Multi-Document Update and Opinion Summarization

Kumar Puspesh

In partial fulfillment of
the requirements for the degree of

Master of Technology

Indian Institute of Technology Kharagpur

2008

Under the guidance of,
Prof. Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

CERTIFICATE

This is to certify that the thesis titled **"Multi-Document Update and Opinion Summarization"** submitted by Mr. Kumar Puspesh to the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur in partial fulfillment of the requirement for the degree of Master of Technology during the academic year 2007-2008 is a record of authentic work carried by him under my supervision and guidance.

Date: May 7th, 2008

Prof. Sudeshna Sarkar
Professor
Department of CSE
IIT Kharagpur

Abstract

Automatic summarization involves reducing a text document or a larger corpus of multiple documents into a short set of words or paragraph that conveys the main meaning of the text. Two particular types of summarization often addressed in the literature are keyphrase extraction, where the goal is to select individual words or phrases to "tag" a document, and document summarization, where the goal is to select whole sentences to create a short paragraph summary. A recent area which is evolving is of opinion summarization where a consolidated summary of various opinionated sentences or phrases on certain features of the topic are identified and presented in a suitable manner. In this document, we discuss about a summarization system built using MEAD framework for multi-document summarization and update summarization and another system build specially for the purpose of opinion summarization using novel techniques for automatic feature extraction from product reviews. We also look on novel methods of sentiment analysis for product review opinions which can be extended to other types of texts also without much changes. We try to evaluate different approaches using our system along with some of the summarization systems submitted in earlier Document Understanding Conferences.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

With great pleasure and deep sense of gratitude, I express my indebtedness to Prof. Sudeshna Sarkar for her invaluable guidance and constant encouragement at each and every step of my project work. She exposed us to the intricacies of relevant topics through paper counseling and discussions and always showed great interest in providing timely support and suitable suggestions.

I would also like to express my gratitude to all my friends in the Department of Computer Science and my hostel for their constant support and encouragement. Words are not enough to express my gratitude towards my parents to whom I owe every success and achievements of my life. Their constant support and encouragement under all odds has brought me where I stand today.

Date: May 7th, 2008

Kumar Puspesh
03CS3025
Department of CSE
IIT Kharagpur

to my parents and friends

Chapter 1

INTRODUCTION

Automated text summarization has drawn a lot of interest in the natural language processing and information retrieval communities in the recent years. The task of a text summarizer is to produce a synopsis of any document (or set of documents) submitted to it. The level of sophistication of a synopsis or a summary can vary from a simple list of isolated keywords that indicate the major content of the document(s), through a list of independent single sentences that together express the major content, to a coherent, fully planned and generated text that compresses the document(s). The more sophisticated a synopsis, the more effort it generally takes to produce.

Several existing systems, including some Web browsers, claim to perform summarization. However, a cursory analysis of their output shows that their summaries are simply portions of the text, produced verbatim. While there is nothing wrong with such extracts, per se, the word 'summary' usually connotes something more, involving the fusion of various concepts of the text into a smaller number of concepts, to form an abstract. We define extracts as consisting wholly of portions extracted verbatim from the original (they may be single words or whole passages) and abstracts as consisting of novel phrasings describing the content of the original (which might be paraphrases or fully synthesized text). Generally, producing a summary requires stages of topic fusion and text generation not needed for extracts.

In addition to extracts and abstracts, summaries may differ in several other ways. Some of the major types of summary that have been identified include indicative (keywords indicating topics) vs. informative (content laden); generic (author's perspective) vs. query-oriented (user-specific); normal vs. update; background vs. just-the-news; single document vs. multi-document; neutral vs. evaluative. A full under-

standing of the major dimensions of variation, and the types of reasoning required to produce each of them, is still a matter of investigation. This makes the study of automated text summarization an exciting area in which to work. Now the area of Multi-document summarization can be seen further subdivided into various domains like - opinion summarization, update summarization, query-based summarization etc. Various search engines like Google, Yahoo etc. provide a short snippet alongwith each search result for any query given by the user. The automatic text summarization techniques are of great use in these real-world scenarios.

The Web contains a wealth of opinions about products, politicians, and more, which are expressed in newsgroup posts, review sites, and elsewhere. As a result, the problem of opinion mining has seen increasing attention in recent years. Our work is mainly focussed on product reviews but the methodlogy in general works for a borader rabge of opinions. Documents discussing public affairs, common themes, interesting products, and so on, are reported and distributed on the Web in abundance. Positive and negative opinions embedded in documents are useful references and feedbacks for governments to improve their services, for companies to market their products, and for customers to purchase their objects. Web opinion mining aims to extract, summarize, and track various aspects of subjective information on the Web. Mining subjective information enables traditional information retrieval (IR) systems to retrieve more data from human viewpoints and provide information with finer granularity. Opinion extraction identifies opinion holders, extracts the relevant opinion sentences, and decides their polarities. Opinion summarization recognizes the major events embedded in documents and summarizes the supportive and the nonsupportive evidence. Opinion tracking captures subjective information from various genres and monitors the developments of opinions from spatial and temporal dimensions. For any product there are numerous reviews available online and a summarized view of all those can be more inforamtive to the user in much lesser time. News, blogs and product reviews are some importatn sources of opinions, in general. Because queries may or may not be posed beforehand, detecting opinions is somewhat similar to the task of topic detection at sentence level. We try to look into automatic feature extraction mechanisms from product reviews and further opinion summarization techniques which retrieves

relevant information from the document set, determines the polar orientation of each relevant sentence and finally summarizes the positive-negative sentences accordingly. For example, if there are a number of reviews available on *Canon Powershot TX1 digital camera* and users have pointed out some positive/negative aspects of the camera, then a new user needs to go through all the reviews to know other people's opinions on the camera features. But, while doing so, he might read similar opinions many times i.e.; the problem of redundant information surfaces up. Another problem is the number of documents needed for the user to know the opinions on some specific of the camera. If the user gets to see a summary of the digital camear reviews on the different features, he can easily decide whether the camera satisfies his needs or not.

1.1 The Problem

We have tried to develop a system which tries to solve follwoing problems -

- Given a set of doucments on any specific topic, the general multi-document summarization problem is to identify and generate a simple summary/abstract from the given documents which covers the information present in the document set to as much extent as possible.

- Assuming that the user has already been provided with the summary of the documents in the dataset on a particular topic upto the current time, the problem of *Update Summarization* is to generate summary on that topic so that some new information, that has not already been served, is presented to the user.

- Given a set of unstructred customer reviews on any particular product, the idea is to generate a summary of the product over its key features outlining positive or negative views of the users and the reasons provided for those.

We have tried to develop a system over some existing frameworks to find a reasonable solution for the above problems.

1.2 Organization of the thesis

The Thesis is organized into 8 different chapters. The first chapter introduces us to the problem and its relevance in the real world scenarios. The second chapter gives us the background and the related work in the similar areas. The third chapter describes the motivation and the aim that we want to achieve. The fourth chapter describes the system architecture in complete detail for the problem of Multi-document and Update summarization whereas, the fifth chapter describes the same for Opinion Summarization. The sixth chapter describes the various evaluation measures used and states some of the results. The seventh chapter concludes the work and outlines the scope for future work. The eigth and final chapter cites the references.

Chapter 2

BACKGROUND AND LITERATURE SURVEY

Almost all previous multi-document summarization systems, including SumBasic, have used greedy or heuristic searches to choose which sentences to use, even when they had an explicit scoring function. SumBasic and Microsoft system [Vanderwende, Yih, Suzuki and Goodman] focus on scoring individual words. In contrast, most existing systems are sentence-based. These sentence-based systems use a variety of features, including: sentence position in the document, sentence length, sentence similarity to previously extracted sentences (usually using the maximal marginal relevance (MMR) framework [Carbonell and Goldstein, 1998]), an explicit redundancy score [Daume III and Marcu, 2005], and sentence similarity to the document centroid. In the cases where individual words are considered during sentence selection, important words are identified through graph-based analysis where the nodes in the graph represent words [Mani and Bloedorn, 1997; Erkan and Radev, 2004]. The techniques tried during the 1950's and 60's were characterized by their simplicity of processing, since at that time neither large corpora of text, nor sophisticated NLP modules, nor powerful computers with large memory existed.

Although each of these approaches has some utility, they depend very much on the particular format and style of writing. The strategy of taking the first paragraph, for example, works only in the newspaper and news magazine genres, and not always then either. No automatic techniques were developed for determining optimal positions, relevant cues, etc. True summarizing requires the understanding and interpretation of the text into a new synthesis, at different levels of abstraction. Semantics-based Artificial Intelligence (AI) techniques developed in the 1970's and early 80's promised to provide the necessary reasoning capabilities. Recent approaches use frames or templates that house the most pertinent aspects of stereotypical situations and objects

(Mauldin 91; Rau 91). As outlined in (McKeown and Radev 95), such templates form an obvious basis from which to generate summaries. A fixed output template system is by its definition limited to the contents of the template, and it can never exceed this boundary. One is forced to turn to less semantic, more robust techniques. Since the 1950's, IR researchers have spent a great deal of effort in developing methods of locating texts based on their characteristics, categorizing texts into predefined classes, and searching for incisive characterizations of the contents of texts (Salton 88; Rijsbergen 79; Paice 90).

Scaling down one's perspective from a large text collection to a single text (i.e., a collection of words and phrases), topic identification for extracts can be seen as a localized IR task. The pure IR approach does have limitations, however. IR researchers have tended to eschew symbolic representations; anything deeper than the word level has often been viewed with suspicion. This attitude is a strength, because it frees IR researchers from the seductive call of some magical powerful internal representation that will solve all the problems easily; it is a weakness, because it prevents researchers from employing reasoning at the non-word level. Unfortunately, abstract-type summaries require analysis and interpretation at levels deeper than the word level. Although word-level techniques have been well developed and applied in many practical cases, they have been criticized in several respects (Mauldin 91; Riloff 94; Hull 94) because of these - Synonymy, polysemy, phrases, and term dependency problems all relate to semantics. Using a thesaurus, one can identify synonyms, using a sense disambiguation algorithm (e.g., Yarowsky 92), one can select the correct sense of a polysemous word, using a syntactic parser, one can extract phrase segments and use them as terms (Lewis 92). Latent semantic indexing (Deerwester et al. 90; Hull 94) has been used to remedy the term dependency problem. All these efforts are attempts to bridge the gap between word form and word meaning. Following this trend, there is increasing interest in integrating shallow semantic processing and word based statistical techniques to improve the performance of automatic text categorization systems (Liddy 94; Riloff 94).

Opinion extraction identifying components which express opinions is fundamental

for summarization, tracking, and so on (Ku, Li, Wu and Chen, 2005). At document level, Wiebe, Wilson and Bell (2001) recognized opinionated documents. Pang, Lee, and Vaithyanathan (2002) classified documents by overall sentiments instead of topics. Daves (2003) and Hus (2004) researches focus on extracting opinions of reviews. Riloff and Wiebe (2003) distinguish subjective sentences from objective ones. Kimand Hovy (2004) propose a sentiment classifier for English words and sentences, which utilizes thesauri. However, template-based approach needs a professionally annotated corpus for learning, and words in thesauri are not always consistent in sentiment. Hu and Liu (2004) proposed an opinion summarization of products, categorized by the opinion polarity. Liu, Hu and Cheng (2005) then illustrated an opinion summarization of bar graph style, categorized by product features. Nevertheless, they are both domain-specific. Wiebe et al. (2002) proposed a method for opinion summarization by analyzing the relationships among basic opinionated units within a document. Extracting opinions on products (Hu and Liu, 2004) is different from that on news or writings. For these kinds of articles, major topic detection is critical to expel non-relevant sentences (Ku, Li, Wu and Chen, 2005) and single document summarization is not enough. Pang, Lee and Vaithyanathan (2002) showed that machine learning approaches on sentiment classification do not perform as well as that on traditional topic-based categorization at document level. Information extraction technologies (Cardie et al., 2004) have also been explored. A statistical model is used for sentiment words too, but the experiment material is not described in detail (Takamura et al., 2005). The results for various metrics and heuristics also depend on the testing situations.

Chapter 3

MOTIVATION

There are many systems developed for the solution to multi-document summarization and opinion summarization problems like SumBasic, OPINE etc. But there are not many systems designed for the update summarization task. This is a relatively newer field of summarization was introduced only in Document Understanding Comference 2007.

In real world, update summarization has a lot of value as the content on web is highly dynamic. For example, the customer reviews on various sites increase daily as different users write their comments about the product. So, for a new user who wishes to see the public opinion on the any particular product a good summarization system can help him in getting the required and relevant information without going through all the reviews present on the site. Update summarization can help in keeping this system *online* by continuosly serving much newer information about the product gathered from the reviews which are new. This example also highlights the importance and use of opinion summarization in real world scenario. We can see more usages of opinion summarization or sentiment analysis or subjectivity analysis in follwing scenarios:

- Classifying reviews as positive/negative
- Analyzing product reputations
- Tracking sentiments toward topics and events
- Recognizing hostile messages
- Genre classification
- Opinion-oriented question answering
- Improving information extraction
- Improving word-sense disambiguation

We wished to participate in DUC 2008 specially on the update summarization track. But, the Document Understanding Conferences were merged with TREC and another conference *Text Analysis Conference* (TAC) has been initiated from thsi year. We want to participate in the opinion summarization task of TAC 2008 and our work is mainly directed towards acheiving this target along with exploring other avenues of applications and combination of update and opinion summarization at a later stage.

Our aim is to develop a state-of-art opinion summarization system which can identify the product features and summarize the inforamtion present in the customer reviews irrespective of the product class.

Chapter 4

MULTI-DOCUMENT AND UPDATE SUMMARIZATION

Multi-document summarization creates information reports that are both concise and comprehensive. With different opinions being put together and outlined, every topic is described from multiple perspectives within a single document. While the goal of a summary is to simplify information search and cut the time by pointing to the most relevant source documents, comprehensive multi-document summary should itself contain the required information, hence limiting the need for accessing original files to cases when refinement is required. Automatic summaries present information extracted from multiple sources algorithmically, without any editorial touch or subjective human intervention, thus making it completely unbiased. We have used MEAD framework to develop a system which genrates summaries of desired lengths from a pool of documents on a single topic.

4.1 Multi-Document Summarization : Definition

Given a set of documents $D = (d_1, d_2, ..., d_n)$ on a topic T, the task of multi-document summarization is to identify a set of model units $(s_1, s_2, ..., s_m)$, where $m \leq n$, such that the selected model units s_i carry as much diverse information as possible from the set D and only the information present in the set D. The model units can be sentences, phrases or some generated semantically correct language units carrying some useful information.

4.2 Update Summarization : Definition

In general, given a set of document sets $(D_1, D_2, D_3...)$ update summarization task is to provide summary S_t at time step t from respective document set D_t assuming that the user has already read the summaries provided at time steps earlier than t

i.e., $\forall\, t' \leq t$, summaries S'_t have been already provided to the user, hence information presented in S_t should cover newer information more.

4.3 MEAD framework

MEAD is a public domain portable multi-document summarization and evaluation toolkit. It is a publicly available toolkit for multi-lingual summarization and evaluation. The toolkit implements multiple summarization algorithms (at arbitrary compression rates) such as position-based, Centroid[RJB00], TF*IDF, and query-based methods. MEAD can perform many different summarization tasks. It can summarize individual documents or clusters of related documents (multi-document summarization). Originally, MEAD includes two baseline summarizers: lead-based and random. Lead-based summaries are produced by selecting the first sentence of each document, then the second sentence of each, etc. until the desired summary size is met. A random summary consists of enough randomly selected sentences (from the cluster) to produce a summary of the desired size.

MEAD has been primarily used for summarizing documents in English, but recently, Chinese capabilities have also been added. Query-based summarization is often used in natural language circles, and is (not coincidentally) included in MEAD as well. The MEAD evaluation toolkit (MEAD Eval), previously available as a separate piece of software, has been merged into MEAD as of version 3.07. This toolkit allows evaluation of human-human, human-computer, and MEAD User Documentation computer-computer agreement. MEAD Eval currently supports two general classes of evaluation metrics: co-selection and content-based metrics. Co-selection metrics include precision, recall, Kappa, and Relative Utility, a more flexible cousin of Kappa. MEAD's content-based metrics are cosine (which uses TF*IDF), simple cosine (which doesn't), and unigram- and bigram-overlap. Relevance correlation has previously been used in conjunction with MEAD.

4.4 Data Collection

The data we have used is taken from Document Understanding Conference (DUC) 2006 repository which includes documents and the submitted peer summaries along-

with the humna-generated summaries which were used to evaluate the submitted summaries. The dataset consists of 50 different topics and for each topic there are 25 documents provided (hence, a total of 1250 documents). Each topic has 4 human-generated summaries for evaluation purposes.

4.5 System Architecture

The module is developed over MEAD summarization framework and it toally follows the flow as MEAD. The figure 4.1 shows the overview of the multi-document (general and update both) summarization modules.

Summarizer
Input: a set of documents on a single topic (D), desired length of summary (L)
Output: summary Summ

$S \leftarrow GetSentences(D)$
$\forall s \, \epsilon \, S \;\; F(s) \leftarrow FeatureScripts(s)$
$RankedSents \leftarrow Classifier(F)$
$RerankedSents \leftarrow Reranker(RankedSents)$
$Summ \leftarrow Postprocessor(RerankedSents)$

Figure 4.1: Multi-Document Summarization *Overview*

4.6 Preprocessor

We have used DUC 2006 data and reference summaries for our study. The documents are in a specific xml format. But, the input format of MEAD is a different xml structure. The preprocessor changes the format of the documents and modifies the document a little bit to remove some discrepancies. The documents given by DUC 2006 are not well formatted as they have mistakenly grouped many sentences under the same tag. Which makes the system treat those as a single sentence only.

4.7 Feature Scripts

Feature Scripts are the modules which compute values of various features of the set of sentences. As our system is based on sentence-based algorithms, these modules essentially compute values of various features for each sentence present in the document pool. The feature values for each sentence present in a document are grouped together to form a feature vector for the document. MEAD provides a platform to add different feature vector computing scripts. It uses a three pass feature vector computation model - Cluster level, Document level and Sentence level. The first two levels are optional but computation of feature values at the last level is a must because this is the final step which gives scores to different sentences. The MEAD framework is such that many features can be computed for the same set of sentences. It comes with some simple features like Length, Position etc. Other researchers have also contributed and added Centroid feature in MEAD. We have added Lexrank into the system and studied various combination of features.

4.7.1 Position

This feature is relevant in identifying important sentences as generally in any document, the sentences at the start of the paragraph or article are more important. Position feature assigns each sentence a value as,

$$P(s) = 1/n, \tag{4.1}$$

where n is the number of the sentences in the document.

4.7.2 Length

Sentences having length less than the specified threshold are assumed to be non-relevant for the summarization purpose. The data on which we are working is a crawl of different news articles on same topic. So, it does contain some small phrases which are just a topic name or a bullet etc.

4.7.3 Centroid

A centroid is a set of words that are statistically important to a cluster of documents. As such, centroids could be used both to classify relevant documents and to identify salient sentences in a cluster. The centroid of a cluster is a pseudo-document which consists of words that have tf*idf scores above a redefined threshold.

Centroid is a feature which is dependent on the words present in the sentence. The more important words it contains, more central it is in respect of the document cluster. For computation of centroid feature, we first find out the term frequencies of various words present in the document. Then, for each word TF*IDF is computed where IDF is defined as,

$$IDF(i) = log(\frac{N}{n_i}) \tag{4.2}$$

Where, N is total number of documents and n_i is the number of documents in which the word i is present. Now, for each sentence C_i the combined centroid score is calculated as ,

$$C_i = \sum C_{w,i} \tag{4.3}$$

Where, $C_{w,i}$ is the TF*IDF score of the word w in the sentence i.

4.7.4 LexRank

It is inspired the PageRank algorithm used by Google for ranking of webpages across the world wide web. PageRank is a graph based algorithm which assigns prestige to each node (which are the webpages in general). A variant of PageRank can be used in multi-document extractive generic text summarization. The basic task for any extractive summarization is finding the most central sentences from the cluster of documents - here it is done by finding the most prestigious sentences. (Also, Centrality of a sentence is calculated in terms of centralities of words that it contains).

This approach is based on the concept of prestige in social networks, which has also inspired many ideas in computer networks and information retrieval. A social network is a mapping of relationships between interacting entities (e.g. people, organizations, computers). Social networks are represented as graphs, where the nodes represent the

entities and the links represent the relations between the nodes. A cluster of documents can be viewed as a network of sentences that are related to each other. Some sentences are more similar to each other while some others may share only a little information with the rest of the sentences. We hypothesize that the sentences that are similar to many of the other sentences in a cluster are more central (or salient) to the topic. There are two points to clarify in this definition of centrality. First is how to define similarity between two sentences. Second is how to compute the overall centrality of a sentence given its similarity to other sentences.

To define similarity, we use the bag-of-words model to represent each sentence as an N-dimensional vector, where N is the number of all possible words in the target language. For each word that occurs in a sentence, the value of the corresponding dimension in the vector representation of the sentence is the number of occurrences of the word in the sentence times the idf of the word.

$$idf - modified - cosine(x, y) = \frac{\sum\limits_{w \,\epsilon\, x,y} tf_{w,x}\, tf_{w,y} (idf_w)^2}{\sqrt{\sum\limits_{x_i \,\epsilon\, x} (tf_{x_i,x} * idf_{x_i})^2}\ \sqrt{\sum\limits_{y_i \,\epsilon\, y} (tf_{y_i,y} * idf_{y_i})^2}} \quad (4.4)$$

A cluster of documents may be represented by a cosine similarity matrix where each entry in the matrix is the similarity between the corresponding sentence pair. For computing prestige scores of different sentences -

1. *Degree Centrality* Degree centrality may have a negative effect in the quality of the summaries in some cases where several unwanted sentences vote for each other and raise their centrality. As an extreme example, consider a noisy cluster where all the documents are related to each other, but only one of them is about a somewhat different topic. Obviously, we would not want any of the sentences in the unrelated document to be included in a generic summary of the cluster. However, suppose that the unrelated document contains some sentences that are very prestigious considering only the votes in that document. These sentences will get artificially high centrality scores by the local votes from a specific set of sentences.

2. *Eigen Centrality* This situation can be avoided by considering where the votes

come from and taking the centrality of the voting nodes into account in weighting each vote. A straightforward way of formulating this idea is to consider every node having a centrality value and distributing this centrality to its neighbors. This formulation can be expressed by the equation

$$p(u) = \sum_{v \epsilon adj(u)} p(v)/deg(v) \tag{4.5}$$

where $p(u)$ is the centrality of node u, $adj(u)$ is the set of nodes that are adjacent to u, and $deg(v)$ is the degree of the node v. The above equation can be written equivalently as,

$$p = B^T p \ \ or, p^T = Bp^T \tag{4.6}$$

where the matrix B is obtained from the adjacency matrix of the similarity graph by dividing each element by the corresponding row sum. This equation states that p^T is the left eigenvector of the matrix B with the corresponding eigen value of 1. The centrality vector p corresponds to the stationary distribution of B. However, we need to make sure that the similarity matrix is always irreducible and aperiodic.

4.8 Classifier

This step merges the different feature vectors which were already computed in the last step. Various kinds of classifiers can be incorporated in MEAD.

1. Default classifier:
   It is user programmable in the sense that it allows us to assign different weights to different features. We have used different combination of features to study the quality of summary produced. Each sentence receives a score that is a linear combination of the features listed (provided they are in the input feature file) EXCEPT for the Length feature. The weight of each feature in the linear combination is specified while the classifier is initiated. Length, if it is given, is a cutoff feature. Any sentence with a length shorter than Length is automatically given a score of 0, regardless of its other features. Length is the only feature

that has these semantics.

$$Score(S_i) = feature_1 * weight_1 + feature_2 * weight_2 + ... \quad (4.7)$$

2. Leadbased classifier:
   The leadbased-classifier.pl script is part of the leadbased baseline summarizer. This classifier assigns a score of $\frac{1}{n}$ to each sentence, where is the sentences SNO in the corresponding docsent file. This means that the first sentence in each document will have the same scores, the second sentence in each document will have the same scores, etc. Again, if a Length feature argument is provided, the sentences with lengths less than the specified value are thrown out.

We have experimented mainly with the default-classifier with a combination of different feature scripts like length, position, centroid and lexrank.

4.9 Reranker

The reranker is used to modify sentence scores based on relationships between pairs of sentences. For example, it can be used to give lower scores to repeated instances of a sentence or higher scores to a sentence that has an anaphoric relationship with another sentence. The input to a reranker is a reranker-info file A reranker-info file has three components: compression information, cluster information, and the sentence scores as computed by the reranker. The compression information has the same form as it does in the mead-config file: it specifies whether the BASIS should be words or sentences, and how large the summary should be, either in comparison to the entire cluster (PERCENT) or as an absolute size (ABSOLUTE). The cluster information looks almost exactly like a cluster file, but without the XML headers. Rerankers use this in order to open the cluster to examine and compare the text of each sentence. The sentence scores take the form of a sentjudge file.

The theory behind the idea of reranker step is *Cross-sentence Informational Subsumption* (CSIS) :
- It reflects that some sentences repeat the information present in other sentences and may, therefore, may be omitted during summarization.

- If the information content of sentence a is contained within sentence b, then a becomes informationally redundant and the content of b is said to subsume that of a. e.g.

1. John Doe was found guilty of murder.
2. The court found John Doe guilty of the murder of Jane Doe last august and sentenced him to death.

In the above sentences, more or less same information is present and if both sentences are used in summary, then this reduces the amount of information captured by the summarization system. So, we try to remove redundancy to some extent at this level. Here are the different kinds of rerankers that we have used in the system:

1. Default Reranker

   The default reranker orders the sentences by score from highest to lowest, and iteratively decides whether to add each sentence to the summary or not. At each step, if the quota of words or sentences has not been filled, and the sentence is not too similar to any higher-scoring sentence already in the summary, the sentence in question is added to the summary. After the summary has been filled, the default reranker increases the scores of the chosen sentences and decreases the scores of the disqualified (by similarity) or unchosen sentences.

2. Novelty Reranker

   In the Novelty Track in TREC 2002 (http://trec.nist.gov), users were asked to identify sentences which contain new information, as sentences are passed sequentially through the system. We noticed that human judges often pick clusters of sentences, whereas the default-reranker normally does not care about the spatial relationships between sentences within a document. To exploit this hunch, there is a small modification made to the default-reranker which boosts the sentence ranking if the sentences occuring just before it in the document were selected by the reranker.

3. Update Reranker

   This reranker module is designed for the Update Summarization task which was introduced in DUC 2007. This reranker essentially reduces the score of those individual sentenes which are same or similar to the ones already presented to the user in the previous summaries. This module uses cosile similarity feature to measure the similarity between already presented summary sentences and the current one.

4.10 Postprocessor

Postprocessing involves several tasks like removing unnecessary phrases and words from the summary because generally the most important thing associated with summaries is the clustering of information with as little of unnecessary things as possible. So, we want to prune the sentences which were selected by the reranker to take out only the important parts of those in the summary/extract. This will allow us to include more sentences in the summary to increase the information content.

4.11 Examples

Lexrank Summary

You may wonder why I would write a health column about malaria when there is no malaria in the United States.In fact, Ruebush said, 90 percent of all malaria infections and 90 percentof malaria deaths occur in Africa. "World spending on malaria control and research for Africa ismaybe 10 cents per case per year," said Sachs. "It's quitedreadful. World Bank lending for malaria is de minimus. The bigpharmaceutical companies see it as a disease of the very poor, sothey never view it as much of an investment priority." MANILA, November 26 (Xinhua) – The Philippines has made a bigstride in malaria control with malaria infections rate in thecountry is now generally low, a senior health official said today. That would help develop a practical malaria control strategyin all malaria endemic countries by the year 2005. The malaria problem is increasing because the malaria parasitehas developed resistance to some of the anti-malaria drugs andinsecticides. It is estimated that 300 to 500 million clinical cases and 1.5to 2.7 million deaths occur due to malaria each year, about twiceas

many as 20 years ago, according to papers presented at thethird Pan-African Conference on Malaria which ended here Wednesday. "There is urgent need for research into new malaria compounds"and more urgent need for enhanced joint efforts especially byAfrican countries to fight the disease, they said. Sub-Saharan Africa hosts more than 90 percent of Malaria victims. They established a working group to investigate how to securefunds for malaria control plans and made recommendations on keyareas in malaria prevention, treatment and control, according tothe statement.

Centroid Summary

As Dr. Robert S. Desowitz, an expert in tropical diseases,explains in his engrossing book "The Malaria Capers" (W.W.Norton, 1991), a malaria infection in humans begins when aninfected female Anopheles mosquito, seeking a blood meal to fosterthe development of her eggs, injects into the human bloodstreamthreadlike malaria parasites called sporozoites that have beenstored in her salivary glands.Malaria, which is reaching epidemic proportions in Africa andparts of Asia, Latin America and the southern fringe of the formerSoviet Union, kills about a million people a year, and children areespecially vulnerable. Experts say one child dies of malaria every30 seconds. Around the world, malaria kills 3,000 children under 5every day, a higher mortality rate than AIDS. "World spending on malaria control and research for Africa ismaybe 10 cents per case per year," said Sachs. "It's quitedreadful. World Bank lending for malaria is de minimus. The bigpharmaceutical companies see it as a disease of the very poor, sothey never view it as much of an investment priority." The malaria problem is increasing because the malaria parasitehas developed resistance to some of the anti-malaria drugs andinsecticides. "The economic consequences of malaria-related diseases areenormous. The direct and indirect losses due to malaria in theregion rose from 800 million U.S. Dollars in 1987 to more than2,000 million U.S. Dollars in 1997," the statement said.

Combined Summary

In fact,Ruebush said, 90 percent of all malaria infections and 90 percentof malaria deaths occur in Africa. Malaria, which is reaching epidemic proportions in Africa andparts of Asia, Latin America and the southern fringe of the formerSoviet Union, kills about a million people a year, and children areespecially vulnerable. Experts say one child dies of malaria every30 seconds. Around the world, malaria kills 3,000 children under 5every day, a higher mortality rate than AIDS. "World spending

on malaria control and research for Africa ismaybe 10 cents per case per year," said Sachs. "It's quitedreadful. World Bank lending for malaria is de minimus. The bigpharmaceutical companies see it as a disease of the very poor, sothey never view it as much of an investment priority." MANILA, November 26 (Xinhua) – The Philippines has made a bigstride in malaria control with malaria infections rate in thecountry is now generally low, a senior health official said today. That would help develop a practical malaria control strategyin all malaria endemic countries by the year 2005. The malaria problem is increasing because the malaria parasitehas developed resistance to some of the anti-malaria drugs andinsecticides. They established a working group to investigate how to securefunds for malaria control plans and made recommendations on keyareas in malaria prevention, treatment and control, according tothe statement.

Chapter 5

OPINION SUMMARIZATION

In this chapter, we have talked about the methodology that we have used for the opinion summarization problem. An *opinion* is a person's ideas and thoughts towards something. It is an assessment, judgment or evaluation of something. An opinion is not a fact, because opinions are either not falsifiable, or the opinion has not been proven or verified. If it later becomes proven or verified, it is no longer an opinion, but a fact. In economics, philosophy, or other social sciences, analysis based on opinions is referred to as normative analysis (what ought to be), as opposed to positive analysis, which is based on scientific observation (what materially is). In today's world, people share their opinions in forums, blogs, news articles, discussion platforms etc. This knowledge can be used to understand the bevaior and likes-dislikes of a set of people (even a single user).

Opinion summarization summarizes opinions of articles by telling sentiment polarities, degree and correlated events. Here we discuss our system which tries to identify and analyze opininated sentences to generate a summary in some specific format.

5.1 System Architecture

We have decomposed the problem of opinion summarization into following steps:

- Subjectivity Analysis
  Identifying expression of opinions, emotions, evaluations, sentiments, speculations, uncertainty etc. in natural language. For the case of product reviews, this step can be further subdivided into two steps:
  - Feature Extraction
    Identifying specific attributes or features of the product.

  - Opinion Identification
    Identifying sentences which are likely to contain opinions or subjective expressions about the product or any feature of the product.

- Polarity Classification
  Identifying the polarity - postive, negative or neutral - for each opinion sentence by looking at the modifiers and specific sentiment words.

- Summary Extraction
  Generating a short summary from the set of reviews for the given product (may be for any specific feature or topic).

5.2 Data Collection

For the purpose of Opinion summarization, we collected data from two sources: *amazon.com* and manually annotated Customer Review Datasets (M. Hu and Bing Liu). Data from manually annotated set is mainly used for evaluating our approach. *amazon.com* includes a large database of publicized consumer reviews for a diverse range of products. The reviews on *amazon.com* are mostly unstructured i.e.; the users are not forced to write the review in any specified format. Hence, the customer review data from *amazon.com* poses more challenges for the information extraction task. We have collected data mainly for digital cameras as they are very easily available. The purpose of selecting electronics products as topics of review for our study is to test our approach where the features are more or less well defined. We have collected 50 reviews each for a set of 10 products using our own amazon crawler. The average number of sentences for each product is 1056 in the collected data. This collected data is passed through a preprocessor which identifies the sentence boundaries within the article and formats each of the review in xml format. this converts the unstructured review into much more structured form which is easy to read and process.

The hand annotated data is also from *amazon.com* but only for 5 different products -

1. Digital camera: Canon G3
2. Digital camera: Nikon coolpix 4300

3. Celluar phone: Nokia 6610

4. MP3 player: Creative Labs Nomad Jukebox Zen Xtra 40GB

5. DVD player: Apex AD2600 Progressive-scan DVD player

In total there are 314 diffrent customer reviews and total number of 3944 sentences combined for all the reviews. All the sentences are annotated for features and the degree of polarity - positive or negative. The features for the sentences are not necessarily from the sentences themselves. We run our system on these annotated sentences to evaluate the quality of features extracted.

For identifying the polarity and training a model for sentiment analysis we have followed an approach similar to OpinionFinder. For the experiments we have used a lexicon of over 8,000 *subjectivity clues*. Subjectivity clues are words and phrases that may be used to express private states, i.e., they have subjective usages. Though the lexicon we ahve used consists of only single-word clues. This lexicon was created by expanding the original list of subjectivity clues from (Riloff and Wiebe, 2003). Words that are subjective in most contexts are marked strongly subjective (*strongsubj*), and those thatmay only have certain subjetive usages were marked weakly subjective (*weaksubj*). Each clue has also been assigned a prior polarity, either *positive*, *negative*, *both* or *none*. By far, the majority of clues, 92.8%, are marked as having either positive or negative prior polarity and 0.3% are marked as both. 6.9% of the clues are marked neutral in the lexicon.

5.3 Subjectivity Analysis

In general, subjectivity analysis is the proess of automatically identifying when opinions, sentiments, speculations, and other private states are present in text. This step aims to identify subjective sentences and to mark various aspects of the subjectivity in these sentences. Goal of this module is to develop a system capable of supporting other Natural Language Processing (NLP) applications by providing them with information about the subjectivity in documents. Of particular interest are question answering systems that focus on being able to answer opinion-oriented questions, such as the following:

How is Bush's decision not to ratify the Kyoto Protocol looked upon by Japan and other US allies?
How do the Chinese regard the human rights record of the United States?

To answer these types of questions, a system needs to be able to identify when opinions are expressed in text and who is expressing them. Other applications that would benefit from knowledge of subjective language include systems that summarize the various viewpoints in a document or that mine product reviews. Even typical fact-oriented applications, such as information extraction, can benefit from subjectivity analysis by filtering out opinionated sentences (Riloff et al., 2005).

Now as our work for opinion summarization is focussed more on product reviews, the subjectivity analysis step can be subdivided into -

- Feature Extraction
- Opinion Identification

In the following sections we discuss about the methodlogy that we have followed for finding features from the prodcut reviews and using them to identify subjective sentences.

5.3.1 Feature Extraction

A set of good features/keyphrases (words or nominal compounds of great significance in a text) is a very important part as it works as an alternative representation for documents. Based on inforamtion theory (Shannon, 1948), the information content of a concept c is the negative log likelihood - $logp(c)$, where $p(c)$ is the probability of encountering an instance of concept c. As this probability increases, the informativeness decreases i.e.; a general concept is more frequent than a specific one over a large set of documents. The task of extracting keyphrases from a text consists of selecting salient words and multi-word units, generally noun compounds no longer than a threshold, from an input document. We have tried to rank and filter out the list of features or keyphrases by using a measure called *Pointwise Mutual Inforamtion* (PMI) (Turney, 2001) which relies on probabilities estimated in accordance with the co-occurence behavior of these keyphrases in the context of the topic. Various auto-

matic keyphrase extraction techniques have been discussed in literature e.g. (Turney, 1999) and systems like Extractor, Kea (Frank *et al.*, 1999; Witten *et al.*, 1999) and NPSeeker (Barker and Cornacchia, 2000).

Given a product class C with instances I and reviews R, the aim of this step is to find a set of (feature, opinions) tuples $\{(f, o_i, ..., o_j)\}$ such that $f \epsilon F$ and $o_i, ..., o_j \epsilon O$, where:

a) F is the set of product class features in R.

b) O is the set of opinion sentences in R.

c) f is a feature of a particular product instance.

d) o is an opinion sentence about f.

The solution to this feature extraction is discussed here in this section. We identified the important steps that our algorithm runs throguh and here is a deatailed explanation for those.

POS Tagger

Part-of-speech tagging (POS tagging or POST), also called grammatical tagging, is the process of marking up the words in a text as corresponding to a particular part of speech, based on both its definition, as well as its contexti.e., relationship with adjacent and related words in a phrase, sentence, or paragraph. We have used the **LingPipe** toolkit for the purpose of part-of-speech tagging. Like the other statistical packages in LingPipe (e.g. named entity detection, language-model classifiers, spelling correction, etc.), part-of-speech labeling is based on statistical models that are trained from a corpus of labeled data. We trained the tagger on Brown Corpus which consists of 1.1M tokens and tagged using 93 different POS tags.

In the example shown in figure 5.1, the opinionated phrases are bold. By studying these phrases and the corresponding pos-tag patterns, we found out that generally noun phrases are the ones which finally turn out to be the features for the products. We looked for the patterns which matched opinionated phrases in different types of

*in my opinion it's the **best camera** for the money if you're looking for something that's **easy to use**, small good for travel, and provides **excellent, sharp images**.*
IN/in PP$/my NN/opinion PPS/it '/' VBZ/s AT/the JJT/best NN/camera IN/for AT/the NN/money CS/if PPSS/you '/' RB/re VBG/looking IN/for PN/something WPS/that '/' PN$/s JJ/easy TO/to VB/use ,/, JJ/small NN/good IN/for NN/travel ,/, CC/and VBZ/provides JJ/excellent ,/, JJ/sharp NNS/images ./.

Figure 5.1: Example showing the tag patterns to look for

sentences present in thr product reviews. Different nouns (which have POS tags like NNP, NNS, NN etc.) and a conjunction of nouns alogn with some adjective modifier when present in the text clearly signifies the presence of a possible feature of the product as some opinion is clearly expressed in that sentence. A regular expression pattern can be formulated as $(JJ)*(NN?)(IN\ NN?)*(NN?)*$, where JJ is pos-tag for adjectives, NN? represents pos-tags for different kind of nouns.

Discriminator phrases

To understand the use of *Discriminator phrases* we need to first understand what are the features of any product and how are they related to the product itself. As we have mainly focussed on consumer electronics products as of now, so the features are the phrases which can be a *part* of the product or a *property* of the product, in turn, their parts and properties and so on. There are some *Related Concepts* also which cannot be directly related to the product as such. Table 3.1 shows some of the explicit features' examples from the common reviews for a digital camera. Now, these features occur generally in reviews alongwith the product class name (*camera* in the above case). The phrases which connect or act as a conjunction between the product class name and the feature name are termed as *discriminator phrases*. This relation is somewhat similar to *Meronymy* which defines a semantic relation between two objects if one is a part of the other. Now, we use a manually crafted list of *meronymy discriminators* associated with the product class for the web PMI score calculation (which is defined in detail in the next section). Examples of these discriminator phrases for the product

Table 5.1: Explicit Features Examples

Explicit Features	**Examples**
Properties	CameraSize
Parts	CameraLens
Features of Parts	LensZoom
Related Concepts	Image
Related Concepts' Features	ImageResolution

class **camera** are - "of camera", "camera has", "camera's", "camera comes with", etc. Different discriminator phrases are defined for different product classes and these can be easily extended or modified for different product classes.

Explicit Feature Ontology

First we try to find out the features for the product class given a set of reviews. This step is unsupervised and some extraction patterns are used to achieve this. But, the unsupervised automatic feature extraction makes some assumptions and in actual, guesses some irelevant phrases. The web-PMI scores also don't help in pruning these out as some phrases are pretty common and co-occuring.

This camera is perfect for an enthusiastic amateur photographer.

In the above sentence, clearly the author is talking about the camera in general and actually expressing positive oppinion about it. But, when the automatic feature extracor is run , it identifies *camera* as well as *photographer* as the possible featues of the camera. So, as a solution to this problem we try to make the system a bit supervised, though not completely.

Web PMI

Pointwise mutual information (PMI) (or specific mutual information) is a measure of association used in information theory and statistics. To understand the measure better, we take an dexample - given the problem word *levied* and the four alternative words *imposed*, *believed*, *requested*, *correlated*, which of the alternatives is most similar in meaning to the problem word. Let problem represent the problem word and

camera
lens
focus , autofocus
exposure
aperture
shutter , speed , time
resolution , mp , megapixel
memory , memorycard , mb , megabyte
weight , heavy , light
size , big , huge , small , large
price , money , dollar , rupee , Rs
frame , framerate
feature
software
...

Figure 5.2: A portion of the sample feature ontology for digital cameras

$choice_1, choice_2, ..., choice_n$ represent the alternatives. The PMI-IR algorithm assigns a score to each choice, $score(choice_i)$, and selects the choice that maximizes the score. Clearly, the PMI-IR algorithm is based on co-occurrence. The core idea is that *"a word is characterized by the company it keeps"*. There are many different measures of the degree to which two words co-occur. PMI-IR uses Pointwise Mutual Information (PMI) as follows:

$$score(choice_i) = log_2 \frac{p(problem \,\&\, choice_i)}{p(problem) * p(choice_i)} \tag{5.1}$$

Here, $p(problem \,\&\, choice_i)$ is the probability that $problem$ and $choice_i$ co-occur. If $problem$ and $choice_i$ are statistically independent, then the probability that they co-occur is given by the product $p(problem) * p(choice_i)$. If they are not independent, and they have a tendency to co-occur, then $p(problem \,\&\, choice_i)$ will be greater than $p(problem) * p(choice_i)$. Therefore the ratio between $p(problem \,\&\, choice_i)$ and $p(problem)p(choice_i)$ is a measure of the degree of statistical dependence between $problem$ and $choice_i$. The log of this ratio is the amount of information that we ac-

quire about the presence of *problem* when we observe $choice_i$. Since the equation is symmetrical, it is also the amount of information that we acquire about the presence of $choice_i$ when we observe *problem*, which explains the term mutual information. In our implementation, we have used the power of web to compute *pointwise mutual inforamtion* metric between each fact and discriminator phrases which were discussed in above sections. Hence for our case, given a feature f and discriminator d, the computed Web PMI score is:

$$PMI(f, d) = \frac{Hits(d + f)}{Hits(d) * Hits(f)} \tag{5.2}$$

where, *Hits(x)* is the number of search results hit by the search engine given the search query x. We have used yahoo web APIs to query the web and to get the search result counts.

Algorithm

This paragraph explains the feature extraction algorithm that we have used for automatic keyphrase extraction and filtering those keywords using the explicit feature ontology (if provided). The given outline is for the document level feature extraction i.e., given a product review, the following steps identify and extract the possible list of features of the product which were talked about in the review.

At line 1, the document is preprocessed and sentences are broken apart by the system. This step is done only for identifying where the features are anchored in the review. This way we create a mapping between the extracted features and the sentences in the document. Lines 2-19 details out the flow of the algorithm for each sentence present in the review. At first the sentence is tagged using LingPipe POS Tagger as explained in the previous sections. Then the manually learned patterns are applied on the pos-tagged sentence to extract the unigram and multi-word possible feature words and phrases respectively. Now at this step, due to some error in POStagger and some assumptions made during recognizing the tag patterns, we end up getting some outliers. At this step we remove some of the stopwords using an external list and also some outliers. If there is a explicit feature ontology is provided for the product class, then we filter the features extracted till this point by matching them with the ones

present in the ontology provided. The matching in the lines 8-10 is done by checking for any morphological changes if present.

FeatureExtractor [Document level]

1. Identify sentence boundaries
2. **for** every sentence
3. tag each word in the sentence with its corresponding *part-of-speech*
4. find the tag patterns in the sentence
5. select the possible unigram features
6. select the possible multi-word features
7. remove stopwords and outliers
8. **if** external *feature-list* provided for this product class
9. filter the possible list of features to get a more precise list using the hierarchial feature information provided in the *feature-list*
10. **endif**
11. **for** each feature f extracted
12. **for** each discriminator d phrase
13. calculate the web PMI score as
14. $pmi(f, d) = hits(d + f)/hits(d) * hits(f)$
15. **end loop**
16. $pmi(f) = \max_{d}(pmi(f, d))$
17. **end loop**
18. rank the features according to PMI score and select the features above the threshold
19. **end loop**

Figure 5.3: Algorithm outlining automatic Feature Extraction

Lines 11-17 compute the web PMI scores for different discriminator phrases and

the features extracted till this step. At the end, we rank the features extracted till this point according to the web PMI scores.

At the end, we get a set of features extracted from a review which are filtered using a web PMI ranking and an explicit feature ontology, if at all provided. The following section lists down some example sentences and the extracted features for them.

Examples

1. In my opinion it 's the best camera for the money if you 're looking for something that 's easy to use , small good for travel , and provides excellent , sharp images .
Extracted features : camera[camera], money[price], images[image]

2. the auto-mode is good enough for most shots but the 4300 also boasts 12 versatile scene modes as well as a manual mode though i admit i have n't played with it too much on manual .
Extracted features : scene modes[scene, mode], auto mode[mode]
, mode[mode]

3. awesome camera with huge print quality in a tiny package .
Extracted features : camera[camera],print quality[image]

Note: the features are represented as feat[feat'], where *feat'* is a specific or specialized form of *feat*

5.3.2 Opinion Identification

The goal of opinion identification is to detect where in the documents opinions are embedded. An opinion sentence is the smallest complete semantic unit from which opinions can be extracted. The sentiment words, the opinion holders, and the contextual inforamtion should be considered as clues when extracting opinion sentences and determining their tendencies. As in the previous step we identify the feature

terms or phrases of the document class, we use this extracted information to identify the sentences which contain or might contain useful information about those features. Out intuition is that an opinion phrase associated with a product feature will occur in its vicinity. This idea is similar to (Kim and Hovy, 2004), (Hu and Liu, 2004) and (Popescu and Etzioni, 2005). We use the extra information present in a window of a fixed size ending on the word in any particular sentence. We have extracted some tag patterns such that we get to know the sentiment words also if they occur in vicinity of the feature word. But that's not always the case, which takes us to another step of polarity or sentiment identification. At the end of this step, we have the sentences which are most probable of having some opinion expressions about the feature terms of the product class.

5.3.3 Examples

1. i love the continuous shot mode , which allows you to take up to 16 pix in rapid succession – great for action shots .
2. yes , the picture quality and features which are too numerous to mention are unmatched for any camera in this price range .
3. there are so many functions in this little , yet powerful camera !

5.4 Polarity Analyzer

Sentiment Analysis or polarity classification is the task of identifying positive and negative opinions, emotions and evaluations. Some example sentences are:

1. African observers *generally approved*$^{+}$ of his victory while Western governments *denounced*$^{-}$ it.
2. A succession of officers filled the TV screen to say that they *supported*$^{+}$ the people and that killings were *not tolerable*$^{-}$.

A typical approach to sentiment analysis is to start with a lexicon of positive and negative words and phrases. In these lexicons, entries are tagged with their a priori *prior polarity.*, which is out of context and is jsut a measure of whether the word seems to evoke something positive or negative. For example, *beautiful* has a positive

prior polarity, and *horrid* has a negative prior polarity. However, the contextual polarity of the phrase in which a word appears may be different from the word's prior polarity. In the example sentence in figure 5.4, "Trust", "well", "reason" and "reasonable" have positive prior polarity, but they are not all being used here to

Philip Clap, President of the National Environment Trust, sums up well the general thrust of the reaction of environmental movements: there is no reason at all to believe that the polluters are suddenly going to become reasonable.

Figure 5.4: Example showing contextual sentiment disambiguation

express positive sentiments. Hence, many things need to be considered in phrase level sentiment analysis. In the following paragraphs we explain the method used and some of the approaches taken for incorporating contextual knowledge to identify the correct sentiment.

PolarityAnalyzer [Sentence level]
Input : Sentence S, Lexicon of positive and negative sentiment words L
Output : Polarity of sentences
$\forall i \, \epsilon \, S, \;\; ifPolar(i, L) \rightarrow addToPolarSet(P, i)$
$\forall i \, \epsilon \, S, \;\; ifNotPolar(i, L) \rightarrow addToNonPolarSet(NP, i)$
$\forall p \, \epsilon \, P, \;\; findContextualPolarity(p) \rightarrow Polarity(p)$

Figure 5.5: Polarity Analyzer *Overview*

5.4.1 Prior polarity classification

This step uses a list of words with known semantic orientation as a starting point. These words are assigned their most common polarity and this works as the prior polarity for these words. At the first step a classifier just assumes that a word's polarity is same as its prior polarity and tries to classify the word as either neutral or polar (positive or negative). In the literature various observations have been made

about these polar classified words - words with non-neutral polarity frequently appear in neutral contexts. Some times words occur in some other context also (probably with some different meaning) and sometimes modifiers deviate them from their prior semantic orientation. Hence we incorporate a second step which tries to classify the polar-marked words into positive, negative or neutral categories.

The lens is a lot better and the 4mb produce fantastic pictures.
better : POSITIVE
fantastic : POSITIVE

Figure 5.6: Prior polarity classification

5.4.2 Contextual polarity classification

This step uses various features which are observed for the opinion sentence. **Word token** and **word prior polarity** are simple features which are used at this classification step. **Negated** is a binary feature that captures whether the word is being locally negated: its value is true if a negation word or phrase is found within a window of the *four* preceeding words or in any of the words children in the dependency tree, and if the negation word is not in a phrase that intensifies rather than negates (e.g., not only). The **negated subject** feature is true if the subject of the clause containing the word is negated. The **modifies polarity**, **modified by polarity**, and **conj polarity** features capture specific relationships between the word instance and other polarity words it may be related to. If the word and its parent in the dependency tree share an obj, adj, mod, or vmod relationship, the *modifies polarity* feature is set to the prior polarity of the words parent (if the parent is not in our prior-polarity lexicon, its prior polarity is set to neutral). The *modified by polarity* feature is similar, looking for adj, mod, and vmod relationships and polarity clues within the words children. The *conj polarity* feature determines if the word is in a conjunction. If so, the value of this feature is its siblings prior polarity (as above, if the sibling is not in the lexicon, its prior polarity is neutral).

Its easy to focus on the drawbacks but that does not mean i hate this camera.
drawbacks : NEGATIVE
hate : POSITIVE

Figure 5.7: Contextual polarity classification

5.4.3 Sentiment Cumulation

This step computes the sentiment orientation of the complete sentence by looking at the different sentiment carrying words present in the sentence. WE have used a simple heuristic to merge the individual sentiments carried by differnet words of the sentence. For any subjective sentiment which talks about at most one feature of the product class, we just count the number of POSITIVE, NEGATIVE and NEUTRAL sentiment words and if number of POSITIVE words is greater than that of NEGATIVE words, the sentence is marked POSITIVE. Similarly, if number of NEGATIVE words is greater than that of POSITIVE words, the sentence is marked NEGATIVE, otherwise, if the count is same for both, sentence is marked as BOTH. For sentences which talk about more than features of the product, we have just made an approximation by looking at the sentiment carrying words occuring before the feature term in the sentence. Though this can be further improved and more rules to work around these cases can be incorporated.

With nikon, although picture qualities are as good as any other 4 mp cameras, i've had the following headaches. good : POSITIVE
headaches : NEGATIVE
sentiment assigned : **BOTH**

Figure 5.8: Sentiment cumulated for the sentence

5.5 Summarizer

Traditional Summarization algorithms rely on the important facts of documents and remove the redundant information. Unlike the general techniques, two factors - say, the sentiment degree and the correlated events, play the major roles of opinions summarization. The repeated opinions of the same polarity cannot be dropped beacause they strengthen the sentiment degree. However, the redundant reasons of why they hold this position should be removed while producing the summaries. This step aims to produce a cross-document summary and at the previous step we know the opinionated sentences and the specific features they talk about, we can gather all the opinionated information from the corpus on a specific given topic. Two different types of summaries can be seen useful in case of product reviews - one where a query/topic is provided and the summary contains the opinionated sentences on that topic only and second, where a combined summary on all the different features of the product are summarized.

News and blog articles are also important sources of opinions. Generally speaking, news articles are more objective while blogs are usually more subjective. We have done some experiments on the TREC blog data as well to see the how this summarization model performs. A major differnece in summarization for product reviews and blogs/news comes at the subjectivity analysis phase. In reviews, subjectivity is found by identifying the features of the product - either independently or using an external ontology. Whereas, in case of blogs or news articles subjectivity finding step mailny relies on presence of opinion identification phrases.

Chapter 6

RESULTS AND EVALUATION

6.1 Multi-Document Summarization

Here we present an example run of the update summarization algorithm on a sample data set on *malaria*. The dataset consisted of 50 documents. The first set of sentences is the summary or a ranked list of the dataset normally (using the lexrank and centroid feature scripts and a default classifier). The second set is the *update* summary relative to the first summary i.e., we assume that the user has already read the first summary and so we try to extract some new information other than already presented to the user.

Normal Summary

1) In fact, Ruebush said, 90 percent of all malaria infections and 90 percent of malaria deaths occur in Africa.
2) Malaria, which is reaching epidemic proportions in Africa and parts of Asia, Latin America and the southern fringe of the former Soviet Union, kills about a million people a year, and children are especially vulnerable. Experts say one child dies of malaria every30 seconds. Around the world, malaria kills 3,000 children under 5every day, a higher mortality rate than AIDS.
3) "World spending on malaria control and research for Africa is maybe 10 cents per case per year," said Sachs. "It's quite dreadful. World Bank lending for malaria is de minimus. The big pharmaceutical companies see it as a disease of the very poor, so they never view it as much of an investment priority."
4) MANILA, November 26 (Xinhua) – The Philippines has made a big stride in malaria control with malaria infections rate in the country is now generally low, a senior health official said today.
5) That would help develop a practical malaria control strategy in all malaria endemic countries by the year 2005.

6) The malaria problem is increasing because the malaria parasite has developed resistance to some of the anti-malaria drugs and insecticides.
7) They established a working group to investigate how to secure funds for malaria control plans and made recommendations on key areas in malaria prevention, treatment and control, according to the statement.

Update Summary

1) Malaria kills up to 3 million people a year and sickens another 300 million. Creating a vaccine is crucial because the parasite has begun developing resistance to drugs used to treat malaria, and even mosquitos that spread the disease are withstanding pesticides.
2) You may wonder why I would write a health column about malaria when there is no malaria in the United States.
3) This year, the health care service plans to promote health care awareness, teach people how to prevent malaria by themselves, help village medical stations to detect malaria patients, and provide mosquito-nets to all people in remote, isolated and mountainous areas.
4) It is estimated that 300 to 500 million clinical cases and 1.5to 2.7 million deaths occur due to malaria each year, about twice as many as 20 years ago, according to papers presented at the third Pan-African Conference on Malaria which ended here Wednesday.
5) Annual number of deaths of children under five years of age attributed to malaria hits one million, the experts said, "During the last 10 years, malaria has killed 10 times more children than all the wars that have raged over the same period."
6) "There is urgent need for research into new malaria compounds" and more urgent need for enhanced joint efforts especially by African countries to fight the disease, they said. Sub-Saharan Africa hosts more than 90 percent of Malaria victims.
7) At the end of the summit, heads of state will issue a declaration on tackling malaria in Africa and new statistics on the crippling effect malaria has on economic development in African countries will also be launched.

6.1.1 ROUGE

There are various metrics present which can be used to evaluate summarization systems. We have used a version of **ROUGE** (Recall-oriented Understudy for Gisting

Evaluation) for evaluating the summaries generated by our system. ROUGE is a N-gram based evaluation metric which can be used to measure the similarity between two summaries both - *precision-wise* and *recall-wise*.

$$C_n = \frac{\sum_{C\epsilon\{ModelUnits\}} \sum_{n-gramsεC} Count_{match}(n-gram)}{\sum_{C\epsilon\{ModelUnits\}} \sum_{n-gramsεC} Count(n-gram)} \tag{6.1}$$

Where C_n is the score of nth sentence, Count match (n-gram) is the number of n-grams matched between the peer summary and the reference summary where as Count(n-gram) is the number of n-grams present in each of the model-units. Model-units can be defined as sentences present in the model summary. This metric is a recall-based one. If the denominator is changed to consider the sentences present in the peer summary instead of the reference summary, it will become a precision-based metric.

$$p_n = \frac{\sum_{C\epsilon\{Candidates\}} \sum_{n-gramsεC} Count_{clip}(n-gram)}{\sum_{C\epsilon\{Candidates\}} \sum_{n-gramsεC} Count(n-gram)} \tag{6.2}$$

6.1.2 Simple Cosine Similarity

Simple cosine calculates the cosine similarity with a simple binary count (1 if a word exists (no matter how many times) in a sentence, 0 if it doesn't). Cosine uses idf weights and includes the actual count of tokens for each type. In "The quick brown fox jumped over the lazy dog," simple cosine would only count "the" as 1, while cosine would count it twice and multiply it by its idf weight.

6.1.3 Observations

Tables 6.1 to 6.5 show the perfromance of our system on a particular set of 25 documents on a single topic. We have assumed that the cluster on any topic doesn't contain any irrelevant document. We have tried to evaluate our system in three different scenarios - one with Centroid, one with Lexrank and one with the equally weighted combination of Centroid and Lexrank. Alongwith this, to get a better evaluation of the system, we also compared our system with two systems which were

Table 6.1: Rouge Evaluation : Centroid, Lexrank, Both vs two systems submitted in DUC06

		Reference 1	Reference 2	Reference 3	Reference 4
Rouge-1 (Precision based)	**Centroid**	**0.32**	**0.33**	**0.39**	**0.32**
	Lexrank	**0.346**	**0.361**	**0.38**	**0.319**
	Both	**0.32**	**0.33**	**0.36**	**0.312**
	Peer1	0.419	0.409	0.462	0.40
	Peer2	0.42	0.47	0.414	0.55
Rouge-1 (Recall based)	**Centroid**	**0.36**	**0.37**	**0.44**	**0.36**
	Lexrank	**0.349**	**0.389**	**0.42**	**0.376**
	Both	**0.395**	**0.426**	**0.464**	**0.408**
	Peer1	0.201	0.228	0.20	0.20
	Peer2	0.411	0.486	0.452	0.60
Rouge-2 (Precision based)	**Centroid**	**0.024**	**0.024**	**0.043**	**0.032**
	Lexrank	**0.027**	**0.036**	**0.039**	**0.035**
	Both	**0.022**	**0.033**	**0.044**	**0.033**
	Peer1	0.019	0.038	0.013	0.006
	Peer2	0.042	0.114	0.059	0.203
Rouge-2 (Recall based)	**Centroid**	**0.030**	**0.025**	**0.051**	**0.029**
	Lexrank	**0.0303**	**0.038**	**0.055**	**0.038**
	Both	**0.035**	**0.042**	**0.068**	**0.042**
	Peer1	0.008	0.012	0.008	0.004
	Peer2	0.039	0.097	0.068	0.021

submitted to DUC 2006. The systems we selected, had *teamID* 1 and 24, where *teamID* 1 is a randomly selected system and *teamID* 24 was the system which came first in the DUC 2006 summarization task. All these summaries were evaluated for all 4 reference summaries available.

Table 6.2: *Simple* cosine similarity : Centroid, Lexrank, Both vs two systems submitted in DUC06

	Reference 1	Reference 2	Reference 3	Reference 4
Centroid	**0.15**	**0.16**	**0.21**	**0.19**
Lexrank	**0.164**	**0.171**	**0.207**	**0.179**
Both	**0.17**	**0.176**	**0.205**	**0.184**
Peer1	0.13	0.175	0.16	0.15
Peer2	0.215	0.25	0.23	0.37

Table 6.1 shows the ROUGE evaluation of all the 5 summaries against the 4 reference summaries. One clear observation is that the system which is based on the combination of Lexrank and Centroid feature scripts has a better recall score than

either of the Lexrank-based system or Centroid-based system. Our system doesn't work quite well in terms of ROUGE-1 precision scores and both the peer summarizers outperformed it but our system worked almost similar to the peer summarizer that came first in DUC06 in terms of ROUGE-1 recall score.

Table 6.3: Token Overlap : Centroid, Lexrank, Both vs two systems submitted in DUC06

	Reference 1	Reference 2	Reference 3	Reference 4
Centroid	**0.081**	**0.085**	**0.12**	**0.109**
Lexrank	**0.089**	**0.093**	**0.115**	**0.099**
Both	**0.092**	**0.096**	**0.114**	**0.101**
Peer1	0.074	0.094	0.085	0.080
Peer2	0.12	0.142	0.13	0.226

Table 6.4: Bigram Overlap : Centroid, Lexrank, Both vs two systems submitted in DUC06

	Reference 1	Reference 2	Reference 3	Reference 4
Centroid	**0.109**	**0.011**	**0.017**	**0.013**
Lexrank	**0.011**	**0.013**	**0.017**	**0.015**
Both	**0.010**	**0.013**	**0.019**	**0.014**
Peer1	0.005	0.07	0.005	0.025
Peer2	0.017	0.035	0.028	0.085

Table 6.5: Normalized Longest Common Substring : Centroid, Lexrank, Both vs two systems submitted in DUC06

	Reference 1	Reference 2	Reference 3	Reference 4
Centroid	**0.120**	**0.084**	**0.165**	**0.112**
Lexrank	**0.119**	**0.089**	**0.158**	**0.111**
Both	**0.116**	**0.089**	**0.158**	**0.112**
Peer1	0.106	0.065	0.118	0.089
Peer2	0.128	0.141	0.145	0.278

6.2 Update Summarization

Here is an example of the update summarizer when ran on a dataset of DUC06 on topic *malaria*. The set A consisted of 12 documents whereas set B consisted of 9 documents from a single cluster of DUC06 dataset. We assumed that set B comes later in the chornological order between A and B. The examples in above figures

Cluster A [D0618I-A] **Normal Summary**

[1] Malaria kills up to 3 million people a year and sickens another 300 million. Creating a vaccine is crucial because the parasite has begun developing resistance to drugs used to treat malaria, and even mosquitos that spread the disease are withstanding pesticides.
[2] Malaria is not endemic in this country or to Canada or Europe. But one-third of the world's population lives where the malaria parasite and its carrier mosquitoes thrive, and every year more than 1 million Americans travel to those areas for business or pleasure. Malaria, it seems, is gaining ground annually as control efforts become more costly and cumbersome.
[3] This year, the health care service plans to promote health are awareness, teach people how to prevent malaria by themselves, help village medical stations to detect malaria patients, and provide mosquito-nets to all people in remote, isolated and mountainous areas.
[4] The Yunnan Institute of Malaria Prevention, China's only malaria research organ of its kind, established in 1957, has been running classes in the past 40 years and has trained more than2,000 personnel in malaria prevention and treatment.
[5] Based on satellite mapping and climatic information, the distribution of malaria can now be determined at the community level and the information will benefit national and international efforts for malaria control.

Figure 6.1: Normal Summary generated with Lexrank and Centroid features for the documents in set A

show one step of update summarization. Clearly, the inforamtion which was present in the summary generated for set A are not present in the summary for the set B.

We were not able to evaluate update summarization module as we had no benchmark summaries as this task was only recently introduced. By doing a manual quality evaluation of the summarizer we found out that it works fine in general. But, we were not able to quantify the accuracy and performance of the system.

Cluster B [D0618I-B] **Update Summary (after A has been read)**

[1] Nabarro, now strategic director for human development at Britain's Department for International Development, said in an interview Friday that "collective brainpower" was necessary if health officials had any hope of even halving malaria deaths in a decade. He said wiping out malaria is an impossible goal.
[2] "World spending on malaria control and research for Africa is maybe 10 cents per case per year," said Sachs. "It's quite dreadful. World Bank lending for malaria is de minimus. The big pharmaceutical companies see it as a disease of the very poor, so they never view it as much of an investment priority."
[3] MANILA, November 26 (Xinhua) – The Philippines has made a big stride in malaria control with malaria infections rate in the country is now generally low, a senior health official said today.
[4] That would help develop a practical malaria control strategy in all malaria endemic countries by the year 2005.
[5] The malaria problem is increasing because the malaria parasite has developed resistance to some of the anti-malaria drugs and insecticides.

Figure 6.2: Update Summary generated with Lexrank and Centroid features for the documents in set B after the previous summary has been read

6.3 Opinion Summarization

6.3.1 Subjectivity Analysis

The components of Subjectivity Anlaysis steps are - Feature Extractor and Opinion Identifier. Now, the evaluation of quality of features or keyphrases is an intricate and subjective task (Barker and Cornacchia, 2000). The standard evaluation technique is to compare the overlap between the set of automatically identified keyphrases and a list of human generated ones (Turney, 2003; Frank *et al.*, 1999; Witten *et al.*, 1999). This is is a bit problematic in the sense that the author will provide keyphrases that are not found in the article directly whereas the system will only extract features from the text.

The Feature Extraction module, when run on a set of 34 reviews for a single camera identified 562 features (174 unique features) whereas the test system of Hu and Liu had identified a total of 389 (115 unique features). Our Feature extraction clearly outperforms with a feature per sentence ratio of 1.624 against Hu and Liu's benchmark data which has 1.12 features per sentence on an average. Now, we tried to evaluate our feature extractor's recall and precision on the manually annotated data that we had. The data was for subjectivity analysis as it had all those sentences annotated

Table 6.6: Evaluation on Nikon review set: Subjectivity Analysis

	Test set	Our system
Number of Sentences	346	346
Number of *Subjective* sentences	186	198
Number of Features	389	562
Number of Unique Features	115	174
Number of sentences wrongly classified as *subjective*	0	49
Number of features wrongly selected	0	51

which the annotaters found are expressing some kind of opinion or sentiment about any feature. It doesn't cover those sentences where no opinion has been expressed but some feature of the product has been talked about. So, for the feature extractor step evaluation, we were not able to measure the actual recall of the system. Table 6.6 is the list of values we manually found out for the Nikon test dataset whereas, table 6.7 is the list of values we manually found out for the Canon test dataset.

By observing the above tables, we can say that the overall subjectivity analyzer works

Table 6.7: Evaluation on Canon review set : Subjectivity Analysis

	Test set	Our system
Number of Sentences	642	642
Number of *Subjective* sentences	359	397
Number of Features	644	996
Number of Unique Features	162	368
Number of sentences wrongly classified as *subjective*	0	113
Number of features wrongly selected	0	101

approximately with 85.8% and 82.4% recall values for nikon and canon datasets respectively. This calculation was made assuming the Hu and Liu's annotated dataset as a GOLD standard.

6.3.2 Polarity Analyzer

Again, we were not able to evaluate our sentence level polarity analyzer as we were not able to find any good benchamrk annotated data. Overall we have made some assumptions and approximations in the design of this module, which brings down the precision. But the recall is pretty decent as the prior polarity list (annotated MPQA corpus) is pretty extensive.

Chapter 7

CONCLUSION

In this project, we worked on two different aspects of multi-document summarization problem - update summarization and opinion summarization. We tried to explore some algorithms like centroid and lexrank and tried to see whether a combination of these can perform better or not. We also worked on subproblems of opinion summarization - automatic feature extraction and polarity analysis.

We found out that combined system of lexrank and dentroid features has a better recall than either of the one. This system doesn't works upto expectation on ROUGE-1 scale when compared with the other systems submitted in earlier DUCs. But, this system works well (quite close) on the ROUGE-2 scale when compare with the best system of DUC06.

The opinions summarization task is pretty new and thats the reason we weren't able to evaluate this system as a whole. but to get a beter idea we tried to evaluate the submodules of the system - feature extraction clearly outperforms the Hu and Liu's system and identifies the feature terms of the product quite accurately in those cases where an explicit list of feature terms is available. If that is not present, recall is observed to be fairly high at the cost of precision. We observe quite a few outliers, which is not something unexpected considering the unstructured and unscrutnized format of product reviews. We have tried to develop the system to particiapte in TAC 2008 opinion summarization task.

BIBLIOGRAPHY

[1] T. Wilson, J. Wiebe and P. Hoffman *Recognizing Contextual Polarity in Phrase-level Sentiment Analysis*, Proceedings of HLT/EMNLP 2005.

[2] Ku Lun-Wei, Liang Yu-Ting and Chen Hsin-Hsi *Opinion Extraction, Summarization and Tracking in News and Blog Corpora*, Proceedings of AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs.

[3] M. Jarmasz and C. Barriere,*Keyphrase Extraction : Enhancing Lists*, Proceedings of the Computational Linguistic in the North-East(CLINE) 2004.

[4] Ana-Maria Popescu and O. Etzioni, *Extracting Product Features and Opinions from Reviews*, Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, 2005.

[5] Soo-Min Kim and E. Hovy, *Automatic Identification of Pro and Con Reasons in Online Reviews* Proceedings of COLING/ACL Poster Sessions, 2006.

[6] G. Erkan and D.R. Radev, *Lexrank:graph-based Lexical Centrality as Salience in Text Summarization*, Journal of artificial Intelligence Research 2004.

[7] G. Erkan and D.R. Radev, *LexPageRank - Prestige in Multi-document text summarization*, In the proceedings of EMNLP, 2004.

[8] L. Vanderwende, H. Suzuki, C. Brockett and A. Nenkova, *Beyond SumBasic - task focused summarization with sentence simplification and lexical expansion*, Information processing and management 2007.

[9] W. Yih, J. Goodman, L. Vanderwende and H. Suzuki, *Multi-document summarization by maximizing informative content-words*, In the proceedings of IJCAI 2007.

[10] E. Hovy, C.Y. Lin, *Automated text summarization in SUMMARIST*, Advances in automatic test summarization, 1999.

[11] X. Zhu, A.B. Goldberg, J. Van Gael and D. Andrzejewski,*Imrpoving diversity in Ranking using Random Walks*, in the proceedings of NAACL HLT, 2007

[12] K. Toutanova, C. Brockett, J. Jagrlamudi and M. Gamon,*The PYTHY summarization system- MSR at DUC 2007*, Document understanding conference 2007.

[13] J. Jagadeesh, P. Pingali, V. varma,*Capturing sentence prior for Query-based multi-document summarization*, in the proceedings of DUC 2006.

[14] C.Y. Lin, *ROUGE- a package for automatic evaluation of summaries*, in the proceedings of the Workshop on text summarization branches in DUC. Wentau Yih, J. Goodman,L. Vanderwende and H. Suzuki, *Multi-document summarization by maximising informative content-words*, in the proceedings of the International Joint Conference on Aritifical Intelligence, 2007.