Towards Summarization of Written Text Conversations

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research)
in
Computer Science and Engineering

by

Arpit Sood
200802006
arpit.sood@research.iiit.ac.in

Search and Information Extraction Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2013

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Towards Summarization of Written Text Conversations" by Arpit Sood, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Vasudeva Varma

To anyone who has ever directly or indirectly helped someone

Acknowledgments

First and foremost, I would like to express my deep-felt gratitude towards my adviser Dr. Vasudeva Varma for his excellent guidance and invaluable support. His ideas and motivation have not only been a constant source of inspiration for me but also a paramount in shaping my ideologies and understanding. The interactions I have had with him have always succeeded in boosting my spirits. He trusted me at each step with great responsibilities which encouraged me to become self-dependent for work.

I take this opportunity to thank my seniors Kushal Dave, Aditya Mogadala, Niraj Kumar, Piyush Arora and Ankit Patil for the wonderful and stimulating discussions that I had with them, relating to not just research, but also different perspectives in life. It would be a crime if I do not mention Sudheer Kovelamudi, for he is such a great mentor and helped me a lot during the initial stages of my research.

I would also like to thank Thanvir Mohamed with whom I worked towards this thesis. This research would not have been possible without his continuous support, and interesting and insightful discussions I had with him. I had a great time working with him and Jeevan Shankar on different projects. Their calmness and simplicity taught me about patience and improved me as an individual. I also thank all my fellow researchers in SIEL for creating such a productive ambiance to work in.

I feel blessed to be in a group of really good friends. I would like to offer gratitude to my friends at IIIT Hyderabad Hemant Baid, Nachiket Bhagwat, Sarvesh Ranade, Aman Mahajan, Sumati Prabhakar and many others who never let me feel away from home. I will always cherish the time I spent with my friends Vyas Ram and Anirban watching football, followed by long discussions.

Finally, and most importantly, I would like to thank my father Mr. Anil Sood, my mother Mrs. Anuja Sood and my brother Akshit Sood for belief in my actions and the steps I took to achieve them. This thesis stands as a testimony to the freedom, encouragement and opportunities they have provided me throughout the life. I can never thank them enough for the sacrifices that they have made to provide me with this life.

Abstract

The immense growth of social media and web technologies have enabled users to create, share and exchange information in virtual communities and networks. The ease of usability has attracted considerable amount of users who collaborate on such social platforms. Content generated from such interactions and conversations contains a lot of information that could be of very good commercial and educational value. This necessitates the need to extract information from such conversations. Therefore, the task of information extraction has received lot of attention from industry as well as research community. However, the proportion of content generated is so huge that it leads to the problem of information overload and redundancy. These problems can be dealt appropriately by presenting crux of the content rather than the whole user-generated text. Automatic extraction of important topics, followed by summarization of the conversation may help in reducing considerable human effort to understand the content.

Automatic text summarization is a well-known solution to the problem of information overload [3, 5]. Summarization of news articles has been explored from theory to practical models, but summarization of user-generated content has not been paid much attention to. This can be attributed to the noisy and unstructured text from social media which makes traditional Natural Language Processing (NLP) techniques difficult to apply. For our purpose, we have chosen real-time conversations that are composed of emails and chats from technical domain. Such conversations suffer from a bigger problem of data sparseness as the relations between various entities or objects of discussion and their attributes is not explicitly present. In our research, we try to address these problems by designing two different approaches, a supervised machine learning approach and a topic-focused approach. The machine learning approach models the sentence as a set of features and builds a trainable summarizer using these feature vectors. Thereafter, in topic-focused approach we find hidden topics of the conversation using external resources and bind our summary to these topics using a semantic sentence scoring model.

In this thesis, we have mainly concentrated on extractive methods of summarization with sentence as a basic unit of the summary. This transforms our problem of summarization to "the selection of most representative and informative sentences". We design a supervised machine learning framework where each sentence is represented as a set of features. Along with the frequency and heuristics based methods, we devise new features in the form of Discourse Marker and Sentiment Score to capture the relevancy of a sentence more accurately. We have used Rhetorical Structure Theory (RST) to describe the organization of text. Our feature set aims to capture the statistical, linguistic and sentimental aspects along with the dialogue structure of the conversation. Conversations are modeled using a set of features which

represent sentences. Then we use various machine learning algorithms to build a trainable summarizer on these feature vectors. We use two different datasets and perform five fold cross validation of our approach. We conduct various set of experiments to show the robustness of our system. The inclusion of discourse marker and sentiment score feature leads to the generation of improved summaries. We apply different machine learning algorithms and show that our feature set is generic enough to learn from any of the algorithms. Our approach significantly outperforms the baselines on ROUGE F-scores.

We take a step forward towards our study of analyzing instantaneous conversations by following an unsupervised approach. In contrast to our earlier approach where we propose new features to capture different aspects of a conversation, here we develop a topic-focused approach based on the characteristics of a conversation. The social media conversations attract a substantial number of participants and a single conversation tends to span a wide range of topics interspersed with irrelevant segments. Motivated by this, we detect the primary topic of discussion in the conversation and build a semantic word space using the discovered topics. However, because of the sparseness in data identifying the connected components, primary topic detection becomes a difficult task. There is not enough evidence and explicit relationship present in the text to link different entities and objects with their corresponding attributes. To tackle this problem we develop a two phase approach. In the first phase, we leverage topic modeling using web documents to find the primary topic of discussion in the conversation. Then, in the summary generation phase, we use relevance-based language modeling approach to score sentences depending on their association with the primary topic. For topic modeling, we have used Latent Dirichlet Allocation (LDA) model. We use Hyperspace Language to Analogue (HAL) model approach to build a semantic word space that gives the association between word pairs. HAL is a semantic model based on the concept of distributional similarity that constructs the dependencies between words based on the context in which they appear.

Experiments are performed at both the topic detection stage as well as the summary generation stage to thoroughly evaluate the effectiveness of our approach. Results have shown that topic modeling with web referencing (LDA+WR) can discover the hidden topic structure more accurately. The primary topic detection process is very critical to our approach since the sentence scoring and whole summarization process is guided by the distribution of the primary topic. There is significant improvement in ROUGE scores when summaries are generated using Web for topic modeling. In our approach, we have incorporated topic information in HAL model for sentence scoring. We compare our results with strong baselines including HAL model to show that topic incorporation can lead to significant performance gain with ROUGE as the evaluation metric.

We go beyond the traditional notion of measuring relevance based only on statistics of the document. We devise new robust features to capture the sentiments and dialogue structure of the conversation. We develop a generalized framework for summarization of both chat and email conversations. We tackle the challenging problem of data sparsity by exploiting the semantics of the conversation. We have developed language and domain independent methods and the same work can be extended to blogs and twitter conversations. Our work is one of the most premier works in the area of conversation summarization.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The communication sector of the world has seen dramatic changes since the launch of World Wide Web (WWW) in 1991. Web 1.0 was an early stage of the conceptual evolution of the WWW. The users could only view webpages but not contribute to the content of the webpages. The content creators in Web 1.0 were few in number with the vast majority of users acting as consumers of content. However, with the advent of Social Web or Web 2.0, the amount of content present on the web has increased exponentially. Web 2.0 serves as a collaborative medium, where any user can share and exchange information. Web 2.0 allows users to do more than just information retrieval. It is associated with applications that provides base for information exchange, sharing and interoperability. Such a transition from Web 1.0 to Web 2.0 has attracted large number of users and has lead to information explosion.

It has facilitated the users with more storage capabilities, software and ease of interaction, all through their web browser. Network in Web 2.0 provides a platform for computing. It is a platform where anyone can come and contribute, therefore adding value to the application as they use it. Openness, usability, independence, collaboration and collective intelligence are some of the characteristics which makes it even more interactive and attractive. Various social networking sites, user-created forums for discussions, blogs etc. are all features of Web 2.0. Web has a reflection of the real world, therefore people tend to represent themselves in this virtual world using technologies and applications termed as social media, and use Internet as a medium of communication. Large number of mobile and web-based technologies are associated with social media leading to further growth and gain in its popularity.

With ever growing content on World Wide Web, it has become extremely difficult for users to search for useful information. Search engines, that are supposed to satisfy users information need, have too much information to offer than what is required. This problem is also known as information overload. In this context, it has become the need of the hour to develop information access solutions that can provide an easy and efficient access to users. Summarization is one of many such information access solutions that can address the problem of information overload. Automatic summarization system produces a summary of related social media text documents, thus providing an overall understanding of the topic without having to go through all of the text data.

1.1 Social Media and Social Interactions

Based on the ideological and technological foundations of Web 2.0, social media can be defined as a group of Internet-based applications that allows the users to create and exchange information. It gives means to people to create, share and exchange information in virtual communities and networks. Social media gives way to different kinds of social interactions in the form of Internet forums, weblogs, social blogs, wikis, social networks, microblogging, photographs or pictures, video sharing, rating and social bookmarking. Blogs, picture-sharing, vlogs, wall-postings, email, instant messaging, music-sharing, crowdsourcing and voice over IP are the various kinds of technologies used to support the previously mentioned social media interactions. These interactions are of a lot of commercial interest to organizations as well as the individuals, as one can propagate and share information with larger audience in a very convenient manner. Social media is relatively inexpensive compared to newspapers and television. It also attracts research community as such kind of interactions have lot of information in raw form which when extracted could be of very good educational value. Facebook[1], Google+[2], Stackoverflow[3] and Blogspot[4] are some examples of such social networks where user solely contributes to the data. The content in such kind of online social interactions belongs to the category of User-Generated Content.

1.2 User-Generated Content and Its Impact

User-generated content (UGC) is the material on websites, and various other media sources, that is produced by the users of the website. It covers different kinds of media content available in a range of modern communication technologies. There is no strict presentation format in social media, therefore it is easy to get acquainted with. Social media is accessible and affordable to the users which has also led to the exponential growth in the UGC. However, content is not created by field professionals but by the more general public. Freedom to contribute without any restriction is also responsible for UGC getting mixed up with noise.

The quality of user-generated content varies from extremely useful information to spam. The Organization for Economic Cooperation and Development (OECD) [56] defines UGC under the following conditions:

- The content is published and publicly accessible, even though to a small group of people.
- There must be collaborative element involved in creating the text involving many users. The content must present certain amount of *creativity*. Amount of creativity cannot be defined and depends on the context of the content. Here *creativity* means that the content should be produced

[1] `https://www.facebook.com/`
[2] `https://plus.google.com/`
[3] `http://stackoverflow.com/`
[4] `http://www.blogger.com/`

by the user, instead of copy-pasting it from another source. For example, blog of a user sharing his views, or uploading photos, creating new video, could be considered as UGC. Whereas posting a snippet from some site written by a professional does not fit in the category of UGC.

- The content is generally created outside of professional routines and practices.

In this thesis, we concentrate on user-generated content constituted from written text conversations which are carried out in the form Internet Relay Chats (IRC) and Emails.

1.3 Motivation

In the recent past, there has been a steep rise in the user activity and the number of users contributing to UGC. The communication of users through social media has seen an exponential increase. A substantial chunk of information exchange happens in the form of online conversations like Internet Relay Chats (IRC), Facebook and Twitter[5] streams. Among them, we focus on conversations from the support forums which aim to discuss and resolve user-related issues. Such forums contain a lot of information which can benefit organizations as well as information-seeking users [57]. However to make best use of these vast resources of information, we need to consider the time criticality of the user. This necessitates exploring methods of allowing users to search, locate and browse the required information readily from a collection of text conversations to avoid the problem of information overload. Apart from the problem of information overload, such forums also suffer from the problem of redundancy, where similar topic gets discussed multiple times by different users.

Summarization is a proven effective way to tackle these problems [3, 5]. An effective summary provides the main topics of discussion by removing redundant and unwanted information from the conversation. This saves users time by providing the essence of the document. Consider a scenario, where the user has a query and wants a certain kind of information. Since there is too much content present in the web, user takes assistance of a search engine which returns a set of documents. Although search engine reduces the solution space, but from user perspective there is still lot of effort involved in going through the retrieved set of documents to search for required information which leads to the problem of information overload. Therefore, to prevent oneself from information overload user posts a query, similar to an already existing one again which results in redundancy.

The retrieved documents are conversations which have a multiple participants and a single conversation tends to span a wide range of topics interspersed with irrelevant segments. In order to get the intent behind such long and involved conversation, summarization becomes essential. It prevents an individual from the problem of information overload as one does not need to go through whole document and a short concise summary will do the needful. It eradicates the problem of redundancy, as user can get the intent of the conversation like main topics of discussion by just looking at the summary. Therefore, if the summary satisfies users' information need, it will not be discussed again.

[5] `https://twitter.com/`

1.4 Challenges

User-generated content brings in lot of challenges at different levels, from smaller words units to bigger sentence units. The conversations belonging to the category of user-generated content can be distinguished from traditional writing like news articles on the basis of *form* and *content*. *Form* refers to how the linguistic act is constructed or perceived, while *content* refers to the substance of the linguistic message. In Table 1.1 we present some of the basic differences between a written text conversation and a regular document like news articles on the basis of *Form* and *Content*.

Text Conversation	**Regular Document**
dialogue (commonly incorporates feedback)	monologue (no immediate feedback)
real time	time independent
language grammar is not guaranteed	grammar is consistent
short phrases exist as sentences	small sentences may exist but no small phrases independently
less formal	more formal (no contradictions, proper subject-verb agreement)
noise because of non-standard emotional content words	no non-standard emotional content words exist
discourse level features	no discourse level features
NLP tools are not applicable, inconsistent	NLP tools are consistent
false starts in sentences are allowed	no false starts

Table 1.1 Differences between Text Conversation and Regular Document

Noisy unstructured text data is ubiquitous in real-world communication. Natural language and the creative ways that humans use it can create problems for computational techniques. Electronic text from the Internet (emails, message boards, chatlogs and web pages), contact centers (complaints, emails, call transcriptions), and mobile phones (SMS) is often noisy - contains spelling errors, abbreviations, non-standard words, missing punctuation, missing case information and special characters, some of the problems apart from those mentioned earlier.

The degree of distortion of structure and nature of the text in user-generated content varies

- From domain to domain

- From user to user
- Web technology collaboration environment
  - Presence of dictionaries can lead to a standard.
  - Additional space constraints may tend user to input more information represented by less text, thereby, resulting in increased noise.
- User contributes in different styles in same domain and environment depending on his mood, availability of time etc. can add sparsity to data in addition to noise.

These characteristics of social media conversation have raised new sets of challenges for the task of Information Retrieval and Knowledge Management.

Conversations from social media tend to suffer from the inherent problem of data sparseness. The presence of infrequent terms in the data makes it difficult to analyze conversations. Syntactic features do not give much insight and therefore association between the topics and understanding the intent of the conversation becomes a challenging problem.

1.5 Problem Statement

Automatic text summarization is an information access technique used to present only the most important information from a set of documents, thereby reducing the need to look into the actual documents. In our case, these documents are chat and email conversations belonging to the category of UGC. Please note that we use the term *document* very loosely in this thesis. It is used to refer to both written text email and chat conversations.

In this thesis we aim to reduce the problem of information overload and redundancy by automatically summarizing text conversations and providing an efficient summary to the end user. The goal is to take a conversation and generate a short and concise summary that can be read in lieu of the original conversation. In this thesis, we have mainly focused on development of extractive methods of summarization with sentences as the basic units of summary. Therefore, the problem can be transformed to "the selection of the most informative sentences from the piece of text which best represents the document". We approach this problem of sentence extraction in two different ways, feature extraction based machine learning approach and topic based language modeling approach. In the first approach, we represent each sentence as a collection of features. From these feature vectors we build a supervised machine learning model to rank sentences. Whereas the second approach focuses more on the topics that might be involved in the conversation text. It digs deep into semantic level representation of conversation in the form of topics, where sentences are scored based on their co-occurrence with the topic words. This is an unsupervised approach. Both the approaches are language independent and scalability has been taken into consideration while designing such systems.

1.6 Our Approaches

In this section, we discuss in brief, how we approach the novel problem of conversation summarization. Internet conversation presents challenges that necessitate applications of email and chat conversation summarization techniques. Summarization can be achieved either through *abstraction* or *extraction* of information from source text. While abstractive summarization could provide a more readable summary, state of the art systems are all extractive summarizers, where most representative sentences are picked from the text to form the summary. Extractive approaches of summarization can be attained through different levels of granularity like, words, sentences and paragraphs. Keyword extraction based summaries hardly provide any readability whereas paragraphs do not cover the entire information provided the space constraints. Therefore, we choose the midway solution of using sentences as the information units for the formation of summaries.

Using sentences as the building units for summarization allows us to break the problem of summarization to sentence selection. Our goal is to select those sentences into summary that are highly relevant to the conversation, therefore producing an informative and representative summary. We approach this problem by exploiting the statistical characteristics and semantics of the conversation. We follow a sequential framework, that include *Preprocessing* of text for word spelling correction and sentence boundary identification, a *Feature Extraction* stage where statistical, linguistic and heuristics models are employed to estimate sentence importance and finally a *Summarizer Training*, during which we learn a supervised machine learning model, to rank the sentences and generate a summary. We represent each sentence as a collection of features. Most of the existing features are frequency based and perform poorly on conversations because of the data sparsity in UGC. We devise new features to capture the discourse structure and opinions in the conversations. We use Rhetorical Structure Theory (RST) to explain the purpose of existence of text. In our work, we have utilized five relations from RST to define sentences present in the conversations. We also compute opinion scores for each sentence using Senti Wordnet. Strong opinions (positive or negative) often carry important information with respect to the conversation.

Our feature set is a collection of various frequency-based and heuristic-based measures along with the discourse and sentiment scores. We train our summarizer on this feature set using different machine learning algorithms to show that it is generic enough to learn from any of the algorithms. We use two datasets composed of email and chat conversations from technical domain to validate our approach. We observe that inclusion of new features leads to the generation of improved summaries. We conduct a series of experiments to show the robustness of our approach. A detailed description of our summarizer and the newly devised features is provided in Chapter 3.

Our machine learning approach towards summarization of text conversations is supervised in nature. Therefore, our method requires reference (model) summaries as input to learn the trainable summarizer. Such a task of summary creation requires human intervention. It can also affect the performance of summarizer because human summarization is a cognitive process and manual summaries can be biased. With the motivation of getting rid of manual effort we develop a new unsupervised topic-focused

approach. The conversations from social media contains multiple participants and a single conversation tends to span a wide range of topics interspersed with irrelevant segments, thus topic-focused summarization becomes essential. Here our summarization process is guided by the primary topic of discussion. The most prevalent topic of discussion in the conversation is called the primary topic. We detect the primary topic of discussion in the conversation and build a semantic word space using these topics. However, because of the sparseness in data identifying the connected components and detection of primary topic becomes a difficult task. To tackle this problem we have developed a two phase approach. In the first phase, we leverage topic modeling using web documents to find the primary topic of discussion in the chat. Then, in the summary generation phase, we use relevance-based language modeling approach to build a semantic word space. We use Hyperspace Analogue to Language (HAL) model for this purpose. It is a lexical co-occurrence model which computes the word dependencies based on the context in which they appear. Then the sentences are ranked using this word space based on their association with the primary topic. Experiments performed show that incorporation of topic information in HAL model is beneficial in conversation summarization. We also observe that exploiting topic modeling with web as a resource helps in finding the topic structure more accurately, thereby generating improved summaries. Results show that we have statistically significant performance gain over baselines with ROUGE as the evaluation metric. We have discussed our topic-focused approach towards summarization in detail in Chapter 4.

In our work, we have exploited the statistical characteristics and semantics of the conversation. We examine the role of discourse structure and opinions in conversations. Our approach differs from any of the existing work because we have built a generic framework which targets conversations rather than specifically emails or chats. We have chosen real world conversations from technical domain for the validation of our approach. We tackle the problem of data sparsity by building semantic models for primary topic detection and sentence ranking. The use of extra information provided in the form of web documents can be effectively mined using topic modeling to discover the hidden topic structure more accurately. This topic information guides the summarization process and generates improved summaries. We have conducted different experiments to thoroughly evaluate our approach. This is one of the most premier works in the area of conversations summarization.

1.7 Contributions

In this section we list the main contributions made in this thesis.

- We examine the need for summarization systems in written text conversations.
- We devise new robust features to capture statistical, linguistic and sentimental aspects along with the dialogue structure of the conversations. We examine the role of discourse structure (Rhetorical Structure Theory) and Sentiments in conversation summarization.

- We propose a generic feature set which can summarize both email and chat conversations using a wide range of machine learning algorithms.
- We investigate the importance of latent topics present in the conversations to generate better summaries. We exploit the semantics of the conversation in the form of topic structure.
- We apply topic modeling on web documents to find the topic distribution of sentences. In conversations, the relationship between entities and objects is not explicitly present, thus leading to data sparseness. We tackle this challenging problem by using web as a resource.
- We show how incorporation of topic into language models can provide better summaries. We declare a fine boundary between relevant and non-relevant topics.

1.8 Organization of Thesis

Chapter 2 gives background related to summarization along with the various flavors of summarization. We also discuss how conversation summarization differs from traditional document summarization. Text from social media poses new challenges by differing in its informational need compared to news articles as in case of traditional document summarization. We provide a detailed survey on relevant literature in the context of this thesis. It includes different state of the art approaches including machine learning approaches, centroid-based approaches, topic models in summarization along with their success and drawbacks. We have taken into perspective the social media while discussing past work. Evaluation criteria is being discussed in the later parts of this chapter.

In Chapter 3, we describe general summarization framework explaining various stages in a summarizer. We give a detailed description of the proposed features and show how to compute them automatically. These features are then used to build a trainable summarizer using different machine learning algorithms. It is followed by different set of experiments that we carried out to thoroughly evaluate our algorithm and to show the effectiveness of our approach.

Chapter 4 discusses the role of hidden topics in determining the sentence importance. We discuss an unsupervised approach of finding the hidden topics in conversations using web as a resource. We show how the incorporation of topic information in relevance language models can lead to the generation of better summaries. Later, in the chapter we conduct various experiments to evaluate and determine the significance of topics over generic summarizers. Evaluation results of the experiments are compared to the state of the art approaches in conversation summarization.

Chapter 5 concludes this thesis explaining the work done and discusses the results of our experiments. It discusses the possibilities of adoption of the central idea presented in this thesis to different domains in social media. It also provides details about foreseeable future work with respect to the thesis.

Chapter 2

Summarization Background and Related Work

It is hard to imagine everyday life without some form of summarization. A trailer or preview of a movie is a summary. Abstracts of research papers and scientific articles are summary written by authors. Other examples are minutes of a meeting, a resume, a program for conference, reviews of a product in e-commerce sites, can all be considered as summaries in their respective domains. Such information provided in the form of summaries is easy to read and coherent, therefore can be understood with less effort as compared to raw information. However the creation of these summaries requires human intervention and ample amount of time, along with resources in some cases. With ever growing content on World Wide Web, it is not possible to manually create summaries on such a large scale. These manually created summaries tend to suffer from human biasness as everyone has different perceptions about a topic. Automatic text summarization systems address these problems of scalability and biasness.

2.1 Automatic Text Summarization

Text Summarization is the process of condensing text to its most essential points. When summarization process is carried out by machines, it is termed as Automatic Text Summarization. Although the definition of summarization is obvious, it needs to be emphasized that summarization is a hard problem. A summarization system has to interpret the source content, where content is a depiction of both information and emotion, and identify most relevant information. Classification of information as relevant or irrelevant is subjective to human nature. Summarization is a challenging task for its inherent cognitive process, as an ideal summarization system has to mimic a human mind in the process of abstracting. Summarization is also interesting for its practical and real life applications. Researchers [26] have listed down summarization as a three phase process, consisting of,

- Topic Identification: An initial exploration of text to identify its genre and topic.
- Relevance Assessment: Important and relevant topics are aggregated together, and presented in a new form. This formulation represents the actual concepts which may not be explicitly present in the text.

- Summary Generation: New formulation after relevance assessment is transformed into a coherent human readable format. It often involves cutting and pasting certain portions of text to form a summary.

Each phase may involve several sub processes depending on the context and informational need.

2.2 Flavors of Summarization

Summaries can be classified into many different categories based on their input factors like media, genre and depending on the context within which summary is intended to use. Below we discuss some major categories:

2.2.1 Extractive vs Abstractive

An extractive summary is a summary which is solely constructed by extracting the important passages, sentences or phrases from the source text. In contrast, an abstractive summary may or may not contain words in common with the source text. Although there have been some efforts in creating automatic abstractive summarizers, extraction is still the most feasible approach and dominates the area of summarization. In general, abstractive summaries are written by humans.

2.2.2 Single Document vs Multi-Document

This categorization is based on the original content considered for summarization. In single document summarization, one needs to summarize only single text document. The trend of text summarization has moved from single document to a more challenging problem of multi-document summarization. Multi-document summarization involves generating summary for a set of multiple related documents, where the document set is likely to contain similar content. It is important to note that concatenation of individual single document summaries may not necessarily produce a multi-document summary.

2.2.3 Query-Focused vs Generic

A generic summarizer produces summaries by capturing all the important information from the source text. On the contrary, query-focused summarizer produces summary by taking into context the user information need. Summarizer has user information need in the form of a query, and retrieves the query relevant sentences/passages from the source text. With the immense growth in online search and retrieval, the query focused summarizers would provide a better summary than generic summarizers.

2.2.4 Personalized Summarization

Relevance assessment is a difficult task which changes from person to person. A personalized summarizer caters based on users interest and personal background. The personalized summarization aims to adapt summarization result of a specified document to an intended user based on the users interest which are inferred from social context.

2.2.5 Guided Summarization

In guided summarization, the summary is guided by a fixed template pertaining to a set of questions that is prepared for that particular domain of text. The template is in the form of a questionnaire and the summary should answer the questions in the template using source text. It leads to the production of very focused summaries with respect to the raised questions.

2.3 State of the Art Approaches in Summarization

The introduction of summarization track at conferences like Text Analysis Conference (TAC)[1] and Document Understanding Conference (DUC)[2] has allowed researchers to compare their results which has led to an enormous increase in the number of approaches. These techniques can be categorized under heuristic, discourse based, machine learning, topic modeling approaches among several others. Since it is not feasible to list out all the wide range of approaches in summarization, we provide below only the state of the art approaches and relevant work done in the context of social media, particularly conversations. Most of the work stated below is focused towards summarization of news articles and comparatively less work has been done in conversations from social media.

2.3.1 Linguistic Structure or Discourse Based Approaches

Barzilay and Elbadad [3] describe the notion of cohesion by developing a concept of lexical chains for performing the task of summarization. Cohesion means sticking together different parts of the text. They have achieved cohesion by performing considerable amount of linguistic analysis, which involves co-reference, ellipsis and conjunctions. Cohesion occurs not only at word level but at word sequences too which results in the formation of lexical chains. They made use of lexical chains, sequences of related words in a text spanning short or long distances, to identify important information. After lexical chain identification, the sentences having strong lexical chains are selected for extraction to be a part of the summary. Wordnet [38] distance is used as a relatedness measure to find out lexical chains.

Among the different cohesion building devices "lexical cohesion" is the most easily identifiable and most frequent type, and it can be a very important source for the flow of informative content. However,

[1] `http://www.nist.gov/tac/`
[2] `http://duc.nist.gov/`

this model has a significant disadvantage as it uses some knowledge source like Wordnet, dictionaries etc. Conversations belonging to the category of user-generated content cannot be represented by any static dictionaries or knowledge sources like Wordnet and Wikipedia. The rule based approach followed to identify the candidate lexical chains in monologues is not applicable to social media where the content does not follow any pattern and contains high level of distortion.

Marcu [36] developed a discourse-based approach. He used the structural aspect of the text to discover the relationship between the segments of the text. The relationships were defined by using Rhetorical Structure Theory (RST) [35]. He created an RST tree for the document to be summarized with nodes of the tree representing segments from the text. The tree is constructed in a manner such that segments which are more important occupy the upper level of the tree, whereas less important segments reside deeper in the tree. Summarization is done by performing cuts at different depths of the tree. He showed that discourse trees are good indicators of textual importance. He devised a discourse parsing algorithm to indicate discourse connectives that describe importance.

2.3.2 Centroid and Cluster Based Approaches

Radev et al. [43] exploited the use of cluster centroids to summarize documents. Documents in the form of news articles are modeled as bags-of-words. The first stage consists of topic detection, where the documents that describe same event are grouped together using agglomerative clustering algorithm. It operates over the TF-IDF vector representations of the documents, successively adding the documents to the cluster and recomputing the centroids. Later, the centroid of these clusters are used to identify sentences that are central to the topic of the cluster.

To calculate the sentence importance, two metrics cluster based relative utility (CBRU) and cross sentence information subsumption (CSIS) are introduced. Three sentence level features centroid value, positional value and first sentence overlap are used to approximate these metrics. The final score of each sentence is a combination of the three scores above minus a redundancy penalty for each sentence that overlaps highly ranked sentences. Once the documents are clustered, sentence selection from within the cluster to form its summary is local to the documents in the cluster. The IDF value based on the corpus statistics seems counter-intuitive. Both the position factor and the first-sentence similarity factor heavily weight the first few sentences of the documents in the cluster. Thus this metric is genre-specific and applies primarily to newswire articles. The approach is well known as MEAD [42], and open sourced for research purposes.

MEAD is designed for news articles and the performance degrades with social media conversations as the summarizing text. The use of TF-IDF and other features for calculating sentence importance is highly questionable in case of conversations. The conversations tend to have very less lexical overlap between text and therefore frequency based measures cannot truly represent the conversation text.

Zhou and Hovy [57] have worked on summarizing IRC logs. They cluster the discussions and then generate summaries for each of the clusters by extracting adjacent pairs of response by users. Here multiple topics have been considered to be part of the summary. They have used hierarchical text tiling

for text segmentation. Every cluster contributes to some proportion of the conversation, therefore one cluster can have less significance compared to others with respect to conversations. However, they have not taken into account the relative importance of different clusters in a thread. This is the most significant work in the area of conversation summarization.

2.3.3 Machine Learning Approaches

Recent advances in the field of machine learning have been adapted to summarization through the literature to identify important sentences. Usually, the problem of sentence selection is modeled as a classification or regression problem. Edmundson [16] and Luhn [33] have used features such as average term frequency, title words and cue phrases for scoring sentences to create a summary. They used very simple features which can be applied easily and have been considered as the baseline in the area of text summarization, but their approach does not perform comparatively well on social media. Frequency-based features are not a good representation for conversations where the data is sparse in nature. They have also made use of natural language techniques for detection of entities and objects which perform inconsistently on user-generated text.

A significant amount of work has been done in social media, but the majority of it focuses on email summarization. Summarization has been looked as a noun phrase extraction problem by Muresan et al. [40], where noun phrases are taken as the information units, and machine learning techniques are used to learn rules for extraction. Instead of using linguistic information, Rambow et al. [44] have used features based on the representational structure of the emails to perform summarization. Their results show that using email specific features benefit summarization performance significantly. However, the work is very specific to emails and same features are not applicable to other types of text media.

Carenini et al. [9] have also considered email summarization as a sentence classification problem but differs from Rambow et al. [44] as they have created an unsupervised system. They have shown that the clue words, subjective words and phrases can be used as an evidence to the importance of the sentences. Ulrich et al. [50] have done regression-based classification compared to the previous work where binary classifier have been used for summarization. Instead of doing binary classification where each sentence either belongs to summary or not, Ulrich et al. [50] have considered each sentence to be a part of summary with some probability. They have also shown that usage of speech acts and subjectivity of the sentence as features leads to better performance. However, they have used a manually annotated email dataset, and automating such speech acts to extend it to real world chat and email conversations is a challenging task.

2.3.4 Topic Models for Social Media

In an attempt to go a step further from statistical approaches, topic models have been extensively studied in automatic text summarization. Topic models provide a simple way to analyze large volumes of unlabeled text. These are a suite of algorithms that uncover the hidden thematic structure

in document collections. Arora et al. [1] have used Latent Dirichlet Allocation (LDA) to capture the events being covered by the documents and form the summary with sentences representing these different events. They used the mixture-models to explicitly represent the documents as topics or events. They have used hierarchical LDA to estimate the number of topics. Wang et al. [53] proposed a new Bayesian sentence-based topic model for summarization by making use of both the term-document and term-sentence associations. They have derived an efficient variational Bayesian algorithm for model parameter estimation.

Tang et al. [49] have used LDA for query-focused multi-document summarization. They build probabilistic models for each of the subtopics on sentence level. The model distribution for each sentence shows how possible this sentence is generated from each of the models. Chen et al. [11] have worked on lines similar to Tang et al. [49] with additional advantage that the proposed topic model based on LDA can simultaneously model both the documents and the query. They have showed that improvements over retrieval using cluster-based models can be obtained with reasonable efficiency.

In [21], a hierarchical LDA-style model was utilized to represent content specificity as a hierarchy of topic vocabulary distributions. Chang and Chien [10] proposed the sentence-based Latent Dirichlet Allocation (SLDA), which is accordingly established for document summarization. Different from the vector space summarization, SLDA is built to fit the fine structure of text documents, and is specifically designed for sentence selection. SLDA acts as a sentence mixture model with a mixture of Dirichlet themes, which are used to generate the latent topics in observed words.

An approach to building topic models based on a formal generative model of documents, Latent Dirichlet Allocation (LDA), is heavily cited in the machine learning literature, but its feasibility and effectiveness in information retrieval is mostly unknown. In our research, we study how to efficiently use LDA to improve ad-hoc retrieval. We leverage LDA using web as a resource to discover hidden topic structure of corresponding conversations more accurately. Then we use this topic structure information to guide the summarization process, which leads to the generation of improved summaries.

2.4 Conversation summarization: A Deviation from Document Summarization

Conversations are interactive and spontaneous form of communication between two or more participants who follow some social convention. The afore mentioned properties of conversations are justified as:

- Conversations are interactive because contributions to a conversation in the form of replies are response to what has already been said.
- Conversations are spontaneous and unpredictable in nature. In conversations people express their thoughts, ideas and expressions instantly.

- Conversations are social interactions which depend on social convention. It follow rules of etiquette, discarding which can dissolve and lead to termination of conversation.

Generating summary for such conversations is called conversation summarization. Summarization of conversations is perceived in a different way because of the medium and property of source content. Conversation summarization is in its beginning stages and is being studied extensively in the recent years. Conversations are more unstructured compared to news articles, blogs in content which makes it hard to extract conversation specific features which do not change across different domains and genre.

Conversation summarization differs from the traditional document summarization in its informational need and structure. Written conversations in the form of emails and chats have features like acronyms, hyperlinks, nicknames and spelling mistakes [14] which make traditional Natural Language Processing (NLP) techniques difficult to apply. Text in the form of news articles and books is a monologue whereas conversations fall in the genre of correspondence and requires dialogue analysis [39]. Since dialogue can involve several participants and be less coherent, less fluent, and more fragmented than monologue, conversations pose new challenges. Summarization of dialogue data is difficult because real-time conversations comprise a sequence of exchanges between multiple users that may be synchronous or asynchronous, and may span different modalities [46]. Both the structure of the conversation and the data available can vary with the modality. The level of distortion in conversations vary from domain to domain and from user to user depending on the rules of etiquette.

We would like to mention that proposing a generalized framework for both chat and email conversations is a difficult task. This is because even within the context of conversations there is notable difference between chats and emails. Chat conversations are more unstructured and noisy compared to emails. In case of chat conversations there is a relatively lesser co-occurrence of related terms. The replies are more specific with very less lexical overlap and do not repeat the same thing again unlike formal emails, where we reiterate on the subject of discussion. This implies that chat conversations are more sparse in nature and thus majority of the work has been done on emails compared to chat conversations. Rambow et al. [44] have used features based on the representational structure of the emails to perform summarization. Their results show that using email specific features benefit summarization performance significantly. However, the work is very specific to emails and same features are not applicable to other types of text media. Their features and approach is domain specific and guided by heuristics. Ulrich et al. [50] have proposed a more generic approach where they have used speech acts and subjectivity of text to guide summarization process. However, they have used manually annotated corpus for their purpose. Automating such speech acts and extending it to real world conversations in itself is a challenging task. In our work, we provide a generalized and unified framework to summarize both chat and email conversations. We have exploited statistical, semantic and dialogue structure of the conversations to build a robust summarizer.

2.5 Evaluation for Automatic Text Summarization

Evaluating the quality of a summary has proven to be a challenging problem, principally because there is no ideal summary as much. Studies [25] in past have shown that, human summarizers tend to agree only 60% (approximately) of the times with other's judgement. Apart from the human biasness involved in the evaluation of summaries, such a manual evaluation is also expensive and time consuming in nature. There is always a possibility of system generating a better summary that is different from reference human summary used as an approximation to the ideal output summary.

Lot of interest and activity has aimed at the development of multipurpose information systems since early 2000s. Several Government organizations like Defense Advanced Research Projects Agency (DARPA) and National Institute of Standards and Technology (NIST) have made continuous efforts through conferences like DUC and TAC, along with a belt of other conferences, topic sessions and workshops on the promotion of automatic text summarization and evaluation.

Based on various statistical metrics [31], results show that automatic evaluation using n-grams or unigram co-occurrences between summary pairs correlates well with the human evaluations. Finally, ROUGE, a package for automatic summary evaluation, was developed by Chin Yew Lin [30]. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is a recall oriented metric to automatically determine the quality of the summary. ROUGE package is an openly available resource that has four major measures: ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-S. Below we provide a brief description of ROUGE-N and ROUGE-L measures, which we have used for automatic evaluation of summarization systems.

- **ROUGE-N** is an n-gram recall between the candidate summary and the model summary. The correlation between candidate summary and model summary can be established as a measure of count of overlap of segments between both the summaries. ROUGE-N is computed as

$$ROUGE - N = \frac{\sum_{gram_n \in m} COUNT_{match}(gram_n)}{\sum_{gram_n \in m} COUNT(gram_n)} \tag{2.1}$$

  where n is the length of the n-gram, $COUNT_{match}(gram_n)$ denotes the maximum number of overlapping n-grams between candidate summary and the model summary m. If more than one model summary is provided, then ROUGE-N is

$$ROUGE - N = \frac{\sum_{m \in model} \sum_{gram_n \in m} COUNT_{match}(gram_n)}{\sum_{m \in model} \sum_{gram_n \in m} COUNT(gram_n)} \tag{2.2}$$

  ROUGE-N is a generalized version for n-grams. It can be used to obtain ROUGE-1 and ROUGE-2 by substituting n-grams with unigrams and bigrams respectively. ROUGE-1 scores count all co-occurring words regardless of their orders whereas ROUGE-2 is stricter and takes into consideration the immediate context. ROUGE-2 scores tend to be lower than ROUGE-1.

- **ROUGE-L** is the Longest Common Subsequence (LCS) based recall to estimate the similarity between candidate summary and the model summary. The motivation is that the longer LCS between two text implies more cognate the candidate summary is to the model summary.

  To apply LCS on summary, we take the union LCS matches between a model summary sentence, m_i, and every candidate summary sentence, c_j. Given a model summary of u sentences containing a total of x words and a candidate summary, C, of v sentences containing a total of y word, then summary-level LCS based ROUGE measure is

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(m_i, C)}{x} \tag{2.3}$$

$$P_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(m_i, C)}{y} \tag{2.4}$$

  R_{lcs} and P_{lcs} are Recall and Precision scores. Every sentence of the model summary is considered for LCS match against the candidate summary and the scores are accumulated. These accumulated scores by then normalized with the total words. The difference in the denominator reflects the nature of both the measures. Recall will be high when more sentences matches the model summary, whereas Precision will be favoured when the content of the candidate summary matches as much as possible even if it short in length comparatively.

$$ROUGE_L = F_{lcs} = \frac{2R_{lcs}P_{lcs}}{R_{lcs} + P_{lcs}} \tag{2.5}$$

  F scores tries to balance Recall and Precision values. LCS is a good measure because it does not require consecutive matches but in-sequence matches, which leads to independence in word order compared to n-grams. ROUGE-L also captures sentence level structure effectively. However, ROUGE-L has one main disadvantage that it only counts the main in-sequence words; therefore, the other alternative sequences and shorter sentences are not reflected in the final score.

In this research, we have used ROUGE-1 (unigram), ROUGE-2 (bigram) and ROUGE-L (longest subsequence) scores for evaluation purposes.

Chapter 3

Supervised Sentence Ranking for Conversation Summarization

Summarization in its basic essence is to extract the essential information from a collection of textual content and present it in a readable format. Produced summary is called as the representation of the document. Summarization is a multidisciplinary problem with roots in Information Retrieval, Natural Language Processing, Data Mining and Cognitive Science. Over the years, research in summarization has led to some interesting sub-problems like personalized summarization, multi-document summarization, blog summarization among others.

Conversation summarization is a relatively new area within summarization. It is designed to aid users having access to rapidly flowing stream of conversations from UGC on a topic and has no time to look at every conversation. A short precise summary in such situations can help user by providing the essence of the conversation. In this chapter, we will discuss a sentence extraction machine learning framework. We design robust features and apply different machine learning methods to produce a final informative summary.

3.1 Summarization Framework

Summarization can be viewed from different perspectives, as a decision theory problem, as a classification problem of summary and non summary sentences, as a regression problem of estimating the importance of a sentence, or as an Information Retrieval problem of extracting relevant sentences. Here we use a general model which allows different views to be implemented as individual features of the summarization framework. Then different machine learning methods are used to combine all these distinct features to produce a final summary.

3.1.1 Stages of Framework

Our summarization framework constitutes of different stages. We have separated each module according to its functionality and dependence on other components of the system. The flexible nature

of our framework allows us to implement arbitrary algorithms in a standardized framework. We will discuss the different stages of our framework below,

- **Preprocessing**

  We have chosen conversations in the form of IRC logs and emails from technical domain belonging to the category of UGC. Data collected from the web or any publicly available corpus has unnecessary headlines, HTML tags that does not provide much information about the conversation. Each such conversation is parsed to extract the main content. Since the data belongs to UGC, it is very unstructured and noisy in nature. Spelling mistakes and lack of sentence boundaries are some of the problems that tend to occur in these text documents. We have two modules which add to Preprocessing stage, namely,

  - **Spell Corrector:**

    This module is built for spelling correction of words. Social media content often contains words with missing vowels, repetition of alphabets to lay emphasis, incorrect punctuations to name a few. To get rid of spelling mistakes, we use web as a resource, followed by a distance metric to choose the closest match. Spell corrector operates at word level.

  - **Text Segmenter:**

    We are dealing with summarization as a sentence extraction problem. Each sentence is a content unit holding information. To process a sentence, which is a unit of information, we need to have text with sentence boundaries. But real-time conversations are very instantaneous, informal in nature with no sentence boundaries. In this regard, we propose a new approach for text segmentation. Text Segmenter operates at sentence level.

  We have discussed in detail the Spell Corrector and Text Segmenter module in section 3.2.

- **Feature Extraction**

  Sentences extracted during preprocessing stage are considered as units of information. Each sentence is represented by a set of scoring features, reflecting its relevance on a positive or negative scale. These features may belong to probabilistic language models, entropy based measures, heuristics derived from corpus, linguistic and knowledge based measures among many others. Usually more than one feature is used in scoring to attain robustness.

  We employ different types of feature to capture the statistical, linguistic and sentimental aspects along with the dialogue structure of the conversations. Some of the features have been utilized in past by researchers to solve varying problems in information retrieval, including sentiment analysis, search engine retrieval and tag generation to name a few. We have also designed new features like Discourse Marker, Sentiment Score to capture the information present in the sent. We describe all these features representing a sentence unit in Section 3.3.

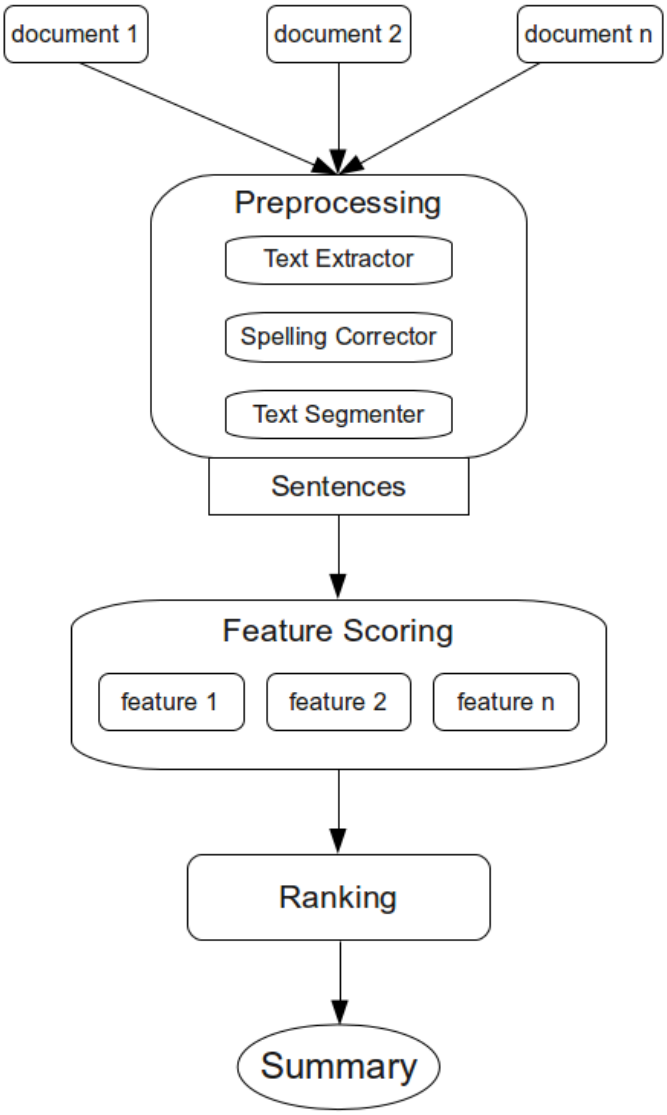

Figure 3.1 System Architecture for Machine Learning Summarization Framework

- **Summarizer Training**

  Rank of a sentence is directly proportional to the importance of a sentence and decides its membership in the summary. Conventionally sentence rank is computed as the weighted linear combination of feature scores. But finding the optimal set of weights for a set of features becomes difficult with increase in feature space. Therefore, to overcome these problems we use various machine learning technique to train our summarizer.

  After feature extraction process, each sentence can be represented as a feature vector. These set of feature vectors are then given as input to learn the trainable summarizer using different machine learning algorithms. Features are pluggable components of the framework, hence every combination of features becomes a unique configuration of summarizer.

- **Summary Generation**

  Summary Generation is the final stage of summarization, where a subset of ranked sentences are selected into the summary till they reach the desired length. Usually compression ratio (CR) between .15 to .25 is used to decide on the length of the summaries. CR is given by,

$$CR = \frac{length\ of\ Summary}{length\ of\ Full\ Text}$$

  Accordingly top k sentences are then picked to form the summary. Picked Sentences are adjusted based on their order of occurrence in the conversation to improve the readability of summary.

This flexible framework allows us to plug new features with ease. It also helps us in investigating the proposed features more thoroughly.

3.1.2 System Architecture

To implement the summarization framework discussed in the previous section, we design a three phase architecture which deals with the text at different level of granularity. We have considered real world conversations comprising of emails and chats that belong to the category of UGC. These conversations are very noisy and unstructured in nature. To make use of such data, we need to preprocess and normalize the text. In the first phase of our architecture, we prepare the dataset for usage by extracting text, correcting spellings and then segmenting the text. In the second phase, we represent each sentence by a set of predefined features. These features are based on the characteristics of the conversations including statistics, heuristics, sentiments and dialogue structure of the text. Finally, in the third phase we use a machine learning algorithm to train the summarizer on a set of feature vectors and generate the summary using top ranked sentences. Figure 3.1 shows our three phase architecture for summarization.

3.2 Data Normalization

Before doing any kind of text processing, we need to reduce the data into its canonical form. Such a process is called data normalization.

3.2.1 Spell Correction

To get rid of spelling mistakes we use web as a resource. After removing stopwords from the text, we query the web using Microsoft Bing Search API[1]. For each word we construct a trigram from the text by considering its adjacent left and right context words, and query the web with this trigram. The web results that we get contain related words. Then using Levenshtein distance[2] we pick the three closest candidate words from which we select the one with the highest frequency. We summarize our algorithm for Spelling Correction as:

Algorithm 1 Spelling Correction

for each word w in the chat **do**
  Initialize priorityQueue Q = { }
  bingResponse := QueryBing (w_0 w w_1)
  snippetSet := extractSnippets (bingResponse)
  for each word w' in the snippetSet **do**
    $Q.push$(Levenshtein's Distance(w, w'))
  end for
  Initialize closestWordSet CWS := Top3Words(Q)
  Initialize $closestWord$:= MaxFreqWord(CWS)
  if w **is not** $closestWord$ **then**
    w = $closestWord$
  end if
end for

where w_0 and w_1 are the left and right adjacent words of the context word w.

3.2.2 Text Segmentation

Since online conversations are not formal in nature, the arguments and points of discussions are not punctuated appropriately which leads to the need for text segmentation. We did text segmentation using two approaches, TextTiling and Longest Contiguous Messages (LCM). In the first approach we used TextTiling [22] algorithm which segments the conversation based on the topic drifts. To assign sentences on per user basis we applied TextTiling on the obtained segments in a hierarchical manner. Whereas in our novel approach of LCM, we concatenate the subsequent messages of the same user, and when the user changes or when a period is encountered we consider it as a sentence boundary. We

[1] http://msdn.microsoft.com/en-us/library/dd251072.aspx
[2] http://en.wikipedia.org/wiki/Levenshtein_distance

have compared both these approaches in Section 3.6.2 and found out that LCM outperforms TextTiling. Table 3.1 shows the application of LCM on a chat conversation excerpt.

Chat Conversation Excerpt	On Applying LCM
jbailey: I've heard that recent kde's are quite nice and fix the usability problems I was having. Hey, I'm setup no gnome, so why try other things?	jbailey: I've heard that recent kde's are quite nice and fix the usability problems I was having. jbailey: Hey, I'm setup on gnome, so why try other things?
chillywilly: cause kde performs better, imho and chillywilly: that's a big reason why I switched jbailey: *shrug* I've had no problems running gnome on a 500mhz celeron.	chillywilly: cause kde performs better, imho and that's a big reason why I switched. jbailey: *shrug* I've had no problems running gnome on a 500mhz celeron.

Table 3.1 Effect of using LCM on a Chat Conversation Excerpt

3.3 Extraction of Sentence Relevancy Features

In extractive text summarization, our main objective is to select the most representative and relevant sentences from the text. To deal with the problem of sentence extraction, we represent each sentence by a feature vector. Each feature vector contains a set of different features which capture the basic information content and the dialogue structure of the conversation. We classify the features into two sets: **basic** and **complete**. The **basic** feature set contains features which are not specific to conversations and consider the conversation as a simple piece of text. These features incorporate the statistical and linguistic aspects of the sentence. Features present in the **basic** feature set are:

3.3.1 Basic Feature Set

- **Mean TF-IDF**

  Frequency has been used as a feature for various text processing tasks. We use the tf-idf [45] scheme to characterize the frequency of a word. The value of this feature is calculated as the mean of the tf-idf values of all the words present in the sentence. This value is normalized by dividing it with the largest corresponding value among all the sentences in the conversation. Let $f(t, d)$ be the raw frequency of term t in document d, then the normalized term frequency, $tf(t, d)$ is given by:

$$tf(t, d) = \frac{f(t, d)}{max\{f(w, d) : w \in d\}} \tag{3.1}$$

  If $DF(t, D)$ represents the document frequency for term t in document collection D, then the inverse document frequency, $idf(t, D)$, is given by:

$$idf(t, D) = log\frac{|D|}{1 + DF(t, D)} \tag{3.2}$$

$$tf - idf(t, d, D) = tf(t, d) \times idf(t, D) \tag{3.3}$$

- **Mean TF-ISF**

  This feature also captures the frequency but takes into consideration only one conversation at a time. The frequency of a word is characterized as term frequency $\times$ inverse sentence frequency (tf-isf) [28]. This feature is calculated as the mean value of the tf-isf of all the words present in the sentence. This value is normalized by dividing it with the largest corresponding value among all the sentences. It is different from the mean TF-IDF feature as here we try to capture the importance of words based on their frequency among sentences, whereas in mean TF-IDF we capture the importance of words based on their frequency among documents. If $f_s(t, s)$ represents the frequency of term t in sentence s, then the normalized term frequency $tf_s(t, s)$ is given by:

$$tf_s(t, s) = \frac{f_s(t, s)}{max\{f_s(w, s) : w \in s\}} \tag{3.4}$$

  If $SF(t, d)$ represents the sentence frequency of a term t in document d, and S_d, the number of sentences in document d, then the inverse sentence frequency, $isf(t, d)$, is given by:

$$isf(t, d) = log\frac{S_d}{1 + SF(t, d)} \tag{3.5}$$

$$tf - isf(t, s, d) = tf_s(t, s) \times isf(t, d) \tag{3.6}$$

- **Sentence Length**

  This feature is included to avoid shorter sentences which can be incomplete and have less probability of contributing to the main summary [27]. Longer sentences tend to contain more information. We use the normalized length of the sentence as the feature. It is the ratio of the number of words in the sentence to the number of words in the longest sentence of the conversation. Sentence Length Score of a sentence s in a document d, $SLS(s, d)$, is given by

$$SLS(s, d) = \frac{n_s}{max\{n_i : i \in d\}} \tag{3.7}$$

  where n_s denotes the number of words in sentence s.

- **Sentence Position**

  Sentence position plays a major role in determining the presence of the sentence in the conversation summary [28]. There can be several ways of defining sentence position: it can be position

of the sentence in the user utterance, section or absolute position in the complete conversation. We consider the absolute position of the sentence in the text as the feature. It is normalized by dividing with the number of sentences present in the conversation. For a sentence s in document d, sentence position score, $SPS(s, d)$ is given by,

$$SPS(s, d) = \frac{n}{N} \tag{3.8}$$

where n denotes the position of the sentence s, and N, the total number of sentences.

- **Similarity to Title**

  The title is a short and precise representation of the primary topic in the conversation. We represent the title and the sentences in vectorial form using tf-isf scheme as given by equation 3.6. This feature is calculated as the cosine similarity [45] between the title and the sentence vector. If $\overrightarrow{s}_{tf-isf}$ and $\overrightarrow{t}_{tf-isf}$ represent the tf-isf vector of a sentence s and title t of the document respectively, then the similarity to title feature score, $STS(s, t)$, for sentence s is computed as,

$$STS(s, t) = \frac{\overrightarrow{s}_{tf-isf} \cdot \overrightarrow{t}_{tf-isf}}{|\overrightarrow{s}_{tf-isf}| \times |\overrightarrow{t}_{tf-isf}|} \tag{3.9}$$

- **Centroid Coherence**

  All sentences are represented in vectorial form using the tf-isf scheme(Equation 3.6). The average of all the sentence vectors represent the centroid. This feature is defined as the cosine similarity between the sentence vector and the centroid vector. Sentences with feature values closer to 1 can be considered to better represent the basic ideas of the document [27]. For a document d with N sentences, centroid vector, C_d, can be calculated as,

$$\overrightarrow{C_d} = \frac{\sum_{s \in d} \overrightarrow{s}_{tf-isf}}{N} \tag{3.10}$$

  For a sentence s from document d, centroid coherence score, $CHS(s, d)$, is given by,

$$CHS(s, d) = \frac{\overrightarrow{C_d} \cdot \overrightarrow{s}_{tf-isf}}{|\overrightarrow{C_d}| \times |\overrightarrow{s}_{tf-isf}|} \tag{3.11}$$

- **Special Terms**

  Online conversations like technical IRCs, forums are often flooded with numbers and proper nouns which represent the important facts [4]. We consider the numbers and proper nouns as special terms. This feature is the count of the number of special terms present in the sentence. We normalize this feature by dividing it with the total count of unique special terms present in the conversation. We use Stanford Parser[3] for this purpose.

[3] `http://nlp.stanford.edu/software/lex-parser.shtml`

3.3.2 Complete Feature Set

In addition to the **basic** feature set, we consider additional features which capture the dialogue structure and the sentiment information of the conversations. We call this *larger* feature set as **complete** feature set. New features introduced are:

- **Is Question**

  The technical forums, IRCs frequently deal with resolving issues and addressing concerns raised by the users [44]. Usually, issues and concerns are raised as questions. This feature is represented by two values, 0 or 1, where 1 represents that the sentence is a question, otherwise not.

- **Sentiment Score**

  This feature captures the sentiment present in the sentence. Opinions with strong sentiment carry lot of information and have more chance of contributing to the summary [17]. Unresolved issues are often marked by negative response whereas user satisfaction is generally denoted by positive response. To get the sentiment score of a word we use SentiWordNet[4], where each word is given a positive and a negative score between [0, 1]. Sentiment score for the sentence is obtained by accumulating the scores for all words present in the sentence. We have normalized the scores for this feature by dividing the sentence score by the number of words present in the sentence. Here we focus on correctly detecting the presence of sentiment rather than classifying the sentiment present as positive or negative. Instead of dealing with all words, we have restricted ourselves to verbs, adjectives and adverbs because majority of the left out words will have positive and negative score of 0, and objectivity of 1. We calculate the sentiment score of a word w, $SS(w)$, as:

  $$SS(w) = positiveScore(w) - negativeScore(w) \tag{3.12}$$

  If N represents the number of words in a sentence s, then sentiment feature score for s, $SFS(s)$, is given by,

  $$SFS(s) = \frac{\sum_{w \in s} SS(w)}{N} \tag{3.13}$$

- **Discourse Marker**

  This feature defines the purpose of existence of a sentence in text. We have used Rhetorical Structure Theory (RST) to define this feature. RST has been used in a variety of ways in different domains of information retrieval, including automated generation of text, as a prompting for the development of linguistic theory, as a guide to text analyzers for summarization, teaching writing

[4] http://sentiwordnet.isti.cnr.it/

skills and as an analysis framework for a wide variety of kinds of text. According to RST, each sentence in a discourse has some meaning with respect to the complete text. RST explains the coherence of the text by assigning relationships between various units (sentences in our case) of the text. RST raises issues about communication, semantics, and especially the nature of coherence of texts.

RST makes use of two kinds of building blocks to describe text, namely, *Nuclearity* and *Relations*. Mann and Thompson [35] showed that a sentence can be decomposed into segments, usually clauses, and then the relationship between these segments is established using the concept of Nuclearity. If a sentence has two clauses, then the main segment is called the nucleus, and the subordinate segment is called as satellite. Nuclearity helps in providing heirarchical structure to the text. Chuang and Yang [12] incorporated this information as a feature for summarization, considering the nucleus as more important than satellite. However, we do not consider the nuclearity aspect of RST. In our case the text belongs to technical domain which has different information need. For example, consider a sentence like *"We want to use XYZ software, but it is not getting installed using ABC command"* which has the first part of the sentence as nucleus followed by a satellite. Here both segments are equally important with respect to the summary. Therefore, we have chosen sentences as the basic units for segmentation instead of clauses.

Relation	Definition	Example
Greet	Greetings and pleasantries.	Hi Sir, Respected Sir/Maam, Thanking you, Bests
Background	Setting context for the discussion. Mainly informative.	My Ubuntu's version is 12.04 on 32 bit computer.
Query	Issue being discussed or a request. Main topic of discussion.	I was trying to install a new editor, but i am not able to do it.... can you suggest anything ?
Elaboration	Content and idea behind query, discussed by the participant who issues the query.	When i used apt-get command, it threw some error. The log is very big, and the terminal closes abruptly.
Solution	Replies from the participant regarding the problem. Authoritative, affirmative and strong opinions.	You need to use sudo to install it. This may be because of false interpretation.

Table 3.2 Brief Description of RST Relations

To describe our data, which are technical chats and emails, we have used only the Relation aspect of RST. There are many relations proposed in RST to define the text, however we use only five relations to define our data. We selected only 5 relations to define the text because when we clustered the sentences of a set of documents where each cluster represented a relation, most of the clusters were empty. Thus, we selected *Greet*, *Background*, *Query*, *Elaboration* and *Solution*

relations that contained sentences. These relations are signaled by using cue phrases like *because*, *if*, *since* etc. Using Marcu's discourse-marker-based hypothesizing algorithm [36], we discover these relations at the level of sentences. Since the same sentence can be marked with two different relations, we consider the following preference order:

$$Query > Solution > Elaboration > Background > Greet$$

Each sentence is marked by one of these five relations as Greet, Background, Query, Elaboration or Solution. These relations define the way text is structured in email and chat conversations. Table 3.2 explains the meaning of these relations in the context of conversations.

3.4 Features Combination Using Machine Learning Algorithms

After selecting the features for summarization, each sentence is converted into a feature vector. These set of feature vectors are then given as input to learn the trainable summarizer using different machine learning algorithms. Various machine learning algorithms have been proposed in the literature. We chose the Decision Tree, the Naive Bayes classifier, Multilayer Perceptron and Support Vector Machines for our experiments.

3.4.1 Decision Trees (C4.5)

Decision tree learning is one of the most widely used method of inductive learning. From the family of decision tree algorithms, we chose C4.5 algorithm [41] to train our summarizer. C4.5 has known to be a very fast and efficient algorithm with good generalization capability. A decision tree is created by representing each node as a feature. Each feature is associated with a set of rules which are learned by the algorithm. While testing each sentence is represented by a feature vector and compared to the nodes of the decision tree. We used J48 which is an open source Java implementation of the C4.5 algorithm in Weka[5] toolkit.

3.4.2 Naive Bayes Classifier (NB)

Naive Bayes Classifier is one of the most practical methods of bayesian learning. Its performance is comparable to those of decision trees. In naive bayes classifier, each instance is described by a conjunction of attribute values and the target function needs to be predicted. In our case, instances are the sentences from the conversation text and the target function is binary, with values 0 or 1. Therefore, the naive bayesian classifier is applied as:

[5] `http://www.cs.waikato.ac.nz/ml/weka/`

$$P(c \in C | f_1, f_2 \cdots f_n) = \frac{P(f_1, f_2 \cdots f_n | c \in C) P(c \in C)}{P(f_1, f_2 \cdots f_n)} \tag{3.14}$$

where C is the set of target classes (i.e. in the summary, 0, or not in the summary, 1) and $f_1, f_2 \cdots f_n$ are the set of features. While training, it tries to maximize the probability of occurrence of events given any class. We use the Naive Bayes implementation available in Weka toolkit.

3.4.3 Multilayer Perceptron (MLP)

Multilayer perceptron is a feedforward neural network with one or more layers between input and output layers. It uses a supervised learning technique called backpropagation[6] for training the network. It has advantage over single layer perceptron, as it can distinguish linearly inseparable data. We use the MLP implementation from the Weka toolkit.

3.4.4 Support Vector Machine (SVM)

Support Vector Machines are supervised machine learning models which construct an N-dimensional hyperplane to separate the training data (positive and negative examples) into two categories with maximum margin. The testing data is classified depending on the side of the constructed hyperplane they fall on. The main advantage of SVMs is that they are robust in nature and eliminate the need for feature selection and manual parameter tuning. We use the LibSVM implementation from the Weka toolkit.

3.5 Evaluation

3.5.1 Dataset

In conversation summarization there is no standard dataset available to validate our approach. Most corpora do not have any model summaries associated with them to use as gold standards, which makes evaluation difficult in nature. We are dealing with conversations which are in the form of chats and emails. As guided by the work of others, Zhou and Hovy [57] and Uthus and Aha [51], we are aware of only one publicly available chat corpus with associated summaries, which is the GNUe Traffic archives. For emails, we have used BC3 corpus. Both these corpora are described in the following sections.

- **GNUe Archives**

  GNUe Archives is a collection of summary digests of discussions on GNUe development. Each digest contains human created summaries in the form of a newsletter based on IRC logs. The summaries are both extractive and abstractive in nature. However, most of the summaries are composed by directly quoting the most informative and relevant sentences from the chat logs.

[6] `http://en.wikipedia.org/wiki/Backpropagation`

These manual summaries form the gold-standard data for evaluation. We have a collection of 325 chat conversations and their corresponding summaries. There are multiple participants in each conversation with various topics being discussed throughout the conversation. The GNUe archives represent the real word chats, which are instantaneous in nature. This data can be downloaded from here[7].

- **The BC3 Corpus**

  The BC3 (British Columbia Conversation Corpus) is a collection of multimodal conversational data developed by Ulrich et al. [50]. The corpus contains 40 email conversations from the World Wide Web Consortium (W3C) mailing list. The conversations are in the form of email threads. It has 3222 sentences and an average of 6 emails per thread. The corpus provides both abstractive and extractive summaries. For extractive summaries, the email threads have been manually annotated at sentence level. For our purpose, we have considered manually created extractive summaries from the corpus as the gold-standard data for evaluation. The BC3 corpus is publicly available for use[8]. This data is free from errors such as spelling mistakes, abbreviations and emoticons, and the email is also segmented into sentences.

3.5.2 Experimental Setup

To evaluate our approach, we used ROUGE as a metric for summarization performance. ROUGE F-scores were used, as they represent both precision and recall aspects, for different matches: unigram (ROUGE-1), bigram (ROUGE-2) and longest subsequence (ROUGE-L). In our experiments we performed 5-fold cross-validation, where we divided the pre-annotated data into two parts, namely, training set and testing set. In other words, we used 80% of the data for training the summarizer and the remaining 20% for testing, where similar process was repeated 4 more times. While generating summaries we have limited the number of words to that in the model summaries.

We compare our methods with following most widely used baseline systems.

- MaxLength: Selects the longest sentences of every participant. This is a proven strong baseline for conversation summarization [19].

- HAL: A sentence extraction based system[9], where the sentences are selected based on the words score in the semantic space built using a lexical co-occurrence model [24].

- FirstSent: Selects the first sentence from each message in the conversation sequence (*Position Hypothesis*).

[7] `http://kt.earth.li/kernel-traffic/index.html`
[8] `http://www.cs.ubc.ca/labs/lci/bc3.html`
[9] Secured 1st position in Document Summarization task in DUC 2007

- DiaSumm: This system creates a summary by extracting an inter-connected structure of segments that quoted and responded to each other.

FirstSent and DiaSumm are hard-to-beat baselines and were used by Zhou and Hovy [57] while summarizing IRC logs. We also perform extractive summarization at different compression rates(from 10% to 25%) to see its effect on precision and recall.

In addition to this, we evaluate the spelling correction and text segmentation modules. To create dataset for this evaluation, we randomly selected 88 chats, where spelling mistakes were manually corrected and the chat was segmented wherever required such that each segment is a user utterance. We compare our Spelling Correction module's performance to a baseline of standard Aspell [2], always picking the first correction suggestion. For text segmentation module, we choose Hearst's TextTiling [22] as the baseline for evaluation. It is discussed in Section 3.2.2.

3.6 Results and Discussion

In this section we discuss the evaluation results of different experiments and compare them to state of the art approaches in the related fields.

3.6.1 Analysis of Spell Corrector

As mentioned in Section 3.2.1, our spelling corrector makes use of web. We have used the same evaluation procedure as used by Whitelaw et al. [55]. It focuses on metrics that reflect the quality of end-to-end behavior, and account for the combined effects of flagging and automatic correction. Precisely, there are three states: a word could be unchanged, flagged or corrected to a suggested word. We would like to state that all words which are flagged for spell correction will undergo changes. Following are the possible scenarios:

1. Type 1: A well spelled word is not flagged (Final State: Correct).
2. Type 2: A well spelled word is wrongly corrected (Final State: Incorrect).
3. Type 3: A misspelled word is not flagged or wrongly corrected (Final State: Incorrect).
4. Type 4: A misspelled word is corrected (Final State: Correct).

Spell Checker	Type 1	Type 2	Type 3	Type 4
Aspell	240690	4521	503	686
WebSpellCorrect	244789	422	230	959

Table 3.3 Performance Analysis of Spelling Correction Module for 88 Chat Conversations

In Type 3, we have combined the scenario when a misspelled word is not flagged or gets flagged but wrongly corrected, because both the situations finally lead to an incorrect word. We compare our algorithm for spelling correction with the baseline on all the above mentioned four categories. Table 3.3 shows that our approach (WebSpellCorrect) outperforms the baseline Aspell by a significant amount. We can see that the baseline introduces lot of Type 2 errors, which is more than the number of misspelled words corrected. On further investigation, we noticed that the technical terms are wrongly flagged by Aspell because of its limited vocabulary in case of such domains. It adds more noise to the documents in the form of misspelled words than initially present. On misspelled words, our system WebSpellCorrect achieves an accuracy of 80.65% compared to Aspell, which has an accuracy of 57.69%.

3.6.2 Analysis of Text Segmenter

Text segmenter is very relevant to summary generation as we treat it as a sentence extraction problem. For text segmentation, we evaluated both the approaches described in Section 3.2.2 for false positives and false negatives and found that our method LCM outperformed TextTiling. This is attributed to the fact that in the case of TextTiling algorithm, segmentation of text into discourse units requires subtopic structure, whereas the chats are spontaneous and unstructured in nature. Therefore, we have used LCM over TextTiling. Table 3.4 compares LCM with TextTiling on false positives and false negatives on 88 chat conversations having 12514 segments.

Text Segmenter	False Positives	False Negatives
LCM	112	875
TextTiling	832	1657

Table 3.4 Performance of Text Segmentation Modules on False Positives and False Negatives

3.6.3 Comparison of Feature Sets

We experimented with two feature sets, namely, **basic** and **complete**, as mentioned in Section 3.3. The **basic** feature set contains the standard features that can be used with other kinds of text genre as well. Some of these features have been used in the past by researchers for different information retrieval

Configuration	GNUe Archives			BC3 Corpus	
	basic	**complete**		**basic**	**complete**
ROUGE-1	0.35370	0.48192		0.54188	0.63724
ROUGE-2	0.22575	0.30132		0.38782	0.52285
ROUGE-L	0.33017	0.45521		0.52913	0.63005

Table 3.5 Feature Set Comparison for C4.5 on Chat and Email Conversations on ROUGE-F Scores

tasks. The **complete** feature set contains additional features which are specific to technical chats and email conversations. This bigger feature set gives more detailed information about the importance of each sentence with respect to the summary.

Table 3.5 shows that using **complete** feature set gives better results compared to **basic** feature set. This is because the new features added help us to better capture the important characteristics of the conversation such as dialogue structure and sentiment information. We have shown the results for both the feature sets on different corpora for Decision Trees. We got similar improvements when **complete** feature set was used over **basic** feature set with other classifiers as shown in Table 3.6, 3.7 and 3.8 respectively.

Configuration	GNUe Archives			BC3 Corpus	
	basic	**complete**		**basic**	**complete**
ROUGE-1	0.42370	0.57179		0.50901	0.58857
ROUGE-2	0.20375	0.36512		0.27477	0.45714
ROUGE-L	0.41024	0.56608		0.49549	0.57191

Table 3.6 Feature Set Comparison for NB on Chat and Email Conversations on ROUGE-F Scores

Configuration	GNUe Archives			BC3 Corpus	
	basic	**complete**		**basic**	**complete**
ROUGE-1	0.41812	0.55894		0.48702	0.56292
ROUGE-2	0.22493	0.36318		0.26574	0.43143
ROUGE-L	0.40736	0.53019		0.46061	0.55823

Table 3.7 Feature Set Comparison for MLP on Chat and Email Conversations on ROUGE-F Scores

Configuration	GNUe Archives			BC3 Corpus	
	basic	**complete**		**basic**	**complete**
ROUGE-1	0.39370	0.51341		0.41745	0.48118
ROUGE-2	0.19875	0.30765		0.23029	0.34172
ROUGE-L	0.38452	0.50282		0.40623	0.46122

Table 3.8 Feature Set Comparison for SVM on Chat and Email Conversations on ROUGE-F Scores

3.6.4 Effect of Compression Ratio on Precision and Recall

As we increase the compression ratio, the ROUGE recall scores increase for all algorithms, since with larger number of sentences in the summary the probability of the right sentences getting selected increases. The ROUGE precision scores decrease with increase in compression ratio but not in a monotonous manner. Figures 3.2 and 3.3 plot the ROUGE-1 Recall and Precision scores against compression ratio for different training algorithms for email conversations. ROUGE-2 and ROUGE-L Recall and Precision scores against different compression ratio is shown in Figure 3.4, 3.5, 3.6 and 3.7 respectively. All the methods improve significantly over the baseline approach. Only FirstSent and DiaSumm are shown, as they performed better than HAL and MaxLength for email conversations. Similar patterns are observed for ROUGE-1, ROUGE-2 and ROUGE-L scores for chat conversations, and therefore they are not shown here.

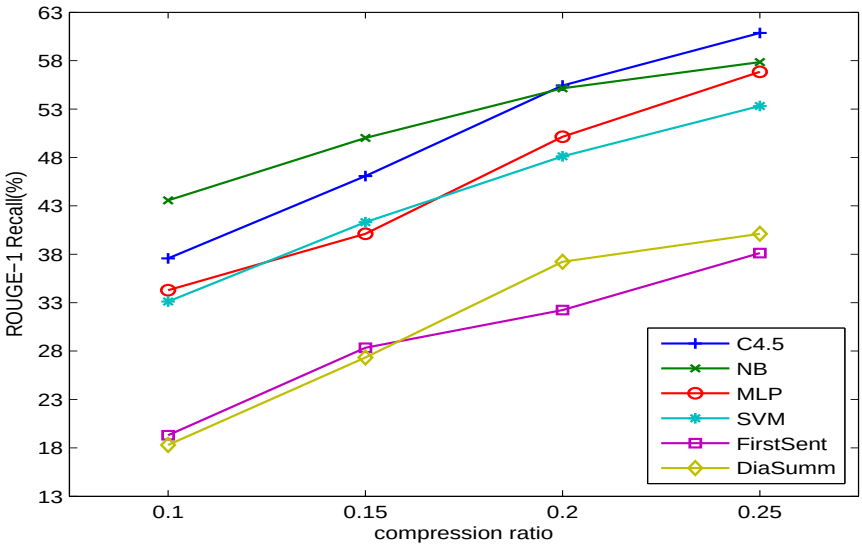

Figure 3.2 Effect of Compression Ratio on ROUGE-1 Recall for Email Conversations

There is relatively less improvement using a higher compression ratio(20% to 25%), compared to a lower one. This is reasonable because when the compression ratio is low, the most salient utterances are not necessarily the longest ones, thus using more information sources helps better identify important sentences; but when the compression ratio is higher, longer utterances are more likely to be selected since they contain more content.

3.6.5 Effect of Learning Algorithms

The learning algorithm used strongly influences the quality of results obtained. As mentioned in Section 3.4, we experimented with four different learning algorithms. In the case of GNUe archives, the best results were obtained by using **Naive bayes** classifier. MLP learning algorithm performed almost

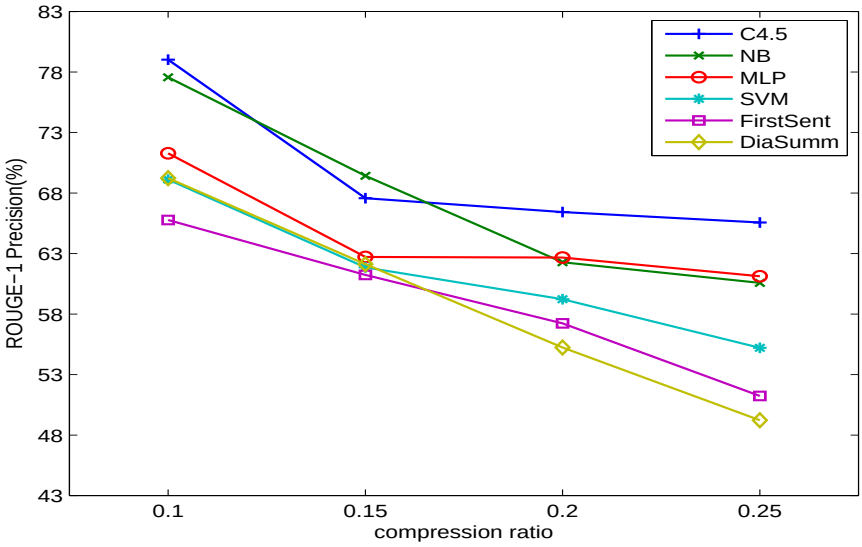

Figure 3.3 Effect of Compression Ratio on ROUGE-1 Precision for Email Conversations

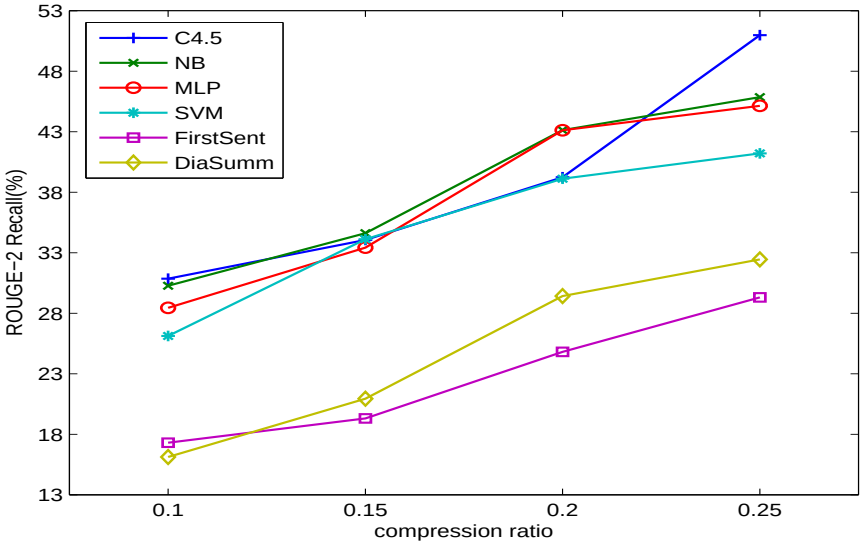

Figure 3.4 Effect of Compression Ratio on ROUGE-2 Recall for Email Conversations

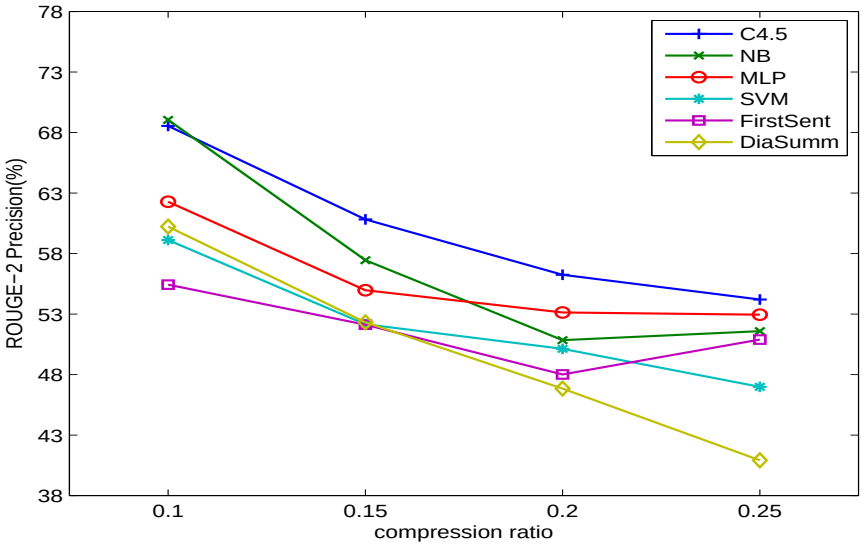

Figure 3.5 Effect of Compression Ratio on ROUGE-2 Precision for Email Conversations

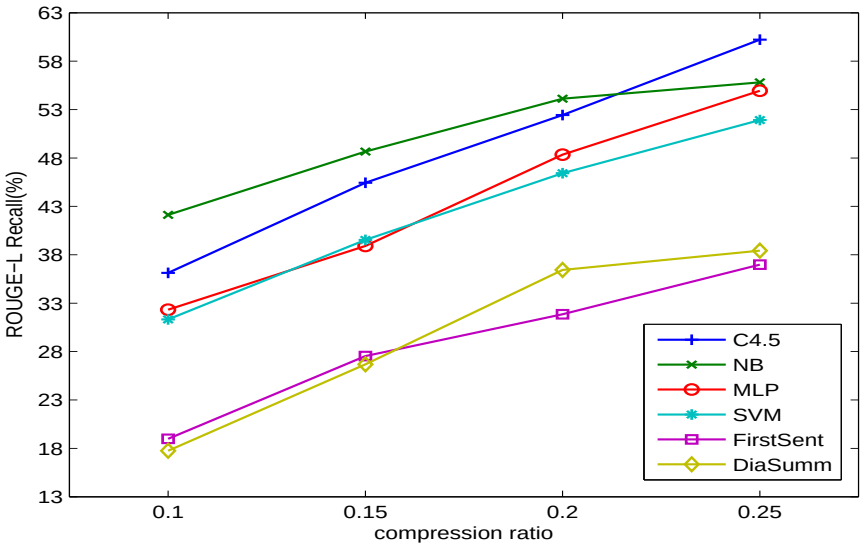

Figure 3.6 Effect of Compression Ratio on ROUGE-L Recall for Email Conversations

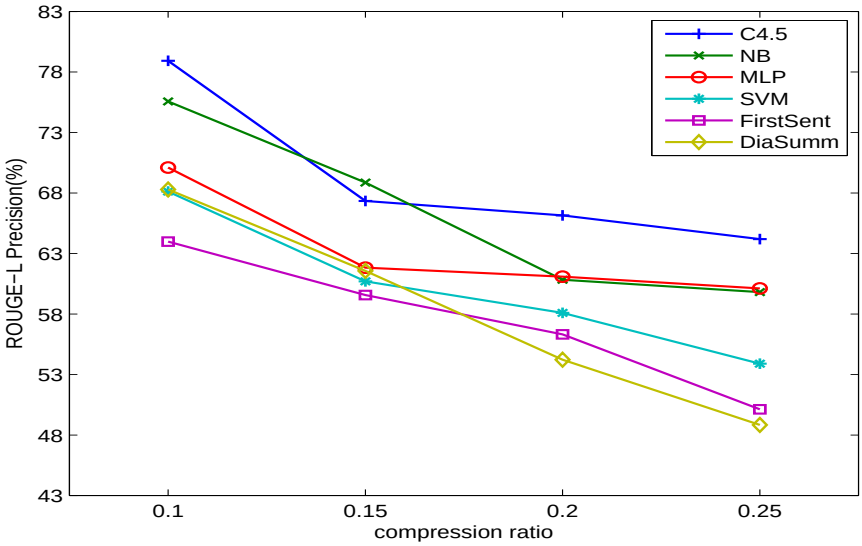

Figure 3.7 Effect of Compression Ratio on ROUGE-L Precision for Email Conversations

Configuration	ROUGE-1	ROUGE-2	ROUGE-L
MaxLength	0.44199	0.27713	0.42209
HAL	0.38874	0.20221	0.37989
FirstSent	0.27651	0.14916	0.26072
DiaSumm	0.42123	0.24495	0.41028
C4.5 (basic)	0.35370	0.22575	0.33017
NB (basic)	0.42370	0.20375	0.41024
MLP (basic)	0.41812	0.22493	0.40736
SVM (basic)	0.39370	0.19875	0.38452
C4.5 (complete)	0.48192	0.30132	0.45521
NB (complete)	**0.57179**	**0.36512**	**0.56608**
MLP (complete)	0.55894	0.36318	0.53019
SVM (complete)	0.51341	0.30765	0.50282

Table 3.9 Average ROUGE F-Scores for Chat Conversations

Configuration	ROUGE-1	ROUGE-2	ROUGE-L
MaxLength	0.27755	0.14909	0.26856
HAL	0.37527	0.24820	0.36104
FirstSent	0.41933	0.26926	0.40487
DiaSumm	0.43815	0.30736	0.41986
C4.5 (basic)	0.54188	0.38782	0.52913
NB (basic)	0.50901	0.27477	0.49549
MLP (basic)	0.48702	0.26574	0.46061
SVM (basic)	0.41745	0.23029	0.40623
C4.5 (complete)	**0.63724**	**0.52285**	**0.63005**
NB (complete)	0.58857	0.45714	0.57191
MLP (complete)	0.56292	0.43143	0.55823
SVM (complete)	0.48118	0.34172	0.46122

Table 3.10 Average ROUGE F-Scores for Email Conversations

as well. There was slight decrease in the performance for decision tree and SVM algorithms, but they were still able to outperform the baselines on the measures of ROUGE F-scores. Whereas in the case of BC3 corpus, **C4.5 decision tree** performed the best. Naive Bayes and MLP algorithms also performed relatively good.

All the machine learning algorithms we considered successfully completed the task of generating a reasonable summary, as all of them performed better than the baselines. We have also shown that our feature set is generic enough to learn from any of these algorithms. Table 3.9 and 3.10 shows the ROUGE-F scores for various classifiers (with both $basic$ and $complete$ feature set) on chat conversations and BC3 email conversations. All the algorithms performed better than the baselines indicating that the **complete** feature set is a correct way to represent the sentences. All the results presented are statistically significant at 99% significance level. We used paired t-test for testing statistical significance. In our summarizer we have explored a large variety of features, where some of them have been previously applied to different type of information retrieval tasks. We have done a fine-grained analysis of the text and have employed features at word level, sentence level and discourse level to best represent the information present in the sentence.

3.7 Summary

In this chapter, we proposed a supervised machine learning approach to summarize online conversations. We developed a three phase architecture to process the text at different levels of granularity, involving word level, phrase level and sentence level. In the first phase, we normalized the data for usage by extracting the text, correcting word spellings and segmenting the text. In the second feature extraction phase, we represented each sentence as a collection of features. These features include both existing features like TF-IDF, SFS, which are categorized under basic feature set, and newly devised features as Discourse Marker, Sentiment Score which belong to complete feature set. These features

capture the statistical, linguistic and sentimental aspects along with the dialogue structure of the conversation. In the last phase, we trained our summarizer using a machine learning algorithm on a set of feature vectors obtained from previous phase.

We showed that newly introduced features significantly improved the summarization performance. Furthermore, we experimented with various trainable machine learning algorithms and saw that our feature set is generic enough to learn from any of the algorithms. We tested the robustness of our features by comparing our results with most widely used systems for conversation, and observed that the performance gain is statistically significant in nature. Thus, we conclude that we have successfully summarized real-time chat and email conversations.

Chapter 4

Topic-Focused Approach to Conversation Summarization

As noted previously, the enormous increase in web technologies has led to the exponential growth of user-generated content. A large number of people express their views and opinions via web on various public forums in the form of chat, emails and blogs. As human beings, we are always in quest to seek more information, learn more and solve issues with the intent of contributing to the society.

Public sites like Stackoverflow, IRC logs, Google groups contains vasts amount of information which may not be present in structured knowledge bases like Wikipedia, ODP that do not get updated very frequently. Real-world issues are being discussed in such public forums resulting in more interaction between people from different parts of the world. It has also attracted many organizations, companies and individuals. Summarizing these expressed viewpoints and discussions can be useful for many such stake holders. In this chapter, we investigate real-world chat conversations from technical domain and provide a strong framework for summarizing them.

4.1 Motivation

In the previous chapter, we discussed a supervised machine learning framework for conversation summarization. We devised new features to investigate different aspects of conversations, like dialogue structure, sentiment, linguistics to name a few. Experiments showed that the features are robust and can produce effective summaries. In-spite of the significant performance gain, our summarization framework has one disadvantage. Our machine learning approach is supervised in nature and requires training data which is composed of conversations and the corresponding summaries. Since summaries need to be produced for training set, thus, comes the role of the human annotators. Human intervention leads to a slow and expensive process of generating manual summaries. It also introduces the problem of feature biasness, since summarization is a cognitive process.

In this chapter, we discuss a new topic-focused unsupervised approach towards summarization of conversations. Real-time conversations attract a substantial number of participants and a single conversation tends to span a wide range of topics interspersed with irrelevant segments. Conversations often digress from the main topic of discussion which may not be relevant to information seeking user. There-

fore, motivated by this, we develop a new framework where the primary topic of discussion guides the summarization process. To find the hidden topics of discussion in the conversation we use topic modeling. However, because of the sparseness in data identifying the connected components and the primary topic of discussion becomes a difficult task. The lack of evidence in linking different entities/objects and their corresponding attributes is a hinderance to the statistical topic modeling. We tackle this problem by using web information during topic modeling to find the hidden topic structure. The primary topic guides the summarization process and is incorporated into relevance based language models. We build a semantic word space to score sentences based on their association with the primary topic.

In the following section, we discuss the motivation and idea behind choosing Topic modeling for primary topic detection.

4.2 Introduction to Topic Modeling

Real-time social web outputs a tremendous amount of information in the form of news, blogs, Web pages, scientific articles, books, images, sound and video making it more difficult to find and discover what we are searching for. The challenge is mining and managing this data is not only due to large volume, but also due to its noisy nature since it belongs to the category of UGC.

We can look at summarization as an information retrieval problem, where we aim to retrieve the relevant sentences, which constitute to form the summary. However, organizing, searching and retrieving information using only keywords is not every effective. There is a need to find a semantic way to associate such data. Therefore instead of finding relevant documents through keyword based search alone, it is better to first find the theme/topic that we are interested in, and then examine documents related to that theme. To achieve this goal, researchers have developed Topic Models. Topic models are a suite of algorithms that uncover the underlying hidden semantic structure of a document collection based on hierarchical Bayesian analysis of the original texts [7, 8, 20]. In simple words, topic modeling algorithms are statistical methods that analyze the words of the original text to discover the theme running through them and how these themes are related to each other. These are unsupervised models and do not require any prior annotations or labeling of the documents - the topics emerge from the analysis of the original texts.

Topic models have been applied to different kinds of media, including emails [37], scientific abstracts [8, 20], and newspapers archives [54]. Adaption of topic models to different kinds of data has resulted in many applications, some of which includes analyzing and finding patterns in genetic data, images, and social networks. Topic modeling can be done using various algorithms. One of the earliest well known topic model is Latent Semantic Indexing (LSI), which is also known as Latent Semantic Analysis [13, 15]. This model was used extensively by research community as a model to improve search results with considerable success. But it suffers from the problem of unscalability as in case of large Internet scale datasets. However, a major drawback of LSI is that it assigns each document to a

single category, which is considered incorrect since a single document can represent multiple topics to different extent.

To overcome the drawbacks of LSI, Probabilistic Topic Models (PTM) [6, 23, 48] were developed. It is a soft clustering algorithm which assign documents to multiple topics with a probability metric. One of the most well studied and simple PTM is the Latent Dirichlet Allocation [8, 48] or LDA model. We use LDA for topic modeling, and will discuss the same in the following section.

4.2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is an unsupervised, probabilistic, text clustering algorithm. LDA is similar to LSI, allowing documents to be categorized into semantic topics. It is however a soft clustering algorithm since it defines each document as a distribution over these topics unlike LSI where each document is part of a single semantic cluster.

LDA can be defined as a generative three level hierarchical Bayesian probabilistic model for collections of discrete data such as text documents. The documents are modeled as a finite mixture over an underlying set of topics which, in turn, are modeled as an infinite mixture over an underlying set of topic probabilities. Thus in the context of text modeling, the topic probabilities provide an explicit representation of the documents. A Bayesian network allows for the definition of a probabilistic process thus giving more operational flexibility compared to LSI.

Let us assume the following notations. The topics are $\beta_{1:K}$, where each β_k is a distribution over the vocabulary of the documents. The topic proportions of the documents are $\theta_{1:D}$, where θ_d is the topic proportion of the d^{th} document. The topic assignments for the documents are $z_{1:D}$, and $z_{d,n}$ is the topic assignment for the n^{th} word in document d. The observed words for the documents are $w_{1:D}$, where $w_{d,n}$ is the n^{th} word in document d. LDA assumes a generative process by which the documents emerged, corresponding to a joint distribution of hidden and observed variables, as given by

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d)(\prod_{n=1}^{N} p(z_{d,n}|\theta_d)p(w_{d,n}|\beta_{1:K}, z_{d,n})) \tag{4.1}$$

Here $\beta_{1:K}, \theta_{1:D}$ and $z_{1:D}$ are the hidden variables which represent the topic structure, and $w_{1:D}$ are the observed variables which represent the documents. The computational problem of LDA is to use the observed documents to discover the hidden topic structure.

4.3 System Architecture

We propose a two phase architecture, which contains a topic modeling phase followed by a sentence scoring language modeling phase. In the first phase of our algorithm, we apply topic modeling to find the primary topic of discussion in the conversation. We expand the concepts present in the conversation, using documents from web, to discover the topics more accurately. Then, in the summary generation

phase, we build a semantic word space to score sentences based on their association with the primary topic. The top ranked sentences are then picked to form the summary. Figure 4.1 shows the architecture of our system.

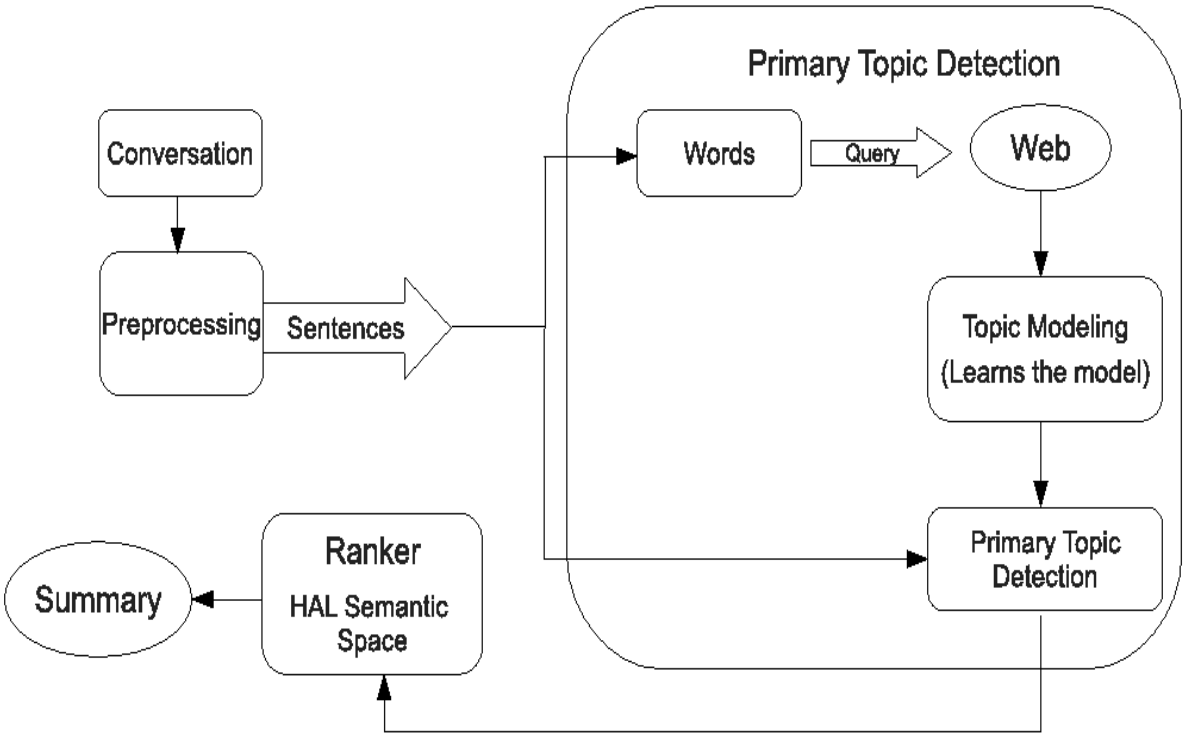

Figure 4.1 System Architecture for Topic-Focused Summarization Framework

4.4 Primary Topic Detection

We apply topic modeling to automatically discover the topics underlying the conversation. We use Latent Dirichlet Allocation (LDA) for topic modeling.

4.4.1 Basic Topic Modeling

Before discussing our main approach, we describe a simpler approach of discovering the topic structure of the conversation. In this approach, we consider each sentence to be a document and apply LDA to the collection of sentences in the conversation. Thus the hidden topic structure of the conversation is discovered. For summary generation, we need the primary topic of discussion in the conversation. We define the *primary topic* as the most prevalent topic in the longest sentences of the conversation. The

topic proportion of each topic in the longest sentences are added up and the topic having the highest sum is considered as the most prevalent and hence chosen as the primary topic.

4.4.2 Topic Modeling with Web Referencing

As shown in the section 4.7.3, we obtained reasonably good results when summaries are generated by employing the above approach. But since the sentences are short in length, thus lacking in sufficient information, a better approach would be to use certain external information to augment the information present in the conversation. This is because when provided with more information, the model would be able to capture the topic structure more accurately. Therefore, we used documents from web to provide extra information. For every word, except the stop words, a query is made to the web using Bing search API. We fetch the first web document obtained as result and that document is considered to be a description of the word queried. Each sentence in the conversation is thus associated with the set of web documents corresponding to the words in that sentence. For every sentence, its corresponding web documents are concatenated and considered as one single document. LDA is applied to this collection of expanded documents and the corresponding hidden topic structure is discovered. The topic distribution of every document in this collection can be considered as the topic distribution of the corresponding sentence. Thus, the topic distribution of every sentence in the conversation is obtained.

The primary topic is identified in the same manner as in the previous approach. In this case, the topics are distributions over the vocabulary of the web documents. However, for summary generation we require the primary topic to be a distribution over the vocabulary of the conversation. The trained model, that has been obtained by applying LDA to the set of web documents, is applied to the conversation considering each sentence as a document. On applying this, we obtain for each topic a distribution over the vocabulary of the conversation.

4.5 Sentence Ranking Using Relevance Based Language Modeling

After topic modeling phase, we get the primary topic of the conversation. We incorporate this topic information into a semantic co-occurrence model using relevance based language modeling approach to generate summaries.

We will start by giving the background of language models and then discuss the motivation behind choosing relevance based language models for sentence ranking.

4.5.1 Introduction to Language Models

A language model or alternatively a statistical language model is a probabilistic mechanism for generating text. Language modeling has been extensively used by researchers in different domains such as speech recognition, machine translation, part-of-speech tagging, parsing and information retrieval.

In applications of language modeling to IR, both query application and retrieval of relevant documents are modeled as simple probability mechanisms. Query formulation is modeled as translation of the user information need. Once the query is issued, an IR system has to retrieve all relevant documents from the corpus. This is very similar to our summarization framework where we extract most relevant sentences instead of documents to form the summary. Retrieving the relevant documents is modeled as the generation of query from a relevant document, that is, each document is treated as a language sample and query as a generation process. Each document is ranked based on the probability of generating the query from the corresponding language model. A document more likely to generate query is treated as more relevant compared to a document which is less likely to generate the same query. But such statistical language models suffer from the sparseness of data to compute the document model, that is, many documents are small in length and may not contain many words in vocabulary which leads to a sparse language model. Relevance based language models are a significant improvement over language models when no training data is provided. We will discuss the same in next section.

4.5.2 Relevance Based Language Modeling

Relevance based language models assumes both the query and the document as samples from an unknown relevance model R, hence it is able to overcome the problem of sparseness in the training data. The relevance based language models approximate $p(w|R)$, the probability of observing a word w in documents relevant to a particular information need R, where $q = q_1, q_2 \cdots q_k$ is the information need as query with q_i representing the i^{th} word of the query. It can be expressed as,

$$p(w|R) \approx p(w|q) = p(w|q_1, q_2 \cdots q_n) = \frac{p(w, q_1 \cdots q_k)}{p(q_1 \cdots q_k)} \tag{4.2}$$

The joint probability in Equation 4.2 can be decomposed using chain rule and assuming query words, $q_1, \cdots q_k$ to be independent of each other while keeping the dependencies on w intact. Following this assumption joint probability can be calculated as,

$$p(w, q_1, \cdots q_k) = p(w) \prod_{i=1}^{k} p(q_i|w) \tag{4.3}$$

The required dependencies can be obtained by the bigram model, but that will again lead to data sparseness problem. A better way would be to capture the semantic relatedness between words in the form of term dependencies. This can be achieved using the probabilistic interpretation of Hyperspace Analogue to Language (HAL) model.

4.5.3 Hyperspace Analogue to Language Model

The Hyperspace Analogue to Language (HAL) model [24] constructs the dependencies of a word w on other words based on their occurrence in the context of w in a sufficiently large corpus. It is based on

the concept of distributional similarity [29]. As stated by Song and Bruza [47], the intuition underlying HAL spaces is that when humans encounter a new concept, they tend to derive its meaning from the accumulated experience of the context in which the concept appears. Lund and Burgess [34] discuss the use of lexical co-occurrence to construct high dimensional semantic spaces in which a word can be represented as a point.

The semantic co-occurrence can be captured by building a $term \times term$ matrix. The weights assigned to each co-occurrence of terms are accumulated over the entire corpus. That is, if $matrix(w', k, w)$ denotes the number of times word w' occurs k distance away from w when considered a window of length K, and $W(k) = K - k + 1$ denotes the strength of this co-occurrence between the two words, then HAL co-occurrence score is given by:

$$HAL(w'|w) = \sum_{k=0}^{K} W(k) \times matrix(w, k, w') \tag{4.4}$$

The length of the window size invariably influence the quality of the association between a pair of terms. As the window size increases, the higher the chance of representing spurious associations between terms. There is no way to determine what the best size of window is, but experiments [34] suggest the window of size 8 in context of IR.

The HAL space built this way can be used to create a probabilistic model to give scores. Lowe [32] has shown how such a co-occurrence matrix can be transformed to give probabilistic scores by normalizing the count of the terms. The probabilistic version of HAL, pHAL, gives the probability of associating a word w' with another word w in a window of size K. This can be expressed in terms of probability of observing w' at a distance of $k < K$ from w, as

$$pHAL(w'|w) = \frac{HAL(w'|w)}{n(w) \times K} \tag{4.5}$$

where $n(w)$ is the frequency of w.

After HAL Space is built, scoring of a sentence is done by accumulating scores over terms associated with the primary topic which was discovered during the topic modeling phase. Given topic terms, t_1, t_2 ... t_k, score of a sentence S is given by

$$Score(S) = \prod_{w_i \in S} \left(P(w_i) \times \prod_{t_k} pHAL(t_k|w_i) \right) \tag{4.6}$$

The top ranked sentences are selected and ordered chronologically to form the summary.

Algorithm 2 HAL Semantic Space Construction

Initialize window size K = Average Length of Sentence in the Conversation
for $i = 0 \rightarrow size(chatDocument)$ **do**
  for $j = i + 1 \rightarrow i + K - 1$ **do**
    matrix$(w_i, j - i, w_j)$++
    matrix$(w_j, j - i, w_i)$++
  end for
end for
Initialize $V = UniqueWords(chatDocument)$
for each $w \in V$ **do**
  for each $w' \in V$ **do**

$$HAL(w, w') := \sum_{k=0}^{K} W(k) \times matrix(w, k, w')$$

$$pHAL(w, w') := \frac{HAL(w, w')}{n(w) \times K}$$

  end for
end for

where w_i and w_j represent the i^{th} and j^{th} words of the document.

4.6 Evaluation

4.6.1 Dataset

We use the same datasets, GNUe Archives and BC3 Corpus, for evaluation that we used in the previous chapter while evaluating machine learning framework for summarization. We have used a collection of 450 chat conversations from GNUe Archives with corresponding model summaries as gold standards for evaluation. For emails we have used 40 email thread conversations and their extractive from BC3 corpus for evaluation. We have discussed the properties of both these datasets in Section 3.5.1.

4.6.2 Experimental Setup

In the topic modeling phase, the number of topics in LDA model has to be predetermined. To the best of our knowledge, there is no method to estimate the number of topics which best captures the topic structure. Therefore, we experimented with a range of values, between 5 to 100, for a set of 60 conversations, and decided to use 20 as the number of topics. For evaluation, we used the remaining 390 chats and 40 email threads.

We use ROUGE as the evaluation metric for summarization performance. We used ROUGE-F scores for unigram (ROUGE-1), bigram (ROUGE-2) and longest subsequence (ROUGE-L) matching. Length

of the generated summaries were restricted to those of model summaries. We compare our methods with the following most widely used baseline systems.

- MaxLength: Selects the longest sentences of every participant. This is a proven strong baseline for conversation summarization [19].
- LexCocc: A sentence extraction based pHAL system[1], where the sentences are selected based on the words score in the semantic space built using a lexical co-occurrence model [24].
- FirstSent: Selects the first sentence from each message in the conversation sequence (*Position Hypothesis*).
- DiaSumm: This system creates a summary by extracting an inter-connected structure of segments that quoted and responded to each other.

DiaSumm and FirstSent are hard-to-beat baselines and were used by Zhou and Hovy [57] while summarizing IRC logs. LexCocc has been chosen to show the effect of topic modeling and web referencing on the model proposed.

4.7 Results and Discussion

4.7.1 Effect on Selecting the Number of Topics

In our approach, we find the primary topic of discussion and incorporate that information in HAL model for sentence ranking. We have used LDA for topic modeling which requires the number of topics T as input. Selecting T is one of the most problematic modeling choices in finite topic modeling. Not only there is no clear method for choosing T (other than evaluating the system on datasets for various values of T), but the degree to which LDA is robust to a poor setting of T is not well-understood [52].

Topics	ROUGE-1	ROUGE-2	ROUGE-L
5	0.22893	0.14826	0.21917
10	0.45651	0.35731	0.43862
15	0.54714	0.35105	0.53698
20	**0.59978**	**0.39284**	**0.58215**
25	0.54130	0.34952	0.52961
30	0.55914	0.37157	0.54816
45	0.52376	0.30182	0.51534
90	0.51262	0.30946	0.49813

Table 4.1 Effect of Number of Topics on ROUGE Scores for Chat Conversations

[1]Secured 1st position in Document Summarization task in DUC 2007

Ideally, if LDA has sufficient topics to model documents, then topic distributions and assignments of tokens to topics should be relatively invariant to an increase in T. This means that if fifty topics are sufficient to accurately model the data, then increasing the number of topics to seventy should not change the topic assignment and distribution significantly. We experimented the same with a range of T values to see its effect on summary generation. As shown in Tables 4.1 and 4.2, with the increase in T, the ROUGE score initially increases and later becomes constant with minor decrease in value. In our approach, only the primary topic of discussion is relevant to us. The stabilization of the ROUGE scores for $T > 20$ implies that the word distribution of the primary topic changes only marginally. In case of chat conversations, 20 were the optimal number of topics, whereas for email conversations 25 gave the best results. However, we chose 20 as the common value for both chat and email conversations for our experiments.

Topics	ROUGE-1	ROUGE-2	ROUGE-L
5	0.20718	0.12283	0.18812
10	0.37819	0.24172	0.35803
15	0.46926	0.34317	0.45092
20	0.57461	0.44192	0.55964
25	**0.60924**	**0.49716**	**0.59168**
30	0.56170	0.42974	0.55765
45	0.55637	0.39986	0.53928
90	0.54918	0.38237	0.52981

Table 4.2 Effect of Number of Topics on ROUGE Scores for Email Conversations

4.7.2 Impact of Web Reference on Topic Modeling

Using web reference module with topic modeling increases the performance by capturing the topic structure more accurately. An improvement in topic structure implies that the primary topic is more likely to be discovered. The word distribution of the primary topic is responsible for summary generation. Table 4.3 and Table 4.4 shows that there is significant improvement in summaries for chat and email conversations when documents from web are modeled for topics.

Configuration	ROUGE-1	ROUGE-2	ROUGE-L
pHAL+LDA	0.46192	0.26654	0.43562
pHAL+LDA+WR	0.59978	0.39284	0.58215
Performance Gain	**0.13786**	**0.12630**	**0.14563**

Table 4.3 Average ROUGE F-Scores for Chat Conversations

Real-world conversations are quite sparse, that is, there is absence of explicit relationship between entities, objects or topic of discussion. Conversations are often carried out with the assumption that

Configuration	ROUGE-1	ROUGE-2	ROUGE-L
pHAL+LDA	0.48745	0.36586	0.47056
pHAL+LDA+WR	0.57461	0.44192	0.55964
Performance Gain	**0.08716**	**0.07606**	**0.08908**

Table 4.4 Average ROUGE F-Scores for Email Conversations

participants involved have enough background to understand the concepts and ideas which form the basis of the conversation. By using web as a resource, we discover more information about the topic of discussion and get evidence to link various concepts which are part of discussion.

4.7.3 Evaluation of Summaries

Our methods outperform the baseline systems for chat and email conversations as shown in Table 4.5 and Table 4.6. Even our simpler approach of topic-focused summarization (pHAL + LDA) achieving better performance than LexCocc indicates that incorporation of topic information is beneficial. Using Web Reference (WR) module with LDA further increases the performance by capturing the topic structure more accurately.

In case of chat conversations, our main approach (pHAL + LDA + WR) beats the baselines by a significant margin on ROUGE-1 (+0.17567), ROUGE-2 (+0.14357) and ROUGE-L (+0.16273) F-scores. LexCocc (pHAL) underperforms other baselines (MaxLength and DiaSumm) because there is a relatively lesser co-occurrence of related terms in spontaneous chats. The replies are more specific with very less lexical overlap and do not repeat the same thing again unlike formal emails, where we reiterate on the subject of discussion. For email conversations, DiaSumm and LexCocc performs comparatively well because of the lexical overlap. The difference in ROUGE-F scores for ROUGE-1 (+0.15114), ROUGE-2 (+0.12079) and ROUGE-L (+0.15027) shows the effectiveness of our approach.

Our simpler approach of (pHAL + LDA) defeats all the baselines, which indicates that using topic information for guiding summarization process leads into more meaning and accurate summaries. The results presented are statistically significant at 99% significance level. We used paired t-test for testing statistical significance.

Configuration	ROUGE-1	ROUGE-2	ROUGE-L
MaxLength	0.42411	0.24927	0.41942
DiaSumm	0.40823	0.23726	0.39315
FirstSent	0.27913	0.15398	0.27016
LexCocc(pHAL)	0.39521	0.21591	0.38128
pHAL+LDA	0.46192	0.26654	0.43562
pHAL+LDA+WR	**0.59978**	**0.39284**	**0.58215**

Table 4.5 Comparison with Baselines on ROUGE F-Scores for Chat Conversations

Configuration	ROUGE-1	ROUGE-2	ROUGE-L
MaxLength	0.27755	0.14909	0.26856
DiaSumm	0.43815	0.30736	0.41986
FirstSent	0.41933	0.26926	0.40487
LexCocc(pHAL)	0.37527	0.24820	0.36104
pHAL+LDA	0.48745	0.36586	0.47056
pHAL+LDA+WR	**0.57461**	**0.44192**	**0.55964**

Table 4.6 Comparison with Baselines on ROUGE F-Scores for Email Conversations

4.8 Summary

In this chapter, we developed a topic-focused approach to summarize text conversations in the form of chat and emails. We did semantic analysis of conversations using topic modeling to find the hidden topic structure. This topic information is incorporated into HAL model to generate better summaries.

We tried to calculate the probability that sentences are related to user's information need. The required term dependencies of a pair of words are captured using hyperspace analogue to language spaces. Previous studies [34, 47] suggest that these spaces capture the semantic dependencies between the words. Based on the past experiments [34], we set the window size to be equal to 8, while constructing the HAL matrix. If the window size is set to 1, then this model reduces to vector space model with the number of key words matched as the relevance score of the sentence towards the information need. Window size depicts the context we are likely to consider. HAL is based on the concept of distributional similarity, and therefore, prefers the sentences which contain words that co-occur highly with the user's query words. We use primary topic of discussion as query for the HAL model since there is no query specified by user. In order to find the primary topic of discussion we leverage topic modeling using web as a resource.

We showed that using web reference module with topic modeling can capture the topic structure more accurately. Topics are used as an anchor text for summarization and guides the summarization process during the sentence ranking phase. For sentence ranking, we build a semantic co-occurrence HAL model to capture the term dependencies effectively compared to n-gram models. In contrast to N-gram models, HAL finds the association between terms using a semantic space and is more effective against sparse text. Experiment show that there is a significant performance gain in ROUGE-1, ROUGE-2 and ROUGE-L scores for both chat and email conversations.

Chapter 5

Conclusions

The growth of web technologies in social media has enhanced the user experience and led to increased collaboration. As users continue to create, share and exchange information through social media channels in the form of written text conversations, the volume of user-generated content increases manifold leading to the problem of information overload. The content remains critical to enable a variety of applications that build on this useful source of information. Information if extracted from such online conversations can be of very good educational and commercial value. Text analytics for such data presents many new challenges for research and development, and has also gained interest from industry.

In this thesis, we devise methods to effectively mine information from user-generated content of written conversations and present it in the form of a summary. Summarization is a well known information access technique to deal with the problem of information overload and redundancy, however relatively less work has been done to summarize conversations. Most of the existing summarization systems have been built on the premise with news articles as input. The noisy and unstructured nature of conversations makes Natural Language Processing (NLP) techniques difficult to apply. Unlike regular documents, conversations have features like false starts, spelling mistakes and unsegmented text which makes traditional document summarization methods unsuitable for conversations. The problem of data sparsity further raises new challenges in text analytics of conversations from social media. We have examined the difficulties put forth by user-generated content for text analysis and developed new framework exploiting the semantic and syntactic structure of conversations.

We proposed a supervised machine learning approach where sentences were modeled as a set of features. These newly devised features capture the statistical, linguistic and sentimental aspects along with the dialogue structure of the conversation. We used a good number of features ranging from discourse marker and sentiment score to statistics of document collection like TF-ISF, SFS, SLS and heuristic based features like CHS, IQ for the experiments. We carried out an extensive analysis over both the feature sets to understand the impact of discourse analysis and sentiments on capturing information. Our work has highlighted the importance of understanding discourse structure and sentiment involved in the conversations for the generation of better summaries. We have used Rhetorical Structure Theory to explain the structure and formation of text. In our approach, we have automated the task of discourse

marker in RST using lexicon. We have shown the robustness of our proposed features by using different machine learning algorithms. Experiments have supported our intuition and we compared our system with hard-to-beat baselines in the area of conversation summarization.

The genericness of our features with respect to their applicability to both email and chat conversations differentiates it from any of the existing machine learning approaches for summarization. Instead of choosing features which are specific to emails or chats, we have developed a more general framework where characteristics of conversations are investigated. All features are easily pluggable into our framework, thereby providing more flexibility.

We took this study a step forward by studying the latent topics and intent underlying conversations rather than processing only statistical features. We proposed a new topic-focused approach where the main topic of discussion from the conversation guides the summarization process. Contrary to the supervised machine learning model that we developed, in this approach we did not require any training data. We leveraged topic modeling to discover the hidden topic structure. Topic distribution was then used to find the primary topic of discussion. This topic information when incorporated into a semantic co-occurrence HAL model led to the generation of improved summaries.

Most of the existing document summarization methods decompose the document into sentences and work directly in sentence space using term-sentence matrix. We exploited the knowledge on the document side, that is, the topics embedded in the documents to better understand the context and guide the sentence selection in the summarization procedure. There is very less lexical overlap in case of real-time text conversations, therefore, frequency based measures like TF-IDF, centroid are not effective measures for primary topic detection. We built a semantic model where instead of doing a term based match, we have done the concept match using the measures of distributional similarity.

Our contribution is in leveraging information and in getting assistance from greater knowledge sources like Web when working on user-generated content while discarding all the help from language tools. We have used Web as a knowledge source over more structured knowledge bases like Wikipedia and Open Directory Project because Web is dynamic in nature and contains more detailed information about the topics contained in the conversations. In-depth information from the Web helps us in finding the relevant topics using topic modeling. We developed a semantic model to tackle the problem of data sparsity. We built a sequential summarization model where in both the topic detection phase as well as the sentence scoring phase has explored the semantics of the conversation.

As we did not make use of any natural language processing tools, this work can be extended to other languages as well. Our approach is completely devoid of language or domain of the conversations, thereby, providing more flexibility. We have experimented with real-time conversations from technical domain where there is no pre-notion or etiquette of discussion. We used ROUGE as the evaluation metric for automatic summarization, and found that the performance gain is statistically significant at 99% significance level. From the results we conclude that we have successfully summarized real-time conversations. Our research work is one of the most premier works in the area of conversation summarization.

5.1 Future Directions

In this section, we will mention some of the possible future extensions of this research. In this thesis, we focused on summarization of conversations belonging to user-generated content. Although the techniques proposed here are adaptable across other domains it will make an interesting problem to apply the topic-focused summarization framework to monologues like news articles or blogs. Topic-focused summaries of news articles would be lot more accurate and valuable to users. News articles have high lexical overlap with no topic shifts, therefore it is a perfect usecase for our methods.

The summarization frameworks that we have developed are solely based on the analytics of the textual information and concepts present in the conversation. The summary generation process is guided by the primary topic of discussion, which is the most discussed topic of conversation. However, the summary generation can also be modeled from the viewpoint of the participants of the conversation. Therefore, inclusion of views of main participants and the ideas championed by them in the main summary can lead to the generation of better summaries.

In the machine learning approach, we have exploited the *relations* aspect of Rhetorical Structure Theory to generate better summaries. However the process of relation discovery is based on lexicon which has to be manually gathered and involves human intervention. Automating RST without external resources can be considered as an essential future problem as it would remove the bottleneck of human annotation, which itself is a costly and time consuming process. Both the approaches that we have developed require very little (Feature-Based Approach) or no (Topic-Focused Approach) use of natural language processing techniques. The modules developed at the different stages of summarization are language independent and hence we would like to investigate the language independence of our methods. The rate at which the information is growing it is very important to build a multilingual summarization system and our research could be a stepping stone towards achieving that goal.

The approaches we have developed have significant performance gain over existing state of the art systems with ROUGE as the evaluation metric. But it does not produce high linguistic quality summaries since there is no special care taken concerning readability issues of the summary. There is a lot of scope to work on improving grammatical quality and coherence in summary through co-reference resolution along with content quality. The current state of the art summarization systems are all extractive in nature, but the community is gradually progressing towards abstractive summarization [18]. Although a complete abstractive summarization would require deeper natural language understanding and processing, a hybrid or shallow abstractive summarization can be achieved through sentence compression and textual entailment techniques. Textual entailment helps in detecting shorter versions of text that entail with same meaning as original text. With textual entailment we can produce more concise and shorter summaries.

Research in summarization continues to enhance the diversity and information richness, and strive to produce coherent and focused answers to users information need.

Related Publications

- **Arpit Sood**, Thanvir Mohamed and Vasudeva Varma, "Topic-Focused Summarization of Chat Conversations", *In 35th European Conference on Information Retrieval (ECIR '13), Digital October Center, Moscow, Russia. (2013)*
- **Arpit Sood**, Thanvir Mohamed and Vasudeva Varma, "Summarizing Online Conversations: A Machine Learning Approach", *In 6th Workshop on Analytics for Noisy Unstructured Text Data, in conjunction with 24th International Conference on Computational Linguistics (COLING '12), Mumbai, India. (2012)*
- Sudheer Kovelamudi, Sethu Ramalingam, **Arpit Sood** and Vasudeva Varma, "Domain Independent Model for Product Attribute Extraction from User Reviews using Wikipedia", *In 5th International Joint Conference on Natural Language Processing (IJCNLP '11), Chiang Mai, Thailand. (2011)*

Other Publications

- Vasudeva Varma, Sudheer Kovelamudi, Jayant Gupta, Nikhil Priyatam, **Arpit Sood**, Harshit Jain, Aditya Mogadala and Srikanth Reddy Vaddepally, "IIIT Hyderabad in Summarization and Knowledge Base Population at TAC 2011", *In Proceedings of Text Analysis Conference (TAC '11), National Institute of Standards and Technology Gaithersburg, Maryland USA. (2011)*

Bibliography

[1] R. Arora and B. Ravindran. Latent dirichlet allocation based multi-document summarization. In *Proceedings of the second workshop on Analytics for noisy unstructured text data*, number 7 in AND '08, pages 91–97, New York, NY, USA, 2008. ACM.

[2] K. Atkinson. Gnu aspell. `http://aspell.net/`, 2011.

[3] R. Barzilay and M. Elbadad. Using lexical chains for text summarization. 1997.

[4] J. Bengel, S. Gauch, E. Mittur, and R. Vijayaraghavan. Chattrack: Chat room topic detection using classification. In *Intelligence and Security Informatics*, pages 266–277, 2004.

[5] A. Berger and V. O. Mittal. Query-relevant summarization using faqs. In *IN PROCEEDINGS OF THE 38TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, pages 294–301, 2000.

[6] D. Blei, L. Carin, and D. Dunson. Probabilistic topic models. *IEEE Signal Processing Magazine*, 27:55–65, 2010.

[7] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Neural Information Processing Systems*, 2004.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[9] G. Carenini, R. T. Ng, and X. Zhou. Summarizing emails with conversational cohesion and subjectivity. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 353–361, 2008.

[10] Y.-L. Chang and J.-T. Chien. Latent dirichlet learning for document summarization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1689–1692, 2009.

[11] D. Chen, J. Tang, L. Yao, J. Li, and L. Zhou. Query-focused summarization by combining topic model and affinity propagation. In *Advances in Data and Web Management*, volume 5446 of *Lecture Notes in Computer Science*, pages 174–185. Springer Berlin Heidelberg, 2009.

[12] W. T. Chuang and J. Yang. Extracting sentence segments for text summarization: a machine learning approach. In *Research and Development in Information Retrieval*, pages 152–159, 2000.

[13] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[14] H. Dong, S. C. Hui, and Y. He. Structural analysis of chat messages for topic detection. *Online Information Review*, pages 496–516, 2006.

[15] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 281–285. ACM, 1988.

[16] H. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285, 1969.

[17] M. A. Fattah and F. Ren. Automatic text summarization. *International Journal of Electrical and Computer Engineering*, pages 25–28, 2008.

[18] P.-E. Genest and G. Lapalme. Framework for abstractive summarization using text-to-text generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 64–73, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[19] D. Gillick, K. Riedhammer, B. Favre, and D. Z. Hakkani-Tur. A global optimization framework for meeting summarization. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 4769–4772, 2009.

[20] T. L. Griffiths. Finding scientific topics. *Proceedings of The National Academy of Sciences*, 101:5228–5235, 2004.

[21] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 362–370. Association for Computational Linguistics, 2009.

[22] M. A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23:33–64, 1997.

[23] T. Hofmann. Probabilistic latent semantic analysis. In *In Proceedings of Uncertainty in Artificial Intelligence, UAI99*, pages 289–296, 1999.

[24] J. Jagadeesh, P. Pingali, and V. Varma. A relevance-based language modeling approach to duc 2005. In *Document Understanding Conference*, 2005.

[25] H. Jing, R. Barzilay, K. Mckeown, and M. Elhadad. Summarization evaluation methods: Experiments and analysis. In *AAAI Symposium on Intelligent Summarization*, pages 60–68, 1998.

[26] K. S. Jones. Automatic summarising: factors and direction. In *Advances in automatic text summarisation*, pages 1–12. MIT Press, 1998.

[27] J. Kupiec, J. O. Pedersen, and F. Chen. A trainable document summarizer. In *Research and Development in Information Retrieval*, pages 68–73, 1995.

[28] J. Larocca, N. Alexandre, D. Santos, C. A. A. Kaestner, A. A. Freitas, and C. do Parana. Document clustering and text summarization. *Proc. of 4th Int. Conf. Practical Applications of Knowledge Discovery and Data Mining*, pages 41–55, 2000.

[29] L. Lee. Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 25–32, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics.

[30] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26, 2004.

[31] C.-Y. Lin and E. H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1 of *NAACL '03*, pages 71–78, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[32] W. Lowe and S. McDonald. The direct route: Mediated priming in semantic space. *Proceedings of the 22nd Conference of the Cognitive Science Society*, pages 806–811, 2000.

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

[34] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996.

[35] W. C. Mann and S. A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[36] D. Marcu. *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. PhD thesis, Department of Computer Science, University of Toronto, Toronto, Canada, 1997.

[37] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *International Joint Conference on Artificial Intelligence*, pages 786–791, 2005.

[38] G. A. Miller. Wordnet: a lexical database for english. *Communications of The ACM*, 38:39–41, 1995.

[39] J. D. Moore and F. Keller. Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. *Cognitive Science*, pages 685–690, 2006.

[40] S. Muresan, E. Tzoukermann, and J. L. Klavans. Combining linguistic and machine learning techniques for email summarization. In *Proceedings of the 2001 Workshop on CoNLL*, volume 7, pages 1–8, 2001.

[41] J. Quinlan. *C4.5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[42] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang. Mead - a platform for multidocument multilingual text summarization. In *LREC*, 2004.

[43] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, volume 4 of *NAACL-ANLP-AutoSum '00*, pages 21–30, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[44] O. Rambow, L. Shrestha, J. Chen, and C. Lauridsen. Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 105–108, 2004.

[45] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.

[46] O. Sandu. *Domain Adaptation for Summarizing Conversations*. PhD thesis, Department of Computer Science, The University Of British Columbia, Vancouver, Canada, 2011.

[47] D. Song and P. Bruza. Discovering information flow suing high dimensional conceptual space. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 327–333, New York, NY, USA, 2001. ACM.

[48] M. Steyvers and T. Griffiths. *Probabilistic Topic Models*, volume 427, chapter Handbook of Latent Semantic Analysis. Lawrence Erlbaum Associates, 2007.

[49] J. Tang, L. Yao, and D. Chen. Multi-topic based query-oriented summarization. In *SIAM International Conference on Data Mining*, pages 1147–1158, 2009.

[50] J. Ulrich, G. Murray, and G. Carenini. A publicly available annotated corpus for supervised email summarization. In *AAAI08 EMAIL Workshop*, Chicago, USA, 2008. AAAI.

[51] D. C. Uthus and D. W. Aha. Plans toward automated chat summarization. In *Meeting of the Association for Computational Linguistics*, pages 1–7, 2011.

[52] H. M. Wallach, D. M. Mimno, and A. McCallum. Rethinking lda: Why priors matter. In *NIPS*, pages 1973–1981, 2009.

[53] D. Wang, S. Zhu, T. Li, and Y. Gong. Multi-document summarization using sentence-based topic models. In *Meeting of the Association for Computational Linguistics*, pages 297–300, 2009.

[54] X. Wei and B. W. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA, 2006. ACM.

[55] C. Whitelaw, B. Hutchinson, G. Chung, and G. Ellis. Using the web for language independent spellchecking and autocorrection. In *Empirical Methods in Natural Language Processing*, pages 890–899, 2009.

[56] S. Wunsch-Vincent and G. Vickery. Participative web: User-created content. Technical report, By the Directorate for Science, Technology and Industry Committee for Information, Computer and Communications Policy. Working Party on the Information Economy, 2006.

[57] L. Zhou and E. H. Hovy. Digesting virtual geek culture: The summarization of technical internet relay chats. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 298–305, 2005.