 1972, becoming prime minister in 1974, briefly in 1977 and again in 1978-79. He was barred from politics in the years that followed a 1980 military coup. He was imprisoned three times for carrying on with political activities despite the ban, mainly through his wife of 51 years, Rahsan, who formed the Democratic Left Party in 1985 and led it until a democratic reform in 1987 allowed Ecevit back into politics. In corruption-tainted Turkish politics, he remains known as the leader with the cleanest slate. Not even his alliance with Yilmaz who was ousted for alleged ties to the mob and rigging the privatization of a bank, tarnished his image.

A.1.2 Summary

Though never a Marxist, Ecevit was in his early years viewed with suspicion by big business for espousing socialism based on heavy government social benefits and a strong role for the state sector in the economy. He has also expressed concern over a U.S.-led multinational force based in Turkey that monitors a no-fly zone over Kurdish-controlled northern Iraq. His frequent visits to Iraq to meet with President Saddam Hussein have in turn raised suspicion in Washington. Bulent

Ecevit, who was asked to form a new government Wednesday, is a former prime minister best remembered for ordering an invasion of Cyprus in 1974 that made him an overnight hero

A.2 News Article 2

A.2.1 Article

As labor battles go, the current one between the National Basketball Association and its players is weird even by sports standards. There is a real possibility that most, if not all, of the coming season will be canceled. In this union battle it is the interests of the best paid, not those who make union scale, that are dominating the discussion. And here it is some of the workers, not the management, who are considering trying to make the union disappear. The current arrangement has produced an unbalanced pay scale of immense proportions. Last year more players than ever before received the union minimum, then $242,000 for rookies or $272,000 for veterans. The number of players making $1 million to $2 million a year the middle class, in NBA terms fell sharply. But Michael Jordan made $33 million. This should not be a surprise. Sports is an entertainment business, not unlike movies. Big stars get millions, while most get union scale. Over the years, NBA efforts to stem the rise of salaries have failed. The most important loophole in its salary cap lets a team sign its own free agent for whatever it is willing to pay. When that was adopted, it was assumed that no team would pay a lot more than a rival could pay. But it has not worked out that way. In the current negotiation, the league has offered to guarantee that its payroll will rise 20 percent over the next four years, from $1 billion to $1.2 billion, and says it is open to proposals to split that money any way the players want, whether by raising the minimum salary or guaranteeing raises for veterans. The union says it is worried about that middle class, but seems determined to preserve the free market. The league got its broadcasters, NBC and Time Warner's cable channels, to agree to pay this year's television fees whether or not there are any games to broadcast. (They will be paid back in later years, either through reduced fees or extra games to show.) Owners hoped the players would think management was willing to wait them out, and come to terms with only a small part of the season canceled. But the union is acting unhurried. It turned aside requests for negotiations this week, saying the players had to meet first. Then there is the issue of union suicide, a tactic that was rejected by the players in 1995. The idea is that if the players had no union, it would be illegal under antitrust laws for the owners to collude. The sky would be the limit. That tactic might fail. The courts could reject a union decertification vote as a sham, and in any case some players may fear that teams would feel free to offer less than the old union minimum. But if the players go that route, it could be a long time before real negotiations get going. Billy Hunter, the union's executive director, warned the owners this week that a prolonged lockout could destroy the league's popularity. That was what all the seers said four years ago, when baseball's World Series was canceled by labor troubles. But fan memories are relatively short, and now baseball seems more popular than ever. With that in mind, both owners and players may choose to battle on for months.

A.2.2 Summary

In this union battle it is the interests of the best paid, not those who make union scale, that are dominating the discussion. The current arrangement has produced an unbalanced pay scale of immense proportions. Big stars get millions, while most get union scale. As labor battles go, the current one between the National Basketball Association and its players is weird even by sports standards. Last year more players than ever before received the union minimum, then $242,000 for rookies or $272,000 for veterans. Sports is an entertainment business, not unlike movies. Over the years, NBA efforts to stem the rise of salaries have failed.

A.3 News Article 3

A.3.1 Article

In little more than a week, the world's leaders will converge on this businesslike city in the heart of Southeast Asia for the annual meeting of the Asia Pacific Economic Cooperation forum. They could hardly be meeting in a more provocative place. On Sept. 1, Malaysia discontinued trading in its currency, the ringgit, and imposed sweeping controls on the flow of capital in its stock and currency markets, particularly on investment from overseas. In doing so, the Malaysian prime minister, Mahathir Mohamad, in effect slammed the door on the global economy that President Clinton and the other leaders are coming here to champion. Mahathir's decision drew jeers from international investors and policy- makers, who warned that Malaysia was seeking a quick fix that would retard its desperately needed reforms and leave it the odd man out when Asia finally recovered from the regional malaise. Now, though, Mahathir's allies are marshaling new economic data that they say indicate that capital controls are breathing new life into the country's moribund economy. Malaysia's foreign reserves rose strongly in September, and there is anecdotal evidence that consumers are starting to spend again. "It's nice to be able to say that since we adopted capital controls, the economy has improved," said Zainal Aznam Yusof, the deputy director of the Institute of Strategic and International Studies, a research organization here that helped draft the policies. "But we want to see whether this is strongly sustainable." Critics said it was predictable that capital controls would be a short-term tonic to Malaysia's economy. Because the country is sheltered from the vagaries of capital flows and currency fluctuations, they said, the government had been able to ease interest rates and encourage consumer spending. Still, the mere fact that Malaysia's experiment has yielded some positive results guarantees that the issue will come up during the APEC meeting. With Mahathir leading the campaign, the cause of capital controls will have a fiery advocate who has a penchant for getting under the skin of Westerners. "Mahathir is a very outspoken political leader," said Chia Yew Boon, an independent analyst in Singapore. "There is no way the likes of Clinton or Jiang Zemin are going to be able to muzzle him," he added, referring to President Jiang of China. Policy-makers in the United States have expressed fears that if Malaysia's gambit is seen as successful, other economically weakened countries in the region, like Indonesia, might be tempted to try it. So far, Indonesian officials have said they would stick to the recovery plan devised by the International Monetary Fund, which stresses economic austerity and open markets. But officials in Japan have expressed some sympathy for Mahathir's policies, while Paul Krugman, an economist at the Massachusetts Institute of Technology, has advocated using them as an emergency measure. Yusof said recent events had vindicated Malaysia's contention that it needed to insulate itself from the ravages of the global financial system. He said the recent near-collapse of a prominent American hedge fund underscored how sudden flows of capital can have destructive consequences. The Long-Term Capital Management hedge fund, based in Greenwich, Conn., was nearly wrecked by a series of wrong bets on Treasury securities after the collapse of the economy in Russia prompted a flight of capital out of that country. "The LTCM fiasco really provides a case study of what could go wrong in the global economy," Yusof said. With capital controls as protection, Yusof said Malaysia was picking up the pieces of its shattered economy. In addition to growing foreign reserves, he said Malaysia had improved its trade balance and revived consumer purchases of durable goods. Foreign investors have also not wholly abandoned Malaysia, as experts had predicted they would. While foreign direct investment fell in September to $142 million, from an average monthly rate of $321 million for the period from January through September it did not dry up completely. For every comforting statistic, though, the critics produce an alarming one. They said the increase in Malaysia's foreign reserves was merely due to the new capital restrictions, which stipulated that Malaysian currency held outside the country would be worthless unless repatriated by Sept. 30. The skeptics also noted that bank lending declined in September, despite several reductions in interest rates. So the consumers who are buying new cars and home appliances are merely dipping into their savings, which means the buying spree will end when their savings are depleted. "The argument was that by imposing capital controls, you'd regain control over monetary policy, which would increase the supply of money and lessen the liquidity crunch," said K.S. Jomo, a professor of political economy at the University of Malaya here. "But that's not happening." The biggest flaw in Malaysia's policy, Jomo and others said, is its timing. With the Asian crisis more than a year old, much of the foreign capital that was in the country has already gone. The critics said Mahathir had spooked would-be investors without even locking in the ones who used to be here. "There is a case to be made for the temporary imposition of capital controls, but to avert a crisis, not to respond to one," Jomo said. In fact, other Asian currencies, like the Indonesian rupiah and the Thai baht, have actually rebounded since Malaysia suspended trading in its currency and fixed the exchange rate at 3.8 ringgit to the dollar. Analysts liken the situation to buying an insurance policy for a disaster that has come and gone. More important, the capital controls are slowing down much-needed corporate and banking reforms. The government's rescue of Renong, a major conglomerate with close ties to Mahathir, is going ahead, though some analysts predict the government will eventually scrap the much-criticized plan. The rescue of politically connected companies remains a tense issue here. On Monday, during the trial of Malaysia's former deputy prime minister, Anwar Ibrahim, on charges of corruption and sex-related crimes, Anwar angrily denied Mahathir's claim that he had approved the bailout. Anwar's sensational trial is a reminder that Mahathir's economic policies cannot be disentangled from politics. Malaysia's 72-year-old prime minister clashed with his former protege over how to respond to the Asian crisis, and he dismissed Anwar the day after imposing capital controls. During boom times, Mahathir won support for his policies by wrapping them in anti-foreigner language. In a speech on Monday, he attacked a familiar target, saying foreign currency traders "are the cause of the currency turmoil," adding: "They spread it worldwide. They precipitated the current recession in every country." But Mahathir's treatment of Anwar has stirred anger and sparked growing social unrest in Malaysia. With protesters chanting for reform on the usually orderly streets of this city, experts said Mahathir needed capital controls to work in order to soothe the country's agitated population. "Things will heat up if the economy does not improve," said Chandra Muzaffar, a professor of political science at the University of Malaya. "Then the whole question of Mahathir and his leadership will remain an issue." In that regard, at least, the leaders who converge on Kuala Lumpur in two weeks will be able to identify with their embattled host.

A.3.2 Summary

In little more than a week, the world's leaders will converge on this businesslike city in the heart of Southeast Asia for the annual meeting of the Asia Pacific Economic Cooperation forum. On Sept. 1, Malaysia discontinued trading in its currency, the ringgit, and imposed sweeping controls on the flow of capital in its stock and currency markets, particularly on investment from overseas. Critics said it was predictable that capital controls would be a short-term tonic to Malaysia's economy. They said the increase in Malaysia's foreign reserves was merely due to the new capital restrictions, which stipulated that Malaysian currency held outside the country would be worthless unless repatriated by Sept. 30.

A.4 News Article 4

A.4.1 Article

Under NATO threat to end his punishing offensive against ethnic Albanian separatists in Kosovo, President Slobodan Milosevic of Yugoslavia has ordered most units of his army back to their barracks and may well avoid an attack by the alliance, military observers and diplomats say. Milosevic, who on one hand is excoriated by Washington as the scourge of Kosovo yet on the other hand is treated as key to peace in Bosnia, acted as the European Union, NATO and the United Nations prepared for a review on Monday of possible military intervention. Russia stepped up its warnings against such action and dispatched its foreign and defense ministers on an unusually high-level mission to see the Yugoslav president Sunday in Belgrade. As he has so often, Milosevic appears to have bowed to foreign demands in the nick of time and yet still accomplished what he wanted. This weekend, foreign diplomatic observers in Kosovo reported that a "military stand-down" had taken place in the province, where Milosevic's forces have waged a fierce offensive against Albanian rebels. The observers said that except for segments of three brigades, most units of the Yugoslav army were "home." The daily reports of the observer mission, made up of U.S., European Union and Russian military experts, are one of the key elements in helping Washington and European capitals decide whether Milosevic has met their demands for a cease-fire. By putting the army back in its barracks, sending some police units out of Kosovo and ordering an end to burning and looting of villages, Milosevic may well avoid a NATO attack, diplomats here and in Washington said. But at the same time, they acknowledge that while NATO looked the other way, he enjoyed a three-month license to overwhelm the Kosovo Liberation Army the rebel army fighting for independence for Kosovo and its ethnic Albanian majority and terrorize the rural civilian population that supports it. His military operation created more than 250,000 refugees, whom the Clinton administration is gearing up to take care of this winter through a variety of relief organizations. U.S. officials said they expected Richard Holbrooke, the U.S. envoy who dealt with Milosevic in negotiating an end to the war in Bosnia, to meet with him on Monday to discuss a political plan for Kosovo. The heart of the disagreement in Kosovo is between Serbia, Yugoslavia's principal republic, which insists on keeping Kosovo as a province, and the ethnic Albanians there who have chafed under Milosevic's repression since he stripped the province of virtual autonomy in 1989, and who now seek independence. The West, fearing the precedent that independence for Kosovo would set in other conflicts in the world, has been trying to mediate a middle course. In essence, diplomats said they believed that the plan Holbrooke will present to Milosevic calls for a three-year interim period leading to a status fairly close to the pre-1989 autonomy arrangement. Since the Kosovo conflict flared up in March, critics of Washington's policy toward Milosevic argue that he has been able to choreograph every move to suit his goal: pushing the Albanian population into submission with impunity. "The United States and its allies have waited four months while he cleaned the clock of the Kosovo Liberation Army," said Morton Abramowitz, head of the International Crisis Group, a policy analysis organization, "and taken three weeks to discuss military action, with the result that 500 Albanian villages were destroyed." Administration officials now acknowledge that when NATO failed to live up to its earlier threat in June to strike Serbia, Milosevic took advantage of the indecision and plunged ahead with an artillery and tank offensive against the lightly armed guerrilla forces, whose bedrock of popular support had helped win them effective control of large swaths of Kosovo territory, including key roads. While he was doing that, Milosevic skillfully managed a key requirement for Washington: he made sure that the war did not spill over into neighboring Albania and Macedonia, fragile countries in a traditionally volatile area. All along, the biggest fear in Washington has been that the Kosovo conflict would engulf neighboring countries and encourage Albania and the ethnic Albanian population in Macedonia to join the cause. Such a possibility raised the specter of a new Balkans conflict just three years after peace was secured in Bosnia. Milosevic catered to Washington's concern that the conflict be contained. The Yugoslav army mined Kosovo's borders with Macedonia and Albania, ensuring that few refugees could escape and limiting routes for arms supplies for the rebels. The Yugoslav leader also understood that Washington was unsure about how to deal with the disorganized Albanian political leadership in Kosovo and the unbending Kosovo Liberation Army, whose main chiefs were hardened emigres returned from Switzerland and Germany. For example, Holbrooke persuaded Milosevic to meet in May with Ibrahim Rugova, the top Albanian political leader in Kosovo, an encounter that turned out to be little more than a photo opportunity. For that procedural breakthrough, Holbrooke recommended the lifting of a ban on foreign investment in Serbia that had been put in place the month before. After meeting with Rugova, Milosevic stepped up his military operations in Kosovo, forcing Washington to reverse itself again and carry out the investment ban. In late June, Holbrooke met with two self-styled Kosovo guerrilla commanders in the province's western town of Junik but then broke off all contact. Clinton administration officials said at the time that they were concerned that NATO intervention would bolster the separatist forces. To try to put the best face on the situation, Washington worked with Moscow to get Milosevic to accept the presence of international monitors who would patrol Kosovo and report on military action. The monitors were slow in getting organized. By August, when the Yugoslav army, backed by the Serbian special police, were in full swing against the rebels and burning and looting villages in the process, the monitors found it difficult to gain access to the fighting. They drove up to roadblocks, knew something was going on from the sounds and the smoke, but could not be precise. In recent days as the tanks and artillery have withdrawn, access has improved, the monitors say. But there are some areas in central Kosovo around Likovac and Gornje Obrinje that the monitors have ruled off limits because of land mines on the roads. The mines are believed to have been planted by the guerrillas. Gornje Obrinje was the site of a massacre of 18 ethnic Albanian women, children and elderly people on Sept. 26. A British reporter who walked across fields into the village on Sunday said about 10 mortar shells, apparently from the Serbian police or the Yugoslav army, were fired at the village early Sunday afternoon. The Yugoslav army and police forces have been responsible for the vast majority of atrocities in the Kosovo conflict, said a report by New York-based group Human Rights Watch, released here on Sunday. The report said the rebels had also violated the laws of war by taking civilian hostages and carrying out summary executions. But the violations by the guerrillas were on a “lesser scale” than the government abuses, the author of the report, Fred Abrahams, concluded. The report focused on what it called a watershed in the conflict the attack by police forces on three ethnic Albanian villages in late February and early March in the Drenica region of central Kosovo. At least 83 people, including 24 women and children, were killed in the attack, which involved helicopters, artillery and armored personnel carriers. In the Yugoslav capital, Belgrade, which is a four-hour drive north through rolling countryside from Kosovo’s capital of Pristina, Milosevic remains politically secure. That is in part, his domestic critics say, because diplomats like Holbrooke and the head of the U.N.refugee agency, Sadako Ogata, insist on going to see him, thus enhancing his stature.

A.4.2 Summary

Under NATO threat to end his punishing offensive against ethnic Albanian separatists in Kosovo, President Slobodan Milosevic of Yugoslavia has ordered most units of his army back to their barracks and may well avoid an attack by the alliance, military observers and diplomats say. By August, when the Yugoslav army, backed by the Serbian special police, were in full swing against the rebels and burning and looting villages in the process, the monitors found it difficult to gain access to the fighting. In recent days as the tanks and artillery have withdrawn, access has improved, the monitors say. Gornje Obrinje was the site of a massacre of 18 ethnic Albanian women, children and elderly people on Sept. 26.

A.5 News Article 5

A.5.1 Article

The New York Times said in an editorial on Monday, Nov. 23: The Russian reform movement has produced few leaders with an uncompromising dedication to democracy. Galina Starovoitova was one, and her murder in St. Petersburg on Friday was a terrible loss for Russia. In a bleak season of economic collapse and political timidity, the killing can only heighten fears that Russia is slipping into an ugly era of intolerance and political violence. Initial evidence suggests that the killing was a political assassination. Ms. Starovoitova was gunned down in the lobby of her apartment building, shot three times in the head, typical of Russian contract killings. She was a member of the Russian parliament and a recently declared candidate for governor of the region around St. Petersburg. In recent weeks she had spoken out forcefully against political extremism, denounced the anti-Semitic statements of a Communist parliamentarian and was campaigning aggressively against financial corruption in the St. Petersburg municipal government. Ms. Starovoitova's activities were fully in character with a career built around principles of liberty, tolerance and the rule of law. She championed democracy and human rights long before they became politically acceptable in Moscow, and courageously stood by Boris Yeltsin and other reformers as Russia struggled to find a new political course when the Soviet Union disintegrated. An ethnographer by training, Ms. Starovoitova proved to be a skillful and effective politician. She first gained national attention a decade ago when she set aside her academic work about the ethnic history of Leningrad and ran successfully for a seat in the Soviet parliament from Armenia, a startling victory for a Russian. She later represented St. Petersburg in the Russian legislature. Ms. Starovoitova was a woman of irrepressible energy and infectious enthusiasm. But her good humor and quick smile belied a steely commitment to combat the corruption and ethnic divisions that she correctly considered to be the enemies of Russian democracy. The least Yeltsin can do is to hunt down her killers and bring them to trial. That would be the exception in a nation where political violence is rarely prosecuted. Her countrymen can honor her memory by following her example.

A.5.2 Summary

The New York Times said in an editorial on Monday, Nov. 23 : The Russian reform movement has produced few leaders with an uncompromising dedication to democracy. She championed democracy and human rights long before they became politically acceptable in Moscow, and courageously stood by Boris Yeltsin and other reformers as Russia struggled to find a new political course when the Soviet Union disintegrated. She first gained national attention a decade ago when she set aside her academic work about the ethnic history of Leningrad and ran successfully for a seat in the Soviet parliament from Armenia, a startling victory for a Russian.

Bibliography

[1] Eurowordnet. http://www.illc.uva.nl/EuroWordNet/.

[2] *Roget's Theasurus of English Words and Phrases*. Longman Group UK Limited, London, 1987.

[3] *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*. ACM, 1998.

[4] *Proceedings Document Understanding Conference 2004*. Boston, USA, July 2004.

[5] Laura Alonso Alemany and Maria Fuentes Fort. Integrating cohesion and coherence for automatic summarization. Budapest, Hungary, April 12–17 2003.

[6] Regina Barzilay and Michael Elhadad. Using Lexical Chains for Text Summarization. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 111–121. The MIT Press, 1999.

[7] Regina Barzilay and Lillian Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003: Main Proceedings*, pages 16–23, 2003.

[8] Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL*, pages 113–120, 2004.

[9] Regina Barzilay and Kathleen R. McKeown. Sentence fusion for multidocument news summarization. In *Computational Linguistics*, volume 31, pages 297 – 328. MIT Press., 2005.

[10] R. Brandow, K. Mitze, and Lisa F. Rau. Automatic condensation of electronic publications by sentence selection. *Inf. Process. Manage.*, 31(5):675–685, 1995.

[11] Meru Brunn, Yllias Chali, and Christopher J. Pinchak. Text summarization using lexical chains. New Orleans, LA, 2001.

[12] A. Budanitsky. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures, 2001.

[13] Yllias Chali and Maheedhar Kolla. University of lethridge summarizer at duc04. In *DUC04*, Boston, USA, July 2004.

[14] B. Daille, E. Gaussier, and J. Lange. Towards automatic extraction of monolingual and bilingual terminology, 1994.

[15] R.E. De War and D.L. Neal. Weka machine learning project: Cow culling. Technical report, The University of Waikato, Computer Science Department, Hamilton, New Zealand, 1994.

[16] William P. Doran, Nicola Stokes, Joe Carthy, and John Dunnion. Assessing the impact of lexical chain scoring methods and sentence extraction schemes on summarization. In *CICLing*, pages 627–635, 2004.

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

[18] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. (JAIR)*, 22:457–479, 2004.

[19] David K. Evans, Judith L. Klavans, and Nina Wacholder. Document processing with linkit.

[20] C. Fellbaum. *Wordnet: An Electronic Lexical Database.* Cambridge, US: The MIT Press, 1998.

[21] Michel Galley and Kathleen McKeown. Improving word sense disambiguation in lexical chaining. In *IJCAI*, pages 1486–1488, 2003.

[22] M. Halliday and R. Hasan. *Cohesion in English.* Longman, London, 1976.

[23] S. Harabagiu, G. Miller, and D. Moldovan. Wordnet 2 - a morphologically and semantically enhanced resource, 1999.

[24] Marti A. Hearst. Tilebars: Visualization of term distribution information in full text information access. In *CHI*, pages 59–66, 1995.

[25] E.H. Hovy. *Automated Text Summarization*, chapter The Oxford Handbook of Computational Linguistics, pages 583–598. 2005.

[26] Anette Hulth. Reducing false positives by expert combination in automatic keyword indexing. In *RANLP*, pages 367–376, 2003.

[27] Lovins JB. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 1968.

[28] H. Jing. Sentence reduction for automatic text summarization, 2000.

[29] John S. Justeson and Slava M. Katz. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1995.

[30] John Chen David Elson David Evans Judith Klavans Ani Nenkova Barry Schiffman Kathleen McKeown, Regina Barzilay and Sergey Sigelman. Columbia's newsblaster: New features and future directions (demo). In *In Proceedings of NAACL-HLT'03*, 2003.

[31] Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artif. Intell.*, 139(1):91–107, 2002.

[32] Julian Kupiec, Jan O. Pedersen, and Francine Chen. A trainable document summarizer. In *SIGIR'95*, pages 68–73. ACM Press, 1995.

[33] Chin-Yew Lin and Eduard H. Hovy. Identifying topics by position. In *ANLP*, pages 283–290, 1997.

[34] Chin-Yew Lin and Eduard H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *HLT-NAACL*, 2003.

[35] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2(2):159–165, 1958.

[36] Inderjeet Mani, Barbara Gates, and Eric Bloedorn. Improving summaries by revising them. In *ACL*, 1999.

[37] Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Therese Firmin, and Beth Sundheim. The tipster summac text summarization evaluation. In *EACL*, pages 77–85, 1999.

[38] Daniel Marcu. The rhetorical parsing of natural language texts. In *ACL*, 1997.

[39] Daniel Marcu. *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. PhD thesis, University of Toronto, 1997.

[40] Daniel Marcu. Discourse trees are good indicators of importance in text. In *Advances in Automatic Text Summarization*, pages 123–136, 1999.

[41] Daniel Marcu. Discourse-based summarization in duc-2001. In *DUC01*, New Orleans, LA, 2001.

[42] A.G. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[43] Jane Morris and Graeme Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.

[44] Kemal Oflazer Orhan Bilgin, zlem etinolu. Building a wordnet for turkish. *Romanian Journal of Information Science and Technology*, 7(1-2), 2004.

[45] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[46] Jian-Yun Nie Quan Zhou Le Sun. Is sum: A multi-document summarizer based on document index graphic and lexical chains. In *DUC05*, Vancouver, CA, July 2005.

[47] Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. MEAD - a platform for multidocument multilingual text summarization. In *LREC 2004*, Lisbon, Portugal, May 2004.

[48] Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. In Udo Hahn, Chin-Yew Lin, Inderjeet Mani, and Dragomir R. Radev, editors, *ANLP/NAACL00-WS*, Seattle, WA, April 2000.

[49] Lisa F. Rau and Paul Jacobs. Creating segmented databases from free text for text retrieval. In *SIGIR91*, pages 337–346, New York, NY, 1991.

[50] Mirella Lapata Regina Barzilay. Modeling local coherence: An entity-based approach. In *Proceedings of ACL*, 2005.

[51] Tat-Seng Chua Shiren Ye, Long Qiu and Min-Yen Kan. Nus at duc 2005: Understanding documents via concept links. In *DUC05*, Vancouver, CA, July 2005.

[52] Gregory H. Silber and Kathleen McCoy. Efficient text summarization using lexical chains. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI'2000)*, January 9–12 2000.

[53] D. St-Onge. Detecting and correcting malapropisms with lexical chains. Master's thesis, Department of Computer Science, University of Toronto, 1995.

[54] Simone Teufel and Marc Moens. Sentence extraction as a classification task. In Inderjeet Mani and Mark T. Maybury, editors, *ACL/EACL97-WS*, Madrid, Spain, 1997.

[55] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*, 2003.

[56] Peter D. Turney. Learning algorithms for keyphrase extraction. *Inf. Retr.*, 2(4):303–336, 2000.

[57] E. Newman J. Dunnion J. Carthy F. Toolan W. Doran, N. Stokes. News story gisting at university college dublin. In *DUC04*, Boston, USA, July 2004.

[58] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *ACM*, pages 254–255, 1999.