The Pennsylvania State University
The Graduate School

ANALYZING SUBJECTIVITY AND SENTIMENT OF ONLINE FORUMS

A Dissertation in
Information Sciences and Technology
by
Prakhar Biyani

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2014

The thesis of Prakhar Biyani was reviewed and approved* by the following:

Prasenjit Mitra
Associate Professor, Information Sciences and Technology
Dissertation Advisor, Chair of Committee

John Yen
Professor, Information Sciences and Technology

Alexander Klippel
Assistant Professor, Geographical Information Science

Marcel Salathe
Assistant Professor, Biology

Cornelia Caragea
Assistant Professor, Computer Science
Special Member

Peter Forster
Assistant Dean for Online Programs and Professional Education, Information Sciences and Technology

*Signatures are on file in the Graduate School.

Abstract

Online social media has emerged as a popular medium for seeking and providing information, opinions and social support. Online sites such as discussion forums, blogs and health communities have tremendous amounts of user generated data in their archives. Analyzing this content for its subjectivity and sentiment has important applications such as improving information search in social media, understanding users for providing content personalization, identifying influential members in online communities, etc. In this dissertation, I will discuss my works on subjectivity analysis of online forum threads, identifying the type of social support (emotional or informational) present in and analyzing sentiment of user messages in an online health community (OHC). For subjectivity analysis, I show that thread-specific non-lexical features such as thread structure and dialogue acts expressed in thread posts are highly informative for inferring thread subjectivity. For sentiment analysis of messages of the OHC, I use unlabeled messages to augment a small training data using co-training and build highly accurate sentiment classifiers. For support identification, I build supervised classifiers using several generic and novel domain-specific features and analyze the posting behaviors of regular members and influential members in the OHC in terms of the type of support they provide in their messages. I find that influential members generally provide more emotional support as compared to regular members in the OHC. Experimental results demonstrate that all the proposed models significantly outperform various state-of-the-art models.

Table of Contents

List of Figures

List of Tables

Dedication

I dedicate this dissertation to my parents who constantly supported me during this wonderful journey of my PhD.

Chapter 1

Introduction

Owing to their widespread use, online forums contain enormous amount of data concerning wide range of domains (e.g., health, software, hardware, travel) in the form of discussions and question-answering between forum users. This data encompasses interesting information ranging from viewpoints, opinions, and sentiments to precise factual information provided by members of these sites. Analyzing such data has many important applications including information management for building smart information retrieval systems, understanding user perspectives/opinions pertaining to different topics, identifying leaders in online communities, finding language patterns commonly used to express different moods/emotions by people, etc. This dissertation specifically focuses on analyzing subjectivity and sentiment of the content posted in online forums. I address the following three problems:

1.1 Subjectivity Analysis of Online Forum Threads and its Use in Thread Retrieval

Subjectivity analysis essentially deals with separating factual information and opinionated information. It has been actively used in various applications such as opinion mining of customer reviews in online review sites, improving answering of opinion questions in community question-answering (CQA) sites, multi-document summarization, etc. However, there has not been much focus on subjectivity anal-

ysis in the domain of online forums. Online forums contain huge amounts of user-generated data in the form of discussions between forum members on specific topics and are a valuable source of information. Subjectivity analysis of online forum threads has many important applications including improving search in online forums. I perform subjectivity analysis of online forum threads. I model the task as a binary classification of threads in one of the two classes: subjective (seeking opinions, emotions, and other private states) and non-subjective (seeking factual information). Unlike previous works on subjectivity analysis, I use several non-lexical thread-specific features for identifying subjectivity orientation of threads. I evaluate the proposed methods by comparing them with several state-of-the-art subjectivity analysis techniques. Experimental results on two popular online forums demonstrate that the proposed methods outperform strong baselines. Next, I combine the subjectivity analysis model with a state-of-the-art thread retrieval model to improve thread retrieval. I match the intent of user queries with the type of information in a thread, in addition to the lexical match between the two, to enhance retrieval performance.

1.2 Sentiment Analysis of an Online Cancer Support Community Using Co-training

People share their health concerns on Online Health Communities and obtain social support during difficult phases of their lives when they or their loved ones suffer from serious diseases. Sentiment classification in such communities has not received much attention as compared to other domains such as product reviews, online forums, etc. However, identifying sentiments expressed by members in an online health community can be helpful in understanding the community and its features, e.g., dominant health issues, emotional impacts of interactions on members, finding influential members, etc. I perform sentiment classification of user posts in CSN. I use a small amount of labeled data to provide initial supervision to my model and then use a semi-supervised machine-learning algorithm, co-training, which uses information contained in unlabeled data to perform sentiment classification. I use domain-dependent and domain-independent features as the two

views of user posts (as required in co-training). Using the dataset from a popular OHC, the Cancer Survivor Network of the American Cancer Society, I demonstrate that using the unlabeled data improves sentiment classification performance significantly. The proposed algorithm is highly accurate (achieving up to 85% accuracy) and outperforms strong baselines by 6%-20%.

1.3 Identifying Emotional and Informational Support in Online Health Communities

A large number of online health communities exist today helping millions of people with social support during difficult phases of their lives when they suffer from serious diseases. Interactions between members in these communities contain discussions on practical problems faced by people during their illness such as depression, side-effects of medications, etc and answers to those problems provided by other members. Analyzing these interactions can be helpful in getting crucial information about the community such as dominant health issues, identifying sentimental effects of interactions on individual members, identifying influential members, etc. In this thesis, I analyze user messages of an online cancer support community, Cancer Survivors Network (CSN), to identity the two types of social support present in them: *emotional* support and *informational* support. I model the task as a binary classification problem. I use several generic and novel domain-specific features. Experimental results demonstrate high classification accuracy achieved by the proposed method. I, then, use the classifier to predict the type of support in CSN messages and analyze the posting behaviors of regular members and influential members in CSN in terms of the type of support they provide in their messages. I find that influential members generally provide more emotional support as compared to regular members in CSN.

Chapter 2

Subjectivity Analysis of Online Forum Threads and its Use in Thread Retrieval

2.1 Introduction

A large number of online forums in various domains (e.g., health, sports, travel, camera, laptops, etc.) exists today, containing huge volumes of user-generated data in the form of discussions between members. The topics discussed in the threads of these forums are very unique in nature as they are often related to practical aspects of life (e.g., *How much to tip after bad service?*). Since such information is not available in other webpages, online forums are increasingly becoming very popular among internet users for discussing real life problems.

In this work, I *analyze subjectivity of online forum threads*. I identify two types of threads in an online forum: *subjective* and *non-subjective* and model the subjectivity analysis task as a binary classification problem. Subjective threads discuss subjective topics that seek opinions, viewpoints, evaluations, and other private states of people, whereas non-subjective threads discuss non-subjective topics that seek factual information. Figure 2.1 shows a subjective thread from an online forum, Trip–Advisor New York. Figure 2.2 shows a non-subjective thread from the same forum. In the former, the topic of discussion is *whether to tip*

or not after bad service?, which seeks opinions, whereas the latter seeks factual information about *bands/artists playing in December in Madison Square Gardens.*

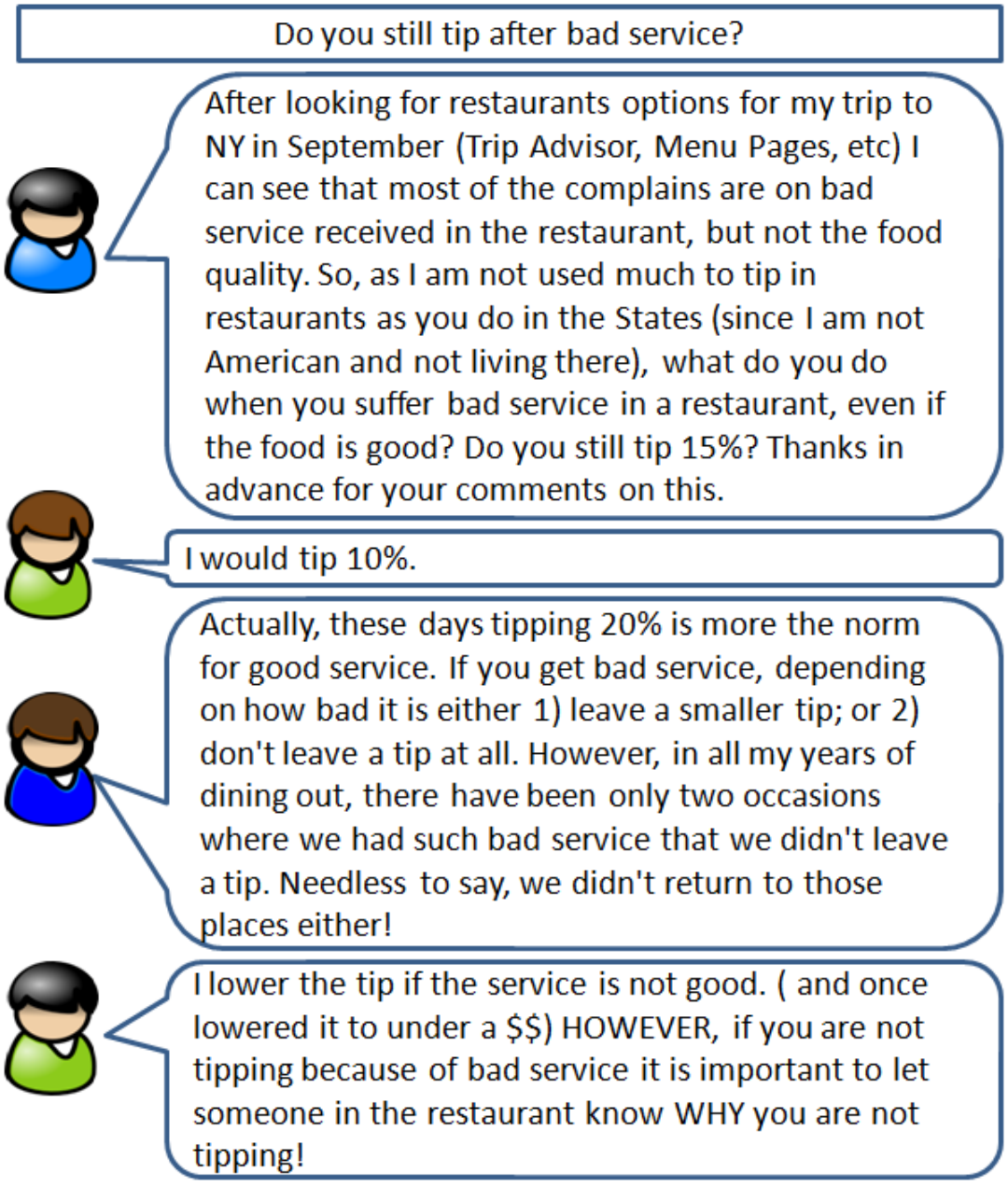

Figure 2.1: An example thread with subjective topic.

Figure 2.2: An example thread with non-subjective topic.

2.1.1 Why Subjectivity Analysis of Online Forum Threads?

- ***Improving search in online forums***: Internet users search online forums, generally, for two types of information. Some of them search the forums for subjective information such as different viewpoints, opinions, emotions, evaluations, etc., on specific problems instead of a single correct answer. Other users want short factual (objective) answers. Previous works on online forum search have focused on improving the lexical match between searcher's query keywords and thread content [1, 2, 3]. However, these works do not take into account a searcher's intent, i.e., the *type of* information a searcher wants. Let us consider the following two example queries issued by a searcher to some camera forum: 1) How is the resolution of Canon 7D, 2) What is the resolution of Canon 7D. The two queries look similar, but they differ in their intents. In the first query, the searcher wants to know what other camera users think about the resolution of the Canon 7D, how are their experiences (good, bad, okay, excellent, etc.) with the camera as far as its resolution is concerned and other such types of *subjective* information. The second query, however, is *objective* in nature in which the searcher wants a factual answer, which, in this case, is the value of the resolution of the camera. Hence, queries having similar keywords may differ in their intents. Search algorithms based only on keyword search would perform badly for

these types of queries. I believe that by knowing the type of information (subjective or objective) contained in a forum thread, these types of queries can be addressed in a better way. A forum search engine can then match the searcher's intent with the type of information a thread contains in addition to the keyword match between the two and thus, handle the queries more intelligently.

- ***Abuse detection***: Online forums are informal in nature. Often, discussions in threads get heated with users getting engaged in abusive conversations. Forum administrators continuously monitor forums for such contents and remove them as they are against the community rules. These conversations are subjective in nature and hence can potentially be detected by analyzing threads for subjectivity.

Previous works on subjectivity classification have extensively used lexical features such as bag-of-words, n-grams, combinations of n-grams and parts of speech tags, etc [4, 5, 6]. A major issue with these features is their high dimensionality feature space and hence there is a risk of model overfitting especially with small training data. In this work, I explore the possibility of using non-lexical and thread specific features for the subjectivity classification of threads. Specifically, I explore the following research question: *Can non-lexical thread specific features (e.g., number of users in a thread, number of posts in a thread, etc.) help in inferring the subjectivity of online forum threads?* To address the question, I propose and evaluate several thread specific features for subjectivity classification. I compare the performance of my classification model with various state-of-the-art techniques and show that the model outperforms the baselines in most of the cases.

2.1.2 Contributions

The work has the following contributions:

1. The present work is the first to perform subjectivity analysis of online forum threads and use it in improving thread retrieval.

2. I propose two new types of non-lexical features for subjectivity analysis of online forum threads: *structural* features and *dialogue act* features. Previous

works on subjectivity analysis have mainly used lexical and syntactic features like n-grams, POS tags, subjectivity clues, etc. I empirically show that, for online forum threads, in addition to the traditionally used features, thread's structure and information about dialogue acts expressed in its posts also help in analyzing its subjectivity.

3. I incorporate the subjectivity label information of threads in a state-of-the-art forum thread retrieval model and show that subjectivity information improves retrieval performance as measured by different metrics.

2.2 Related Work

Subjectivity analysis has been an active field of research due to its important applications in opinion mining, sentiment analysis, question-answering, summarization, etc. Here, I, first, provide a brief survey of works on subjectivity analysis in general and then review the works that performed subjectivity analysis in different domains (online review sites, community answers, etc.) and used it in different applications (opinion mining, question-answering, etc.).

2.2.1 Subjectivity Analysis

Wiebe et al. [7] did a seminal work on generating and using a gold standard dataset for subjectivity classification. They performed subjectivity classification of sentences using basic features such as presence of a pronoun, an adjective, a modal, etc. in the sentence. Bruce et al. [8] performed a case study of manual subjectivity tagging. Wiebe and Riloff [9] performed subjectivity classification of sentences in World Press articles using unannotated data. They used high precision rule-based classifiers for generating an initial training data and then used semi-supervised learning to iteratively learn subjectivity patterns and augment the training data. Su and Markert [10] performed word sense subjectivity classification using the training data generated from the existing opinion mining resources and showed that the performance is comparable with that of the classifier trained on a dedicated training set. Other works have performed subjectivity classification across different languages [11, 12]. They discussed and evaluated methods

to develop subjectivity analysis tools for selected languages by applying machine translation on the available subjectivity analysis tools and resources for English language. Banea et al. [13] performed subjectivity classification in six different languages and showed that including multilingual information improves the classification performance across all the languages. Mukund and Srihari [14] proposed a vector-space classification algorithm boosted by co-training for subjectivity classification of sentences in Urdu Language.

2.2.2 Opinion Mining

An integral part of opinion mining and sentiment analysis is to separate subjective sentences from objective ones and then to identify the polarity (negative, neutral or positive) of the opinions expressed in the subjective sentences [15]. Works in this area have mainly focused on online review sites for summarizing product reviews given by different users of those products [16, 17]. This work, in contrast, deals with online forum threads. A review in a review site is a continuous piece of text written by a person with additional information such as ratings, date and time. On the other hand, a thread in an online forum has a distinctive structure due to the presence of messages posted by multiple users. Also, a review, usually, has a single role of providing user's feedback on a product whereas posts in a thread have multiple roles, e.g., a post can be a question, solution, feedback, junk, etc [18]. These differences make subjectivity analysis of online forum threads different from that in review sites in both nature and the approaches that can be used for the analysis. For example, thread structure, role of posts and other thread-specific information can be used as features for subjectivity analysis (as will be described later in the chapter).

2.2.3 Question-Answering

Subjectivity analysis has also been used to improve question-answering in online communities and social media [19, 20, 21, 4, 22]. Yu et al. [4] classified documents and sentences from news data into facts and opinions with the aim of improving answering of complex opinion questions. Stoyanov et al., [21] used subjectivity filter on answers, separating factual sentences from opinion sentences, to improve

answering of opinion questions. Somasundaran et al., [22] identified different types of attitudes in questions and answers and then use it to improve opinion question answering on web-based discussions and news data by matching the attitude types of questions and answers. Li et al.[5] classify questions in Yahoo QA as subjective or objective using semi-supervised learning by utilizing the text of labeled questions and their unlabeled answers for learning subjectivity patterns. Li et al., [23] used graphical models to rank answers based on their topical and sentiment relevance to opinion questions. Gurevych et al. [20] used an unsupervised lexicon based approach to classify questions as subjective or factoid (non-subjective). They manually build a lexicon of subjective words and word patterns from annotated questions and classify test questions based on a score calculated using the number of patterns present in them. Moghaddam et al., [24] performed aspect-based question answering in product reviews and showed that taking into account the match between opinion polarities of questions and answers improved answer retrieval. Oh et al., [25] improved answering of non-factoid why-questions by using supervised classification for re-ranking answers based on their sentiment and other properties. All these previous works focused on improving question-answering of non-factoid (i.e., opinion) questions in product reviews and community QA sites. In contrast, the current work employs subjectivity analysis to improve an ad-hoc vertical retrieval model for an online forum. I show that using the subjectivity match, retrieval performance can be improved for both subjective and non-subjective queries.

2.2.4 Online Forums

In the domain of online forums, there have been two recent works that are close to the current work. Hassan et al. [26] performed sentence-level attitude classification in online discussions to model user interaction that may be helpful in facilitating collaborations. Zhai et al. [27] classified sentences in online discussions as evaluative or non-evaluative for getting relevant opinion sentences. In contrast, this work does thread-level subjectivity classification as I am interested in knowing the subjectivity of the overall topic of discussion of a thread and plan to use it for improving online forum search in the future. There have been works analyzing dialogic structure of posts in online debates to find on which side of the debate (FOR

or AGAINST) the posts are [28] and identify disagreements between posts [29]. However, the current work is very different from these works. I identify eight types of dialogue acts expressed in a thread posts and use them to infer subjectivity of the thread's topic.

2.3 Subjectivity Classification

In this section, I formulate the subjectivity classification problem and describe various features used in the classification task.

2.3.1 Problem Formulation

An online forum thread discusses a topic specified by thread starter in the title and the initial post. The topics of discussion in the threads can either be subjective or non-subjective (See Figures 2.1 and 2.2 for examples of subjective and non-subjective threads, respectively). Based on the definitions of subjective and objective sentences given by [8], I define a subjective topic of discussion as a topic that seeks people's opinions, viewpoints, evaluations, speculations, and other private states and a non-subjective topic as a topic that seeks factual information. I call a thread subjective if its topic of discussion is subjective and non-subjective if it discusses a non-subjective topic. I assume that in online forum threads subjective topics have discussions in subjective language (i.e., expressing different private states) and non-subjective topics have discussions in objective language (i.e., expressing facts and verifiable information). We note that there may be some cases where the assumption does not hold good, however, analysis of such exceptional cases is not the focus of this chapter and is left for future work.

Problem statement: Given an online forum thread T, the task is to classify it into one of the two classes: *Subjective* (denoted by s) or *Non-Subjective* (denoted by ns).

In this work, I assume that a thread has a single topic of discussion which is specified by the thread starter in the title and the initial post. Analyzing subjectivity of threads with multiple topics is a separate research problem that is out of scope of this work.

2.3.2 Feature Engineering

As discussed before, I wanted to explore the effect of using various thread specific features for subjectivity analysis of online forum threads and compare them with the state-of-the-art subjectivity analysis techniques. In this section, I describe the features used and intuition behind using them. Table 2.1 lists the features used.

2.3.2.1 Structural Features

I posit that subjective threads have different structural properties than non-subjective threads. Since subjective topics have more scope of discussion, we expect the subjective threads to be longer and invoke more participation of users than non-subjective threads. I use the length of a thread and the participation of users in a thread as features. For the length, I use the length of the initial post, the length of the thread and the average of the length of all the reply posts in the thread as features. All the lengths are measured in terms of the number of words. For the participation, I use the number of users that participated in the given thread, the number of posts and the average number of posts by a user in a thread as features.

2.3.2.2 Dialogue Act Features

Online forum threads have conversational nature and hence there are different types of dialogue acts (question, solution, feedback, etc.) expressed in thread posts [18, 30, 31]. For example, a thread starts with a *question* posted by the thread starter. Then, there are posts (by other users) that ask for some *clarifying* details about the question and the thread starter provides *further details* to make the question clearer. After getting the details, users suggest *solutions* and finally there are *feedbacks* (by the thread starter or other users) to the suggested solutions that can be *positive* or *negative*. Also, there may be posts that ask the *same question* (as asked in previous posts) and posts that are *junk* and not related to thread discussion. I posit that dialogue acts expressed in the posts maybe helpful in identifying thread's subjectivity. In a subjective thread, there could be multiple solutions suggested for a question (e.g. *Sony or Nikon which is better?*) as there is no single correct answer to subjective questions and hence multiple

Feature Name	Description
	Structural Features
InitPostLength	Total number of words in the initial post.
ThreadLength	Total number of words in the thread.
NumPost	Total number of posts in the thread.
NumUser	Total number of users in the thread.
AvgPostAuthor	Average number of posts by a user in the thread.
AvgLengthPost	Average number of words in a post in the thread.
	Dialogue Act Features
numQues	No. of *question* posts in the thread.
numRepeat	No. of *repeat question* posts in the thread.
numClar	No. of *clarification* posts in the thread.
numDetails	No. of *further details* posts in the thread.
numSol	No. of *solution* posts in the thread.
numNegFB	No. of *negative feedback* posts in the thread.
numPosFB	No. of *positive feedback* posts in the thread.
numJunk	No. of *junk* posts in the thread
	Subjectivity Lexicon-based Features
NumSubjTitle	No. of subjectivity clues in the title of the thread.
NumSubjInit	No. of subjectivity clues in the initial post of the thread.
NumSubjReply	No. of subjectivity clues in all the reply posts of the thread.
	Sentiment Features
InitSentiAvgPos	Positive sentiment score of initial post based on all the indicative word patterns in it.
InitSentiAvgNeg	Negative sentiment score of initial post based on all the indicative word patterns in it.
InitSentiStrngPos	positive sentiment score of initial post based on the strongest indicative word patterns in it.
InitSentiStrngNeg	Negative sentiment score of initial post based on the strongest indicative word patterns in it.
ReplySentiAvgPos	Average of positive sentiment scores of all the reply posts based on all the indicative word patterns in them.
ReplySentiAvgNeg	Average of Negative sentiment scores of all the reply posts based on all the indicative word patterns in them.
ReplySentiStrngPos	Average of positive Sentiment scores of all the reply posts based on the strongest word patterns in them.
ReplySentiStrngNeg	Average of Negative Sentiment scores of all the reply posts based on the strongest word patterns in them.

Table 2.1: Description of various features used for subjectivity classification.

feedbacks would be given. In contrast, in non-subjective threads, since questions seek factual materials (e.g., *what do the numbers on camera lens mean?*), there is little scope of discussion or disagreement among solution providers and hence there would be less solutions suggested and less number of feedbacks. Also, in subjective threads, the discussions can get heated due to disagreements with users posting inappropriate content such as abuses which are *junk* as they are not related to the discusion whereas in non-subjective threads, these situations are unlikely to happen. To explore the impact of dialogue acts on a thread's subjectivity, I used

eight dialogue acts in thread posts as proposed by [18] and used their presence in a thread as features for the subjectivity classification. The dialogue acts are as follows: 1. Question, 2. Repeat Question, 3. Clarification, 4. Further Details, 5. Solution, 6. Negative Feedback, 7. Positive Feedback, 8. Junk. I implemented their classification model to identify the dialogue acts in thread posts. I designed 8 features corresponding to the 8 dialogue acts for a thread. Each feature represents the number of posts in a thread that belong to a given dialogue act class.

2.3.2.3 Subjectivity Lexicon Based Features

Subjective threads discuss subjective topics seeking private states such as opinions, emotions, evaluations, etc. whereas non-subjective threads seek factual information. This difference should result in differences in the vocabularies of these two types of threads. Subjective threads should contain words that are used to express subjectivity whereas non-subjective threads should either not have these words or have less number of these words. I call these words *subjectivity clues* in this work. Hence, the frequency or term counts of subjectivity clues in a thread should be a good indicator of its subjectivity. I use a publicly available subjectivity lexicon compiled from MPQA corpus by [32] to get the subjectivity clues. The lexicon contains 8221 subjectivity clues. Some of the examples of subjectivity clues from the lexicon are *abhor, abuse, bother, champion*. I count the number of subjectivity clues in the title, initial post and all reply posts of a thread, normalize the subjectivity clue counts with the number of words in the corresponding element (title, initial post, reply posts) and use them as features. For a thread, I computed three lexicon features: NumSubTitle, NumSubInit and NumSubReply. I calculated NumSubTitle and NumSubInit by normalizing the frequency counts of subjectivity clues in the title and the initial post, respectively, by their total number of words. For computing NumSubReply, I first calculated the normalized frequency counts of subjectivity clues for all the reply posts and then added all the normalized counts.

2.3.2.4 Sentiment Features

These features take into account the sentiment/emotion of a thread. We expect subjective threads to have posts with higher sentiments (as they expose private states) than the posts in non-subjective threads. To calculate sentiment features for a thread, I compute sentiment strength of its individual posts using the SentiStrength algorithm [33]. I use the implementation of the algorithm available at http://sentistrength.wlv.ac.uk/. The algorithm is developed specifically to compute sentiment strength scores for short informal pieces of text common in social media such as forum posts, blog comments, etc. SentiStrength calculates both positive as well as negative sentiment scores for a piece of text. This feature is desirable as the posts can express sentiments of multiple polarity and a single sentiment score (positive, negative or neutral) will not be able to capture the individual sentiments. For both polarities, the algorithm gives two types of scores for a piece of text (i) using the strongest sentiment-indicative word patterns and (ii) using all the sentiment-indicative word patterns and taking their average. Thus, we get four different sentiment strength scores for each post. I use the four sentiment strength scores for the initial post and averages of the four sentiment scores for all the reply posts as features, thus getting eight sentiment features for a thread (see Table 2.1).

2.4 Using Subjectivity Classification To Improve Thread Retrieval

As explained in Section 2.1, subjectivity analysis of online forum threads can potentially be helpful in improving information search in online forums. Here, I discuss how information about subjectivity of threads can be utilized in thread retrieval systems. I use a state-of-the-art probabilistic model for forum thread retrieval [2] as the baseline and incorporate subjectivity of threads in the model to see if it helps improve retrieval performance. Next, I formulate the retrieval problem and then discuss how to utilize the subjectivity information of threads and queries to improve thread retrieval.

2.4.1 Problem Formulation

Given a query Q and its subjectivity orientation (subjective or non-subjective), a corpus of n threads (T) and likelihood of all the threads being subjective, $P(Subj|T)$, generate a ranked list of threads $L = T_1, T_2, ...T_n$ such that for all $1 \leq i, j \leq n$ and $i < j$,, $rel(T_i, Q) \geq rel(T_j, Q))$, where $rel(T, Q)$ is the relevance score of thread T with respect to query Q. $P(Subj|T)$ is the probability of a thread being subjective and is calculated using subjectivity classifier which is explained next.

2.4.2 Probabilistic Retrieval

Bhatia and Mitra [2] used a probabilistic model based on inference networks that utilizes the structural properties of forum threads. Given a query Q, the model computes $P(T|Q)$, the probability of thread T being relevant to Q, as follows:

$$P(T|Q) \stackrel{rank}{=} P(T) \prod_{i=1}^{n} \left\{ \sum_{j=1}^{m} \alpha_j P(Q_i|S_{j_T}) \right\} \tag{2.1}$$

where:

$P(T)$ is the prior probability of a thread being relevant,

Q_i is the i^{th} term in query Q,

S_{jT} is the j^{th} structural unit in the thread T,

α_j determines the weight given to component j and $\sum_{j=1}^{m} \alpha_j = 1$.

Note that the term $\prod_{i=1}^{n} \left\{ \sum_{j=1}^{m} \alpha_j P(Q_i|S_{j_T}) \right\}$ models lexical match between query and thread content. In order to estimate the likelihoods $P(Q_i|S_{j,T})$, I use the standard language modeling approach in information retrieval [34] with *Dirichlet Smoothing* as follows:

$$P(Q_i|S_{j_T}) = \frac{f_{Q_i,j_T} + \mu \frac{f_{Q_i,j_C}}{|j|}}{|j_T| + \mu} \tag{2.2}$$

Here,

f_{Q_i,j_T} = frequency of term Q_i in j^{th} structural component of thread T,

f_{Q_i,j_C} = frequency of term Q_i in j^{th} structural component of all the threads in the collection C.

$|j_T|$ is the length of j^{th} structural component of thread T,
$|j|$ is the total length of j^{th} structural component of all the threads in the collection C,
μ is the Dirichlet smoothing parameter.

In this work, I set μ to be equal to 2000, a value that has been found to perform well empirically [35].

Thus, the model computes the overall probability of a thread being relevant to the query by combining evidences from different structural units of the thread. The model considers three different structural units of threads – thread title, thread's initial post and the set of follow-up reply posts.

2.4.3 Incorporating Subjectivity Information in the Retrieval Model

In absence of any information about thread's content, subjective threads are more likely to be relevant to subjective queries and vice versa for non-subjective threads. I conceptualize this idea by taking into account the match between subjectivities of threads and queries in addition to the lexical match between them. Specifically, I incorporate the subjectivity match using the term $P(T)$ (in Equation 2.1) which represents the prior probability of a thread being relevant to a query. I use the following two settings to incorporate subjectivity match between threads and queries into the retrieval model:

1. **Subjectivity probability of a thread as its prior relevance probability**: For subjective (or non-subjective) queries, a thread's prior probability of being relevant is taken to be its probability of being subjective (or non-subjective). More precisely, for a subjective query, Q_s, relevance score of a thread T is calculated as follows:

$$P(T|Q_s) \stackrel{rank}{=} P(Subj|T) \prod_{i=1}^{n} \left\{ \sum_{j=1}^{m} \alpha_j P(Q_{si}|S_{j_T}) \right\} \tag{2.3}$$

   Here, $P(Subj|T)$ is the probability of thread T being subjective as outputted by the subjectivity classifier. Likewise, for a non-subjective query, the term

$P(Subj|T)$ is replaced by $P(NSubj|T)$ which is the probability of thread T being non-subjective. For a thread T, $P(Subj|T) + P(NSubj|T) = 1$.

2. **Re-ranking using subjectivity probability**: A two-step ranking model is used. First, threads are ranked according to their lexical similarity with the query where $P(T)$ is taken as constant for all the threads and then re-ranking of threads (at various ranks) is performed based on their subjectivity probabilities. Basically, for a subjective query, re-ranking is sorting (in descending order) the ranked list of threads based on their subjectivity probabilities. Re-ranking for a non-subjective query is done similarly.

3. **Combining subjectivity match and lexical match**: For a query and a thread, relevance score is calculated by taking convex combination of their lexical match and subjectivity match. Mathematically for a subjective query Q_s, ranking of threads is done using the relevance score defined as follows:

$$P(T|Q_s) \stackrel{rank}{=} \lambda.P(Subj|T) + (1-\lambda).\prod_{i=1}^{n}\left\{\sum_{j=1}^{m}\alpha_j P(Q_{si}|S_{j_T})\right\} \tag{2.4}$$

Here, λ_s is a parameter used to assign weight to the lexical match and subjectivity match for subjective queries. For non-subjective queries, $P(Subj|T)$ is replaced by $P(NSubj|T)$ and λ_s by λ_{ns}.

2.4.4 Getting Subjectivity Information for Threads and Queries

To obtain subjectivity probability for a thread ($P(Subj|T)$), I use the subjectivity classifier. I used the trained classifier to get confidence for all the threads (of belonging to the subjective class) in the forum dataset. I used the confidence scores as the subjectivity probabilities. For determining query subjectivity, I took help of human annotators (discussed in Section 2.5.1). Figure 2.3 shows the schematic of the retrieval model.

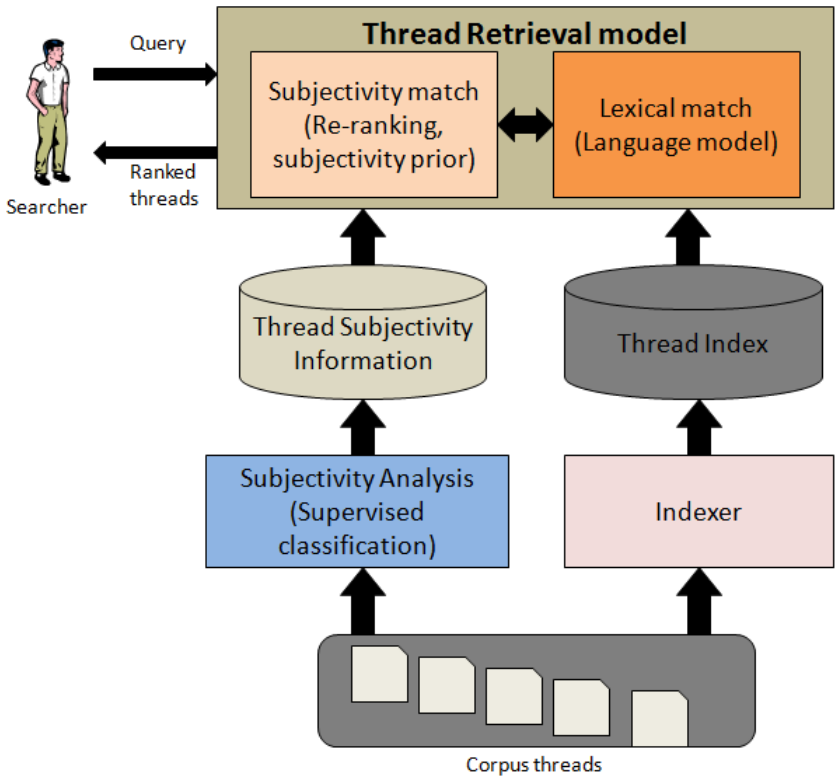

Figure 2.3: Schematic of the retrieval model.

2.5 Experiments

2.5.1 Data

To conduct experiments, I used a the dataset as used by Bhatia and Mitra [2]. The dataset consists of threads crawled from two popular online forums: 1. **Trip Advisor–New York** that contains travel related discussions mainly for New York city [1] and 2. **Ubuntu Forums** that contains discussions related to the Ubuntu operating system [2]. It has 83072 and 113277 crawled threads from the Trip Advisor–New York and Ubuntu forum respectively, a set of 25 queries and associated relevance judgments for both the datasets. For a query, the dataset has graded relevance judgments: 0 for totally irrelevant threads, 1 for partially relevant threads and 2 for highly relevant threads. To conduct subjectivity classification experiments, I randomly sampled 700 threads from both the datasets. Table 2.2

[1] http://www.tripadvisor.com/ShowForum-g60763-i5-New_York_City_New_York.html
[2] http://ubuntuforums.org

provides various statistics of the tagged data.

Statistic	**Trip–Advisor**	**Ubuntu**
Total # threads	609	621
Total # posts	6591	3603
Total # users	1206	1786
Average thread length (in terms of # posts)	10.82	5.80
Average thread length (in terms of # words)	907	387.57
Average # users in a thread	1.98	3.41

Table 2.2: Statistics of the tagged dataset.

I hired two human annotators for tagging the threads. The annotators were asked to tag a thread as subjective if its topic of discussion is subjective or non-subjective if the topic of discussion is non-subjective. The annotators were provided with a set of instructions for annotations. The set contained definitions of subjective and non-subjective topics with examples and guidelines for doing annotations. The annotations for each dataset were conducted in three stages. First, the annotators were asked to annotate a sample of 20 threads from the dataset using the instruction set. Second, separate discussions were held between the authors and each annotator. Each annotator was asked to provide his arguments (for his annotations) and, in case of inconsistent arguments, he was educated through discussions. Next, he was given the full dataset for annotation.

The overall percentage agreement between the annotators and Kappa value for the Trip Advisor dataset were 87% and 0.713 respectively and for the Ubuntu dataset were 88.7% and 0.732 respectively, indicating substantial agreement between the taggers in both the cases. For experiments, I used the data on which the annotators agreed. There were 412 subjective and 197 non-subjective threads in Trip Advisor dataset and 231 subjective and 390 non-subjective threads in Ubuntu dataset. The tagged dataset can be downloaded from the authors' website. [3].

For retrieval experiments, I used the Trip Advisor dataset. Three human annotators were asked to annotate queries in the dataset as subjective or non-subjective. First, two annotators tagged all the 25 queries getting agreement on 22 queries. The third annotator was then asked to disambiguate the tags of the three queries on

[3]http://www.personal.psu.edu/pxb5080/dataSubj.html

which the two annotators disagreed. There are 10 subjective and 15 non-subjective queries. Table 2.3 lists some of the queries. Queries in bold are the ones on which the annotators disagreed.

Subjective	**Non-Subjective**
hotel rates in Manhattan	new york to niagara falls
chinese food in brooklyn	educational trips in new york
best thanksgiving turkey	beaches in new york city
how safe is new york	winter temperature in new york city
vegetarian restaurants in manhattan	penn station to JFK

Table 2.3: Examples of subjective and non-subjective queries.

2.5.2 Baseline for Subjectivity Classification

Lexical features such as n-grams and parts-of-speech tags have been shown to perform well for subjectivity analysis tasks. Many works have used these features for subjectivity classification [5, 4, 6]. In this work, I use classifiers built on these features as baselines. I used the *Lingua-en-tagger* package from CPAN[4] for part-of-speech tagging. The extracted features and their description is given in Table 2.4. The table describes feature generation on a sentence containing three words W_i, W_{i+1} and W_{i+2}. POS_i, POS_{i+1} and POS_{i+2} are the parts-of-speech tags for the words W_i, W_{i+1} and W_{i+2}, respectively. For feature representation, I used term frequency as the weighting scheme (I empirically found it to be more effective than *tf-idf* and *binary* representations), and used minimum document frequency for a term to be included in the vocabulary as 3 (I experimented with minimum document frequency 3, 5 and 10 and 3 gave the best results).

I extracted the above features (Table 2.4) from the textual content of different structural units (title, initial post, reply posts) of the threads. First, I built a basic model where I used only the text of the titles (denoted by t) for classification. Then,

[4]http://search.cpan.org/dist/Lingua-EN-Tagger/Tagger.pm

Feature type	**Generated feature**
Uni	W_i, W_{i+1}, W_{i+2}
Uni+Bi	$W_i, W_{i+1}, W_{i+1}, W_iW_{i+1}$, $W_{i+1}W_{i+2}$
Uni+Bi+Tri	$W_i, W_{i+1}, W_{i+1}, W_iW_{i+1}, W_{i+1}W_{i+2}$, $W_iW_{i+1}W_{i+2}$
Uni+POS	$W_i, POS_i, W_{i+1}, POS_{i+1}, W_{i+2}, POS_{i+2}$
Uni+Bi+POS	$W_i, POS_i, W_{i+1}, POS_{i+1}, W_{i+2}, POS_{i+2}, W_iW_{i+1}, W_iPOS_{i+1}$, $POS_iW_{i+1}, W_{i+1}W_{i+2}, W_{i+1}POS_{i+2}, POS_{i+1}W_{i+2}$
Uni+Bi+Tri+POS	$W_i, POS_i, W_{i+1}, POS_{i+1}, W_{i+2}, POS_{i+2}, W_iW_{i+1}, W_iPOS_{i+1}$, $POS_iW_{i+1}, W_{i+1}W_{i+2}, W_{i+1}POS_{i+2}, POS_{i+1}W_{i+2}, W_iW_{i+1}W_{i+2}$, $W_iW_{i+1}POS_{i+2}, W_iPOS_{i+1}W_{i+2}, POS_iW_{i+1}W_{i+2}$, $W_iPOS_{i+1}POS_{i+2}, POS_iW_{i+1}POS_{i+2}, POS_i, POS_{i+1}W_{i+2}$

Table 2.4: Feature Generation for sentence W_i W_{i+1} W_{i+2}. Uni, Bi, Tri and POS denote unigrams, bigrams, trigrams and parts-of-speech tags respectively.

I used the text of initial posts and reply posts. I experimented with the following four settings: title (t), initial post (I), title and initial post (t+I), entire thread (t+I+R).

2.5.3 Experimental Setting

I used various supervised learning algorithms to perform the classification experiments. I experimented with Multinomial NaiveBayes, Support Vector Machines, Logistic regression, Bagging, Boosting and Decision Trees. Logistic regression gave the best results with the proposed features whereas in case of the baseline lexical features, Multinomial NaiveBayes outperformed all the other classifiers. I used Weka data mining toolkit with default settings to conduct the experiments [36]. To evaluate the performance of classifiers, I used macro-averaged precision, recall and F-1 measure. For a metric M, macro-average M_{mav} is calculated by taking weighted average of M for both subjective and non-subjective classes for each fold and then taking mean of the weighted averages across all the folds. For n-fold cross validation, M_{mav} is mathematically defined as follows:

$$M_{mav} = \frac{1}{n}\sum_{i=1}^{n}\frac{n_{s_i}M_{s_i} + n_{ns_i}M_{ns_i}}{n_{s_i} + n_{ns_i}} \tag{2.5}$$

where n_{s_i} and n_{ns_i} are the number of subjective and non-subjective threads in the test set in the i^{th} fold . M_{s_i} and M_{ns_i} are the values of metric M for the subjective and the non-subjective classes, respectively, in the i^{th} fold . I used $n = 10$. I use F-1 measure to compare performances of two classifiers. A naive baseline that classifies all the threads in the majority class will have a macro-averaged precision, recall and F-1 measure of 0.457, 0.676 and 0.545 respectively for Trip–Advisor and 0.394, 0.628 and 0.485 respectively for Ubuntu. I consider these values to be the lower bounds for any the proposed methods.

To conduct retrieval experiments, I used the Indri language modeling toolkit[5]. While indexing, stemming was performed using Porter's stemmer [37] and stopwords were removed using a general stop word list of 429 words used in the Onix Test Retrieval Toolkit [6]. The queries and relevance judgments available with the dataset as discussed in Section 2.5.1 were used for retrieval experiments. For the baseline retrieval model, I used the optimal parameter settings as used in the original work [2]. In order to compare the performance of various retrieval methods, I report precision, Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP) at ranks 5, 10 and 15 [38].To evaluate the retrieval models, I compute the following metrics for a query [38]:

1. **Precision:** Precision at rank k ($P@k$) is defined as percentage of relevant documents in top k retrieved documents.

2. **Normalized discounted cumulative gain (NDCG):** NDCG at rank k ($NDCG@k$) is defined as $DCG@k$ divided by $IDCG@k$ where $DCG@k$ is discounted cumulative gain and $IDCG@k$ is ideal discounted cumulative gain. $DCG@k$ is calculated as:

$$DCG@k = \sum_{i=1}^{k} \frac{rel(i)}{\log_2(i+1)} \tag{2.6}$$

   Here, $rel(i) \in \{0, 1, 2\}$ is the relevance score of document at rank i. $IDCG@k$ is the maximum $DCG@k$ produced by sorting the retrieved documents (till rank k) by their relevance score.

[5] http://lemurproject.org
[6] http://www.lextek.com/manuals/onix/stopwords1.html

3. **Average precision (AP):** AP at rank k (AP@k) is defined as:

$$AP@k = \frac{\sum_{i=1}^{k} P(i) \times rel(i)}{No.\ of\ relevant\ documents} \tag{2.7}$$

   Here, $P(i)$ is precision at rank i.

4. **Reciprocal rank (RR):** RR is defined as the multiplicative inverse of the rank of the first relevant document.

For comparing performances of different methods, I take mean of these metrics for all the queries. Mean of AP and RR are called MAP and MRR respectively. For precision, NDCG and MAP, I use ranks (k) 5 and 10. For the model using combination of lexical and subjectivity match (denoted as lexSubjCombine), I use $\lambda_s = 0.25$ and $\lambda_{ns} = 0.08$.

2.5.4 Classification Results

2.5.4.1 Baseline Results

Table 2.5 reports the results of the subjectivity classification obtained from different baselines. A total of 24 experiments (using the six types of features for the four settings (t, I, t+I, t+I+R)) were conducted for both the datasets. From the table, we note that titles give fair estimate of thread's subjectivity and initial posts (I) provide a better estimate. Incorporating text from initial post and title (t+I) improves the performance slightly over the initial post (I) setting. Further, adding the text of reply posts (t+I+R) gives the best performance. This is expected as titles only contain some keywords related to the discussion topic whereas initial posts contain the entire problem of discussion and reply posts constitute a major portion of the discussion in the thread. We also note that unigrams+bigrams+POS and unigrams+bigrams consistently perform better than the other features for all the settings except for the title (t) setting where unigrams and unigrams+POS performed the best.

Trip–Advisor												
	t			I			t+I			t+I+R		
	Pr.	Re.	F-1	Pr.	Re.	F-1	Pr.	Re.	F-1	Pr.	Re.	F-1
U	0.618	0.644	0.625	0.662	0.665	0.664	0.671	0.673	0.672	0.703	0.716	0.706
U+B	0.56	0.586	0.565	0.713	0.718	0.715	0.700	0.704	0.702	0.738	0.747	**0.723**
U+B+T	0.627	0.55	0.564	0.703	0.658	0.669	0.697	0.655	0.666	0.721	0.732	**0.723**
U+POS	0.56	0.586	0.565	0.669	0.673	0.671	0.686	0.69	0.688	0.701	0.713	0.704
U+B+POS	0.606	0.616	0.610	0.704	0.711	0.704	0.701	0.709	0.704	0.733	0.741	0.71
U+B+T+POS	0.614	0.522	0.566	0.709	0.67	0.68	0.706	0.675	0.684	0.722	0.736	0.716
Ubuntu												
	t			I			t+I			t+I+R		
	Pr.	Re.	F-1	Pr.	Re.	F-1	Pr.	Re.	F-1	Pr.	Re.	F-1
U	0.546	0.578	0.553	0.652	0.646	0.648	0.649	0.643	0.645	0.694	0.689	**0.691**
U+B	0.551	0.58	0.557	0.662	0.655	0.658	0.659	0.654	0.656	0.688	0.67	0.675
U+B+T	0.548	0.576	0.554	0.656	0.646	0.649	0.657	0.647	0.651	0.696	0.663	0.669
U+POS	0.626	0.647	0.633	0.644	0.638	0.64	0.649	0.641	0.644	0.694	0.688	**0.69**
U+B+POS	0.552	0.564	0.556	0.659	0.652	0.655	0.659	0.652	0.655	0.72	0.696	**0.701**
U+B+T+POS	0.551	0.557	0.554	0.646	0.631	0.636	0.64	0.63	0.633	0.705	0.657	0.662

Table 2.5: Classification performance of different baseline features (Table 2.4) extracted from different structural components of the forum threads. t, I and R are title, initial post and set of all reply posts of a thread respectively. U, B, T and POS are unigrams, bigrams, trigrams and parts-of-speech tags respectively.

2.5.4.2 Performance of the Proposed Classification Model

Table 2.6 reports the results of the proposed classification model. I achieve an overall accuracy of 77.01%, a precision of 0.763 and an F-1 measure of 0.764 on the Trip–Advisor dataset and an overall accuracy of 70.05%, a precision of 0.692 and an F-1 measure of 0.684 on the Ubuntu dataset. I further analyze the classification performance for the individual classes. Table 2.7 reports precision, recall and F-1 measure for subjective and non-subjective classes for both the datasets. We observe that the classification performance for the subjective class is better than the non-subjective class for the Trip–Advisor dataset. This can be attributed to the significantly more number of subjective threads than non-subjective threads (refer to Section 2.5.1) in the Trip–Advisor dataset and hence more patterns for the classifier to learn for the majority (subjective) class leading to the better performance for that class. Similarly, for the Ubuntu dataset, we see a better performance for the non-subjective class whose number of threads are significantly more than that of the subjective class.

Next, I compare the performance of the proposed classification model with the baselines. As can be seen from Table 3.4, the classification model outperforms the best performing baseline (U+B for the t+I+R setting, refer to Table 2.5),

thus outperforming all the 24 baselines, for the Trip–Advisor dataset. For the Ubuntu dataset, the proposed model achieves an F-1 measure of 0.684 whereas the best performing baseline (U+B+POS for the t+I+R setting, refer to Table 2.5) achieves an F-1 measure of 0.701. In this case, the model outperforms 21 out of the 24 baselines. The other two baselines that performed better than the proposed model are unigrams (U) for the t+I+R setting and unigrams+POS (U+POS) for the t+I+R setting with an F-1 measure of 0.691 and 0.69 respectively. Thus, we see that we achieve classification performance which is similar to, and, in most cases, better than that obtained from the baseline features by using thread specific features which are much less in number (number of baseline features is of the order of the size of the vocabulary whereas number of features in the proposed model = 25.)

Metric	**Trip–Advisor**	**Ubuntu**
Classification Accuracy	77.01%	70.05%
Precision	0.763	0.692
F1-Measure	0.764	0.684

Table 2.6: Classification results.

	Trip–Advisor			**Ubuntu**		
	Precision	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**
Subjective class	0.805	0.871	0.837	0.647	0.429	0.516
Non-subjective class	0.675	0.558	0.611	0.718	0.862	0.783
Overall	0.763	0.77	0.764	0.692	0.7	0.684
Best performing baseline	0.738	0.747	0.723	0.72	0.696	0.701

Table 2.7: Classification performance of the proposed model for subjective and non-subjective classes on the two datasets.

2.5.4.3 Relative Performance of Different Types of Features

In this subsection, I investigate the effect of different types of features used for the subjectivity classification task. I perform the classification experiment using only one type of feature at a time. Table 2.9 shows the relative performance of different types of features. We see that, for both the datasets, structural features

gave the best performance which confirms the hypothesis that thread structure is a strong indicator of its subjectivity orientation. Lexicon-based and Sentiment features are the second best performing features, outperforming the dialogue act features, for the Trip–Advisor forum whereas for the Ubuntu forum, dialogue act features outperform the two types of features with sentiment features being the worst performing and Lexicon-based features being the third best performing features. This difference in the relative performance of Sentiment and Lexicon-based features across the two forums may be attributed to the difference in the nature of the two forums. Trip–Advisor is a non-technical forum where majority of discussions are subjective in nature and hence there are more number of subjectivity clues and sentiment indication patterns for the classifier to learn, whereas discussions in Ubuntu forum are technical and hence, usually, non-subjective in nature.

Further, I use ensemble methods and a stacked classification approach. I make ensembles of the four classifiers corresponding to the four types of features. For a test instance, I calculate the final prediction of the ensemble using two methods: 1) averaging the confidences of the four classifiers (denoted by $EnsembleAvg$), 2) taking prediction of the most confident classifier out of the four classifiers (denoted by $EnsembleMostConf$). Next, I used a stacked classifier where the confidences of the four classifiers were provided as features for the second stage classifier. Finally, I build classifier using the combined feature set. The model is denoted by $FeatureAll$. We see that combined performance of all the features ($FeatureAll$ model) is better than the performances of all the individual types of features. However, ensemble models perform worse than the combined feature model and models built on individual feature types. This suggests that the predictions of the four classifiers are quite different from each other. The conflicts between the classifiers in terms of their predictions result in a lower performance of the ensemble models. Similarly, the stacked classifier performed worse than the classifier built using all the features.

For further analysis, I conduct classification experiments on threads on which the annotators disagreed. I train the classifier on the threads on which the annotators agreed and test it on the two set of threads and their labels corresponding to the two annotators. Table 2.8 reports the results of this set of experiments. We note that for both the annotators, classification performance is worse than the

Annotator	Precision	Recall	F-1 score
Trip Advisor			
Annotator 1	0.54	0.543	0.54
Annotator 2	0.465	0.48	0.47
Ubuntu			
Annotator 1	0.392	0.445	0.433
Annotator 2	0.346	0.37	0.367

Table 2.8: Classification results on threads on which the annotators disagreed.

classifier trained on the threads on which the annotators agreed. This is expected as the cases where annotators disagreed must be difficult for the learned classifier and hence the decreased performance for those threads.

	Trip–Advisor			Ubuntu		
Class	**Pr.**	**Re.**	**F-1**	**Pr.**	**Re.**	**F-1**
Structural	0.741	0.75	0.742	0.692	0.697	0.67
Dialogue Act	0.683	0.703	0.683	0.639	0.654	0.598
Subjectivity Lexicon Based	0.713	0.727	0.699	0.622	0.643	0.569
Sentiment	0.71	0.726	0.699	0.534	0.602	0.525
EnsembleAvg	0.644	0.681	0.662	0.646	0.65	0.648
EnsembleMostConf	0.631	0.678	0.65	0.6	0.627	0.613
Stacked Classifier	0.74	0.749	0.741	0.678	0.688	0.663
AllFeatures	**0.762**	**0.768**	**0.763**	**0.692**	**0.7**	**0.684**

Table 2.9: Classification results for NYC and Ubuntu datasets obtained using different types of features.

2.5.4.4 Most Informative Features

I study the importance of individual features by measuring the chi-squared statistic with respect to the class variable. Table 2.10 shows top 10 features, ranked by their chi-square values. From the table, we note that, for both the datasets, five out of six structural features (ThreadLength, NumPost, AvgPostLength, NumAuthor, InitPostLength) are among the top 10 most informative features which again confirms that a thread's structure is a strong indicator of its subjectivity.

We note that the lexicon-based features and the sentiment features have relatively higher ranks in Trip Advisor dataset as compared to the Ubuntu dataset. We also note that, for Trip–Advisor, two of the three lexicon-based features (NumSubReply, NumSubInit) are among the top 10 features whereas for Ubuntu, only one lexicon-based feature (NumSubReply) is ranked among the top 10 features. This observation is consistent with the previous observation where we noted that sentiment and lexicon-based features performed relatively better in Trip–Advisor as compared to Ubuntu and can be attributed to the difference in the nature of the two forums as explained in the previous subsection. Among the lexicon-based features, NumSubReply is the most informative feature which suggests that, for a thread, reply posts are more helpful than initial post and title of the thread in identifying the thread's subjectivity. This is also manifested in case of sentiment features where features corresponding to reply posts (ReplySentiStrngPos, ReplySentiAvgNeg, etc.) are ranked higher than the corresponding features for the initial post (which are not in the top 10 list). These observations are consistent with the results we got from the baselines where we found that incorporating text from reply posts gave the best performance across all the features. We note that, for Ubuntu, there is one dialogue act feature (NumSol) in the top 10 list, whereas for Trip–Advisor, none of the dialogue act features are in the list.

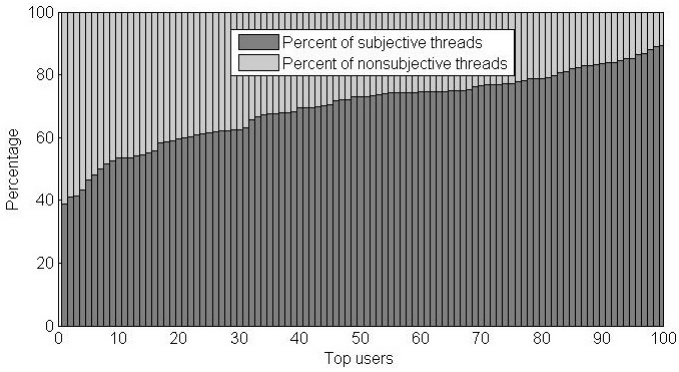

Figure 2.4: Figure showing distribution of threads from top 100 users in subjective and non-subjective classes for Trip Advisor – New York forum.

Next, I analyze the behavior of users in the two forums in terms of starting a subjective or non-subjective thread. I used the subjectivity classifiers to predict labels of all the threads in the two datasets. Since most of the users have started

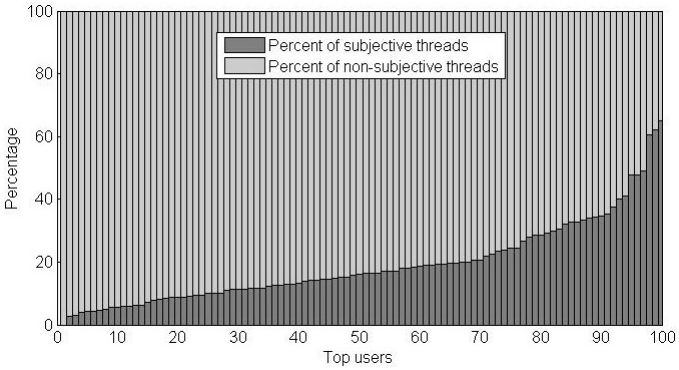

Figure 2.5: Figure showing distribution of threads from top 100 users in subjective and non-subjective classes for Ubuntu forum.

very few threads, I take into account top 100 users for the two forums. I ranked the users according to the number of threads they have started and selected top 100 users from the ranked list for both the forums. For Trip–Advisor forum, the top 100 users have started 43 or more threads and the top user has started 397 threads. For Ubuntu forum, the top 100 users have started 28 or more threads and the top user has started 140 threads. Figure 2.4 and Figure 2.5 show the distribution of threads started by top 100 users in the subjective and non-subjective classes for Trip–AdvisorNew York and Ubuntu forums respectively. Each vertical bar corresponds to one of the top 100 users. Yellow and red portions in a bar represent the percentage of subjective and non-subjective threads started by the user represented by the bar. We note that most of the users started more subjective threads (than non-subjective) in Trip Advisor forum whereas in Ubuntu forum, most users started more number of non-subjective threads. We also observe that in Trip – Advisor forum, more users have higher percentage of non-subjective threads (than subjective threads) as compared to Ubuntu forum where a very few users have started more subjective threads than non-subjective threads.

2.5.5 Retrieval Results

Table 2.11 presents retrieval results for subjective and non-subjective queries, and the overall average result. **Subjectivity Prior Model** denotes the setting where thread's subjectivity probability is used as its prior relevance probability (as explained in Section 2.4.3). We see that using subjectivity information of threads

Trip–Advisor	**Ubuntu**
ThreadLength	ThreadLength
NumSubReply	NumPost
AvgPostLength	NumSubReply
NumPost	NumUser
NumUser	AvgPostLength
ReplySentiStrngPos	InitPostLength
ReplySentiAvgNeg	NumSol
InitPostLength	ReplySentiAvgNeg
ReplySentiAvgPos	ReplySentiStrngPos
NumSubInit	ReplySentiStrngNeg

Table 2.10: Top 10 features ranked by chi-square values for the two datasets.

improves MRR, MAP and NDCG values for both subjective and non-subjective queries against the baseline model. We also note that precision values remain almost unchanged (across all the settings) till rank 25. This is an interesting observation and suggests that subjectivity match does not help much in finding more relevant threads. Instead, it improves ranking of threads by changing relative ordering of ranked threads. MAP takes into account ordering of ranked results and NDCG takes into account ordering and graded relevance (0, 1, 2) of the ranked results. Also, for higher ranks, re-ranking results in performance drop with re-rank at 100 giving worst results. This is because at higher ranks, we start getting lexically less relevant or irrelevant threads and sorting by subjectivity probabilities may bring those threads at lower ranks, hence, worsening the performance. This is consistent with our previous observation that subjectivity match does not identify relevant documents by itself but helps getting more relevant documents from a set of lexically relevant documents.

For the re-ranking setting, we see that the performance worsens as we increase ranks after 5 and 20 for subjective queries and non-subjective queries respectively. Specifically, for subjective queries, re-ranking at 5 outperforms other settings with improvement of 11.96%, 21.03% and 10.4% in NDCG@5, MAP@5 and MAP@10 respectively over baseline model. For MRR, **LexSubjCombine model** performs best with 14.52% improvement. For NDCG@10, **Subj. Prior Model** performs best with 3.51% improvement. For non-subjective queries, re-ranking at 15 per-

Model	MRR	P@5	P@10	NDCG@5	NDCG@10	MAP@5	MAP@10
				Subjective queries			
Baseline	0.8033	0.56	0.52	0.7745	0.7180	0.7264	0.6662
Top 5 Re-rank	0.85	0.56	0.52	**0.8672**	0.7322	**0.8792**	**0.7355**
Top 10 Re-rank	0.7833	0.58	0.52	0.7433	0.6964	0.7504	0.6873
Top 15 Re-rank	0.7333	0.6	0.55	0.7370	0.7018	0.7361	0.6956
Top 20 Re-rank	0.6666	0.58	0.51	0.6958	0.6619	0.6687	0.6324
Top 25 Re-rank	0.6833	0.6	0.56	0.6940	0.6946	0.6764	0.6480
Top 50 Re-rank	0.6833	0.48	0.48	0.6966	0.6590	0.5939	0.5654
Top 100 Re-rank	0.5833	0.4	0.42	0.5634	0.6422	0.5229	0.5061
Subj. Prior Model	0.825	0.56	0.54	0.8010	**0.7433**	0.7880	0.6882
LexSubjCombine Model	**0.92**	0.6	0.53	0.8067	0.7601	0.7853	0.7270
				Non-subjective queries			
Baseline	0.7288	0.546	0.546	0.6838	0.6988	0.7	0.651
Top 5 Re-rank	0.7733	0.546	0.546	0.7056	0.7263	0.6688	0.6499
Top 10 Re-rank	0.7911	0.56	0.546	0.8148	0.7644	0.7475	**0.7078**
Top 15 Re-rank	7966	0.546	0.533	**0.8220**	**0.7658**	**0.7938**	0.6761
Top 20 Re-rank	**0.8111**	0.546	0.546	0.7556	0.7514	0.7462	0.6803
Top 25 Re-rank	0.7539	0.546	0.5066	0.7485	0.7160	0.7301	0.6608
Top 50 Re-rank	0.6333	0.5066	0.4866	0.6533	0.6542	0.6278	0.5870
Top 100 Re-rank	0.5358	0.2533	0.32	0.5080	0.4989	0.4496	0.4152
Subj. Prior Model	0.7355	0.546	0.546	0.7827	0.7597	0.7518	0.7045
LexSubjCombine Model	0.78	0.5467	0.5067	0.7314	0.7347	0.7481	0.6956
				Average			
Baseline	0.7586	0.552	0.536	0.7201	0.7065	0.7105	0.6572
Top 5 Re-rank	0.804	0.552	0.536	0.7703	0.7286	0.7530	0.6842
Top 10 Re-rank	0.7880	0.568	0.536	0.7862	0.7372	0.7486	0.6996
Top 15 Re-rank	0.7713	0.568	0.54	0.7880	0.7402	**0.7707**	0.6840
Top 20 Re-rank	0.7533	0.56	0.532	0.7317	0.7156	0.7152	0.6612
Top 25 Re-rank	0.7257	0.568	0.528	0.7267	0.7075	0.7086	0.6557
Top 50 Re-rank	0.6533	0.496	0.484	0.6706	0.6561	0.6142	0.5784
Top 100 Re-rank	0.5548	0.312	0.36	0.5302	0.5563	0.4789	0.4516
Subj.Prior Model	0.7713	0.552	0.544	**0.7900**	**0.7532**	0.7663	0.6980
LexSubjCombine Model	**0.836**	0.568	0.516	0.7766	0.745	0.763	**0.7082**

Table 2.11: Retrieval results.

forms best in terms of NDCG@5, NDCG@10 and MAP@5 with improvement of 20.21%, 9.58% and 13.4% respectively. For MRR and MAP@10, re-ranking at 10 and 20 outperform other settings respectively with corresponding improvement of 10.15% and 8.72%. For average results, **Subj. Prior Model** performs best in terms of NDCG@5 and NDCG@10 with improvement of 9.7% and 6.61%. **LexSubjCombine model** performs best in terms of MRR and MAP@10 with improvements of 10.2% and 7.76% respectively. Re-ranking at 5, 10 and 15 performs best in terms of MRR, MAP@5 and MAP@10 respectively with corresponding improvement of 5.98%, 8.47% and 6.45%. We note that the improvements for the two types of queries are much higher than improvements in average results. This is because we have different settings performing best for the two types of queries.

For subjective queries, re-ranking at low ranks seems to work better whereas for non-subjective queries, re-ranking at higher ranks works better.

Model	MRR	P@5	P@10	NDCG@5	NDCG@10	MAP@5	MAP@10
			Subjective queries				
Baseline	0.814	0.6	0.543	0.756	0.74	0.727	0.684
Top 5 Re-rank	0.786	0.6	0.543	**0.81**	0.727	**0.827**	0.716
Top 10 Re-rank	0.786	0.543	0.543	0.722	0.701	0.753	0.682
Top 15 Re-rank	0.786	0.628	0.543	0.777	0.740	0.773	**0.724**
Top 20 Re-rank	0.69	0.628	0.514	0.714	0.697	0.681	0.650
Top 25 Re-rank	0.643	0.6	0.543	0.673	0.693	0.636	0.638
Top 50 Re-rank	0.69	0.457	0.486	0.726	0.669	0.589	0.561
Top 100 Re-rank	0.619	0.43	0.457	0.645	0.697	0.587	0.545
Subj. Prior Model	**0.821**	0.543	0.543	0.788	**0.7463**	0.78	0.665
			Average				
Baseline	0.758	0.552	0.536	0.720	0.706	0.710	0.657
Top 5 Re-rank	0.777	0.552	0.536	0.748	0.719	0.72	0.666
Top 10 Re-rank	0.7880	0.544	0.536	0.789	0.735	0.746	**0.691**
Top 15 Re-rank	**0.7913**	0.56	0.53	**0.81**	**0.746**	**0.779**	0.683
Top 20 Re-rank	0.7733	0.56	0.528	0.751	0.726	0.725	0.665
Top 25 Re-rank	0.7257	0.552	0.512	0.735	0.703	0.703	0.65
Top 50 Re-rank	0.666	0.488	0.484	0.693	0.656	0.628	0.584
Top 100 Re-rank	0.588	0.328	0.376	0.583	0.574	0.521	0.477
Subj.Prior Model	0.765	0.536	0.5	0.788	0.745	0.756	0.684

Table 2.12: Retrieval results of experiments conducted using three categories for queries: subjective, non-subjective and none.

As mentioned in Section 2.5.1, there were three queries on which the first two annotators disagreed and a third annotator was then asked to disambiguate their tags. The final tags of all the three queries is subjective (Table 2.3). For further analysis, I conduct retrieval experiments by taking the final tags of the three queries as "ambiguous", i.e., neither subjective nor non-subjective, and hence do not use the subjectivity match for those queries. That is, for these three queries, I simply use the baseline model. Table 2.12 reports results for this set of experiments. Since there will be no change in the retrieval performance for non-subjective queries in this set of experiments, I only report results for subjective queries and average results. We see that the overall retrieval performance remains almost the same as in Table 2.11. This is expected because of the minor change in the settings of the two experiments. However, it would be interesting to see if having third category of "neither subjective nor non-subjective" for queries improves retrieval performance on a larger dataset with more queries and threads. I leave it for future work.

2.6 Chapter Summary

In this chapter, I discussed subjectivity classification of online forum threads and its application in improving thread retrieval. I proposed a supervised machine learning model for subjectivity classification. I used two types of novel thread-specific features, structural features and dialog act features, in addition to lexicon-based and sentiment features for the classification task. I evaluated the proposed model by comparing it with various state-of-the-art techniques used for subjectivity classification and showed that the model outperformed them in most of the cases. A major contribution of this work is the introduction of thread-specific features for subjectivity classification of online forum threads which significantly reduces the complexity of the learning model compared to that of the models built on lexical features without compromising the performance of the model. I also incorporated the subjectivity information of threads in a state-of-the-art thread retrieval model and showed that by combining lexical match and subjectivity match between user queries and threads, thread retrieval can be improved.

Chapter 3

Sentiment Analysis of an Online Cancer Support Community Using Co-training

3.1 Introduction

Millions of people discuss their health-related issues on these forums, ask questions about their ailments, the symptoms they experience, medications, side-effects, and share their health concerns to get emotional support [39, 40, 41]. Although it may seem odd at first to share and discuss such important issues with unknown people, or to consult health-related websites, there is substantial value in doing so [42]. People feel much better and change to positive attitudes if they talk to other people after or during a traumatic event such as a disease [43, 44, 42, 45, 46].

Analyzing these large archives of discussions in health communities can help network managers obtaining crucial information about these communities, e.g., identify dominant health issues in the community, understand the effects of interactions on emotional states of individual members, etc. Understanding emotional impacts of online participation on members can help improve the features and functionalities of the community portals to enable facilitation of emotional support to the network members more effectively.

In this chapter, I analyze the sentiment of user messages (or posts) of an online

cancer support community, the Cancer Survivors' Network (CSN) of the American Cancer Society. I identify whether a post is positive or negative based on the polarity of the emotion expressed in it and model the task of sentiment analysis as a binary classification problem. I show that a semi-supervised machine-learning based model using co-training improves the accuracy of classification of posts into two classes: posts with positive sentiments and posts with primarily negative sentiments.

3.1.1 Why Sentiment Analysis of CSN Posts?

Sentiment analysis of CSN posts can help understand the dynamics of the community as follows:

1. **Understand effects of interactions on members:** CSN members interact to get social support. Analyzing sentiment of posts of a member over a period of time can help us identify the emotional effects of interactions on that member. A member may show a positive change in sentiment upon getting replies from other members. A negative change of sentiment may happen upon not getting replies, getting discouraging replies from some members, or other factors.

2. **Find influential members:** Every community has a set of members who influence other members in the community. These members are called leaders or influential members. Influential members may have significant and consistent role in positively changing the sentiments of other members by interacting with them.

3. **Recommend support providers:** Identification of influential members can be used to provide suggestions regarding potential support providers to support seekers. Also, the influential members can be notified when there are new posts seeking emotional support.

4. **Interventions:** Sentiment detection from posts can explicate the sentiment of the writer and has many applications. For example, with consent, a nurse may observe the sentiment and the change of sentiment in posts by a patient to get an advance warning of the deterioration of the sentiment of the patient.

This may, in turn, result in interventions that may save the life of the patient or improve the patients quality of life.

5. **Viewpoints:** Sentiment detection may be used to obtain a balanced set of viewpoints and exam-ples/anecdotes of medicines/treatments that worked and did not work in patients.

Sentiment detection is hard because despite advances in natural language processing, understand-ing the meaning of sentiment words and the subtlety of their meanings automatically is hard. Most research in sentiment analysis has focused on the domain of online product review sites, question-answering sites and online forums [47, 17, 48]. Online health communities (OHCs) are different from these forums with respect to the content of the discussions. In OHCs, discussions tend to be emotional in nature with people seeking and providing emotional support, whereas in other social communities, discussions are more opinionated and the emotional aspect is not predominant. For example, consider the two user messages posted in a camera review site and a cancer support community site, respectively: 1) I love the screen of DSC-W650, however the buttons are too small to operate, and 2) Will be undergoing double mastectomy next week. Feeling quite nervous!

This difference between emotional and opinionated discussions has implications on the approaches that can be used for sentiment analysis. For example, last sentences of the posts play an important role in the classification of sentiments in OHCs due to the presence of two types of emotional support that can be identified in health communities: direct and indirect emotional support (as will be explained in Last Sentence Effect section).

Common approaches for learning sentiment classifiers are based on supervised methods that rely on the availability of labeled data. Labeling data is costly. Thus, the amount of labeled data is often much smaller compared to the entire dataset. I use a semi-supervised approach wherein I use a small set of tagged samples and a large set of untagged samples to build a classifier that identifies positive and negative sentiments in posts.

The semi-supervised learning approach incorporates information from the unlabeled data into the models. I use two types of sentiment features for classification: domain-independent (DI) features and domain-dependent (DD) features. DI

features represent polarity clues [16, 32], emoticons, punctuation marks (among others), which are used to express sentiments in online social media, in general. DD features, on the other hand, are specific to a particular community. I use various DD features such as n-grams and POS tags, extracted from the CSN posts, and thus, specific to CSN. Previous works on sentiment analysis have used these features separately in supervised learning settings [47, 46]. To the best of my knowledge, this work is the first to use these two types of features as two different views of the data in a co-training setting to perform sentiment classification in a health related domain. I first train two supervised classifiers using DI and DD features and then combine them in such a way that one classifier can "guide" the other to minimize the number of mistaken examples. Experimental results show that the domain-dependent and domain-independent views can be used successfully in a co-training setting to improve sentiment classification in a health domain over previous approaches. The classifiers that incorporate information from unlabeled data achieve an F-1 score of up to 0.85, outperforming strong baselines by 6%-20%.

3.2 Related Work

Sentiment analysis has been a highly active research area due to its important applications in mining, analyzing and summarizing user opinions from online sites such as product review sites, forums, Facebook, and Twitter [16, 17, 49, 50]. It essentially deals with identifying the polarity (positive or negative) of a piece of text (often with respect to a particular target). Here, I survey some of the sentiment analysis works.

Pang et al. [47] use supervised machine learning algorithms for sentiment analysis of movie reviews. They train their models using lexical features such as unigrams, bigrams, POS tags, etc. In their later work, they improve the sentiment classification by considering only the subjective sentences and applying polarity classifiers (developed in their previous work) on those sentences [51]. McDonald et al. [52] use joint models based on sequence labeling for sentiment classification at sentence and document level for product reviews. Wan [53] performs sentiment classification of Chinese product reviews using co-training. He used machine

translation to obtain the training data from labeled English reviews. For a Chinese review, he used its Chinese features and translated English features as the two independent views and used them to train their classifiers in a co-training setting. Li et al. [54] use active learning for sentiment classification to address the issue of imbalanced distribution of positive and negative samples in the dataset. They identify highly informative examples from the minority class for manual annotation and automatically label the most informative majority class sample, thus reducing manual annotation efforts.

Recently, there has been a growing interest in analyzing sentiments about various topics/themes/issues talked in social media[55, 48, 50, 49, 56]. Pal et al. [55] detect sentiments about a set of pre-defined themes in blogs. Stavrianou et al. [48] propose an opinion-oriented graphical model for extracting information about opinions expressed in online forums. They focus on certain opinion information (e.g., opinion flow and attitudes of users), that cannot be extracted from a social network graphical model. Unlike social network graphical models, where a user is represented as a node, they use message posts as nodes and take into account the structure of forum threads based on the reply relations between message posts, time of posts, etc. Jiang et al. [50] perform target-dependent sentiment classification of tweets. They do polarity classification on the subjective content of tweets and finally take into account the contextual information of tweets (replies and retweets) using graphical models to improve sentiment classification. Neri et al. [49] analyze opinions expressed on Facebook about an Italian company. They use various state-of-the-art techniques to develop their sentiment and knowledge mining system. Bermingham et al. [56] analyzed comments on the videos posted by particular YouTube channels for detecting online radicalization. They use sentiment analysis tools for detecting sentiments about sensitive topics (related to religion) and network analysis tools to relate gender and influence of a person (in the network).

Similar to the current work, Qiu et al. [46] perform sentiment classification of CSN posts. They use sentiment features, e.g., sentiment clues, sentiment strength, punctuation marks, and two content features: *name* and *slang* to train their classifiers in a supervised setting. In contrast, I use semi-supervised methods and additional features. Specifically, I use domain-specific features such as unigrams,

bigrams, and their POS tags. I also extract some of the features for post classification from the last sentence of a post.

3.3 Approach

3.3.1 Problem Formulation

CSN is a dynamic online community of cancer survivors, their families and friends where users interact by posting messages in discussion threads. In CSN, members discuss cancer related issues such as cancer medications, side-effects, procedures, cancer experiences, etc. Many messages are emotional. Often, members start a discussion thread for seeking support from the community by posting questions or concerns about cancer-related issues, updates about their cancer reports, health conditions, etc. Other members provide emotional support by sharing their own experiences, which often make the (support) seeker feel better. These posts are indicative of the *dynamic* emotional states of the posters. I say that a post is positive if its overall sentiment is positive and negative if its overall sentiment is negative. The overall sentiment of a post is determined by its content as well as its context. For example, a poster may use negative/neutral words with a positive intent of providing (indirect) emotional support (see Table 2) and hence, in this case, the post is labeled as positive even if some of its content is negative because the context of the post (the poster's intent) is positive. I model the task of sentiment analysis as a binary classification problem. **Problem Statement:** Given a post of a discussion thread in CSN, classify it into one of the two classes: *Positive* (denoted by +) and *Negative* (denoted by -).

3.3.2 Feature Engineering

I describe domain-independent features and domain-dependent features used in the classification model and my intuition for these features. Table 1 summarizes all the features.

3.3.2.1 Domain-independent Features

Domain-independent features are the features used for sentiment analysis across domains. I use three types of domain-independent sentiment features: (1) Polarity clues, (2) Sentiment Strength, and (3) Punctuation marks.

(a) **Polarity Clues:** These are the words/phrases/symbols used to express polarity of opinions/emotions in speech or written text. Polarity clues are a good indicator of the polarity of a piece of text and have been used extensively in sentiment analysis [16, 57]. Also, in online interactions, emoticons (such as ":)", ":(", ":-D", etc.) are widely used to express emotional states. We expect the distribution of these positive and negative polarity clues to be different in positive and negative posts. I use the frequency of occurrences of these clues in the post as features for classification. I adopted the list of positive and negative words created by Hu and Liu [16], and I used the list of emoticons available from Wikipedia [1]. I extract three features PosDensity, NegDensity and PosVsNegDensity from a post and its last sentence (See Section 3.3.3 for details related to features of the last sentence). PosDensity is the number of positive polarity clues (positive words and positive emoticons) normalized by the number of words in the post. NegDensity is computed in a similar way. PosVsNegDensity is the number of positive polarity clues over negative polarity clues and is calculated by dividing $(PosDensity + 1)$ by $(NegDensity + 1)$.

(b) **Punctuation Marks:** In online interactions, punctuation is often used to show the intensity of emotions. For example, *I like it!* and *I like it !!!!!!!!* express different intensities of emotion. The intensity of positive emotion is higher in the latter case. I extract question marks and exclamation marks from a post and used their frequency of occurrence in the post as features. I calculate four punctuation features for a post: numQuestion (number of question marks), isQuestion (whether a post contains a question mark or not), NumExclaim (number of exclamation marks) and isExclaim (whether a post contains an exclamation mark or not).

[1] http://en.wikipedia.org/wiki/Listofemoticons

(c) **Sentiment Strength:** These features capture the strength of the sentiments expressed in posts. To calculate sentiment strength, I used the SentiStrength algorithm [33]. The algorithm is specifically designed to calculate sentiment strength of short informal texts in online social media. For a piece of text, the algorithm computes a positive sentiment value and a negative sentiment value. Using SentiStrength, I compute three features: PosStrength (positive sentiment strength), NegStrength (negative sentiment strength) and PosVs-NegStrength (PosStrength/NegStrength). I extract these features from a post and its last sentence.

3.3.2.2 Domain-dependent Features

There are often terms/expressions in a domain that are expressive of or are associated with a particular sentiment. These terms may or may not occur in other domains or, if they occur, then they may be associated with different sentiments. For example, the term *positive* is generally associated with positive sentiment, but in the context of cancer reports it is typically associated with negative sentiment because it is often used to describe positive test results. Similarly, *unpredictable* is a positive attribute for a movie plot, but negative for effects of a medication. To find these domain-dependent terms/expressions, I extract various combinations of unigrams, bigrams and part-of-speech tags. These features have been shown to perform well in sentiment classification [47, 52]. From each sentence in a post, I extract the following features:

1. **Unigrams (BoW):** All words of a sentence.
2. **Unigrams + POS tags (Uni+POS):** All words of a sentence and their part-of-speech tags.
3. **Unigrams + bigrams (Uni+Bi):** All words and sequences of two consecutive words in a sentence.
4. **Unigrams + bigrams + POS tags (Uni+Bi+POS):** all words, their part-of-speech tags and sequences of two consecutive words in a sentence (see Table 1).

Feature Name	Description
Polarity clues (extracted from post and its last sentence)	
PosDensity	Number of postive polarity clues in the post normalized by the number of words.
NegDensity	Number of negative polarity clues in the post normalized by the number of words.
PosVsNeg	Number of positive sentiment words per negative sentiment word. Calculated as (No. of positive polarity clues+1)/(No. of negative polarity clues+1)
Sentiment strength features (extracted from post and its last sentence.)	
PosStrength	Positive sentiment strength of the post as given by SentiStrength algorithm.
NegStrength	Negative sentiment strength of the post as given by SentiStrength algorithm.
PosVsNegStrength	PosStrength divided by NegStrength
Punctuation Marks	
numQues	Number of question marks in the post.
isQues	Whether the post has a question mark or not (1 for yes, 0 for no)
NumExclaim	Number of exclamation marks in the post.
isExclaim	Whether the post has a exclamation mark or not (1 for yes, 0 for no)
Domain-dependent features (for a sentence containing three words W_i, W_{i+1}, W_{i+2} with POS tags $POS_i, POS_{i+1}, POS_{i+2}$ respectively.)	
BoW	W_i, W_{i+1}, W_{i+2}
$Uni + POS$	$W_i, POS_i, W_{i+1}, POS_{i+1}, W_{i+2}$, POS_{i+2}
$Uni + Bi$	$W_i, W_{i+1}, W_{i+1}, W_iW_{i+1}$, $W_{i+1}W_{i+2}$
$Uni + Bi + POS$	$W_i, POS_i, W_{i+1}, POS_{i+1}, W_{i+2}, POS_{i+2}, W_iW_{i+1}, W_iPOS_{i+1}$, $POS_iW_{i+1}, W_{i+1}W_{i+2}, W_{i+1}POS_{i+2}, POS_{i+1}W_{i+2}$

Table 3.1: Description of various features used for sentiment classification.

For feature encoding, I experimented with *tf* (term frequency), *tf-idf* (term frequency - inverse document frequency) and *binary* (if a term occurs in a post or not) and found that *tf* encoding performs better than the other two encodings. In the experiments, I set the minimum document frequency *df* of a term to 5 (I experimented with document frequencies 3, 5 and 10 and found that $df = 5$ gave the best results. The minimum document frequency is defined as the minimum

number of posts in which a term occurs in order to be included in the dictionary.

3.3.3 Last Sentence Effect

In this sub-section, I explain the reasoning behind extracting some of the features mentioned above from last sentences of the posts. In CSN, members provide emotional support to the support seekers mainly in two ways: Direct Emotional Support (DES) and Indirect Emotional Support (IES). In DES, they provide support *directly* by giving encouragement, positive feedback, sympathy, or other sentimental support *without* talking about their own experiences. In IES, they encourage the seeker *indirectly* by sharing their own stories about cancer, situations, experiences, and struggles about cancer that are similar to those of the seeker so that they can relate themselves to the seeker in an effective manner and, finally, talking about their success in dealing with those situations to provide encouragement. This difference between DES and IES results in structural and lexical differences between the posts providing the two types of support. DES posts usually have a positive tone throughout and, hence, mostly contain positive sentiment words. In contrast, IES posts either have neutral tone (e.g., description of treatments, medications, etc.) or have negative tone (e.g., description of feelings and struggles related to personal experience of cancer treatments and side effects of medications) in the major part of the post and positive tone in the end, when the support providers talk about their successes in dealing with situations and say encouraging words. Hence, IES posts often use factual and negative sentiment words to share their personal cancer experience and positive words in the end (usually in the last sentence). To capture these patterns, I extract Polarity Clues and Sentiment Strength features from the entire post and its last sentence, separately. Table 3.2 shows a thread where the thread starter seeks emotional support and the second post and the third post of the thread provides DES and IES, respectively. In many cases, members only write a few words, e.g., *see you, bye*, or their names at the end of the post. To deal with these situations, I define the last sentence to be one with at least three words.

Post	Support type	Sentiment
Will be undergoing double mastectomy next week. Feeling quite nervous as I am sure you all can relate to! Name_A	Seeking emotional support.	Negative
Will keep you in prayers. Good luck!	Providing direct emotional support.	Positive
I undergone a bi-lateral mastectomy with Trampflap reconstruction. It wasn't easy for me to prepare myself and was feeling very nervous. Finally, I thank god as it went well and I found the reconstructed breast acceptable. Good luck Name_A !	Providing indirect emotional support.	Positive

Table 3.2: An example thread showing posts containing direct and indirect emotional support along with their sentiments.

3.3.4 Model Training

Annotating large amounts of data for training high-accuracy classifiers is time-consuming and costly. It is more feasible to have a small amount of annotated data to provide initial supervision to a classifier and then use the unannotated data (which is readily available in large amounts) to improve the classifier's performance which can be achieved using semi-supervised learning such as co-training.

Co-training was originally developed by Blum and Mitchell [58] and applied to webpage classification. One of the requirements for co-training to work is that the data can be represented using two independent views. However, recent works show that the independence criteria can be relaxed [59, 60] without much impacts on the performance. For the webpage classification problem, the content of a webpage and its hyperlink are used as two independent views of a webpage. Hence, two separate classifiers are trained on the two feature sets corresponding to the two views of the data (e.g., the words extracted from the content of a webpage and the words extracted from its hyperlink or the anchor). The two constructed classifiers are then used to predict labels of unlabeled instances. The unlabeled instances

ALGORITHM 1: Co-training algorithm

***I*nput**: Labeled set for training L, Unlabeled set U, number of iterations K, number of negative and positive instances n and p respectively to be moved from the unlabeled set to the labeled set after each iteration, 's' number of unlabeled instances to be moved to a smaller unlabeled set S same as in Blum and Mitchell [58]

$j \leftarrow 0$

Sample 's' posts from U and move them to S.

$U \leftarrow U \setminus S$

while $j \leq K$ && $U \neq \phi$ **do**

  Use L_I to train classifier C_I.

  Use L_D to train classifier C_D.

  $(H_P^I, H_N^I) \leftarrow GetHighConfPost(C_I, S, n, p)$

  $(H_P^D, H_N^D) \leftarrow GetHighConfPost(C_D, S, n, p)$

  $L \leftarrow L \cup (H_P^I \setminus H_N^D) \cup (H_N^D \setminus H_P^I) \cup (H_N^I \setminus H_P^D) \cup (H_P^D \setminus H_N^I)$

  Sample $2p + 2n$ posts from U, move them to S and remove them from U.

  $j \leftarrow j + 1$

end while

***O*utput**: Classifiers C_I, C_D.

predicted with the highest confidence by both classifiers are moved to the labeled data and removed from the unlabeled set, ensuring that conflicting instances (i.e., instances predicted with high confidence by both classifiers, however, in opposite classes) are discarded. This process is repeated until the number of iterations reaches a particular threshold or all the unlabeled data are used up. The idea is that the two classifiers "teach" each other by re-training each classifier at each iteration on the data enriched with labeled instances predicted with high confidence by the other classifier. It has been shown that when there exist a split of features into two independent views, co-training is able to outperform baseline classifiers [58, 61].

I use domain-independent (DI) and domain-dependent (DD) sentiment features as the two views of a post for sentiment classification. The co-training algorithm is shown in Algorithm 1. L and U denote the labeled and the unlabeled datasets, respectively. I and D correspond to the domain-independent and domain-dependent views, respectively. N and P represent positive and negative instances, respectively. Classifiers C_I and C_D are trained on the two views, L_I and L_D, of L respectively and make predictions on unlabeled instances in set S that is created by sampling s instances from U. Instances whose labels are predicted with high confidence (H_P^I, H_N^I and H_P^D, H_N^D) are selected using the method $getHighConfPost$ for both the classifiers (C_I and C_D) and are added to L such

that the conflicting instances are discarded. The method adds n highest confidence negative and p highest confidence positive examples from U to L, for different values of n and p (as described in Section 3.4.4.2). The process is repeated K times or until the unlabeled data is exhausted. During the testing phase, the final prediction (on a test instance) is computed by taking the product of the confidence of the two (trained) classifiers as in [58]. Figure 3.1 shows the schematic of the proposed co-training classification framework.

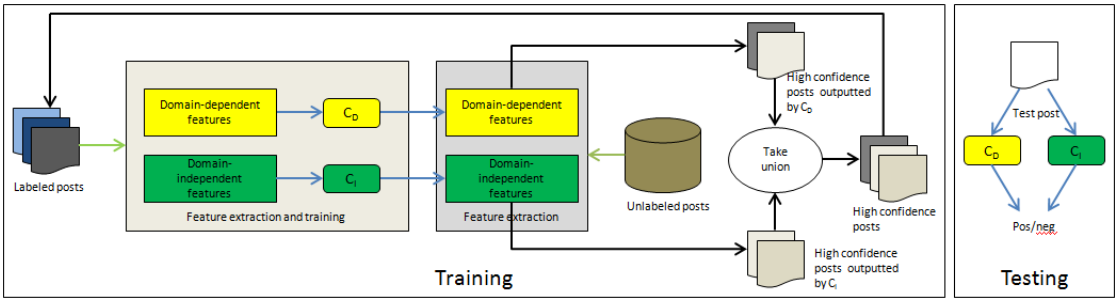

Figure 3.1: Classification model. C_I and C_D are the classifiers trained on domain-independent and domain-dependent features respectively.

3.4 Experiments and Results

I now describe the data and the experimental setting, and present results.

3.4.1 Data

The data used in this work comes from a popular online cancer support community, the Cancer Survivors' Network (CSN), developed and maintained by the American Cancer Society. CSN is an online community for cancer patients, cancer survivors, their families and friends. The features of CSN are similar to many online forums with dynamic interactive medium such as chat rooms, discussion boards, etc. Members of CSN post in discussion boards for seeking and sharing information about cancer related issues, for seeking and providing emotional support and other social support such as celebration of birthdays and success stories. To conduct the experiments, I used posts from the discussion boards of CSN between June 2000 to June 2012. A dataset of 786,000 posts from 75,867 threads

was downloaded. For the labeled data for sentiment classification, I used the same subset of annotated posts as in the work of Qiu et al. [46]. This labeled data is a random sample of 293 posts from the discussion boards of CSN. Each post is annotated as positive or negative based on the sentiment expressed by the poster. There are 201 posts labeled as positive and 92 posts labeled as negative.

3.4.2 Experimental Setting

I experimented with various machine learning algorithms (Naive Bayes, Support Vector Machines, Logistic Regression, Bagging, Boosting, etc.) to conduct the classification experiments. I first built classification models on the two types of features (DI and DD) using supervised learning algorithms. Logistic regression was found to give the best classification performance with DI sentiment features and Naive Bayes Multinomial with the DD sentiment features. I used the Weka data mining toolkit [36] to train and test the supervised learning algorithms. I used my own implementation of co-training. To evaluate the performance of my classifiers, I used macro-averaged precision, recall and F-1 score, ROC area and accuracy. For a metric M (e.g., precision, recall, F-1 score and ROC area), macro-average M_{mav} is calculated by taking weighted average of M for both positive and negative classes for each fold and then taking the mean of weighted averages across all folds. For n-fold cross validation, M_{mav} is mathematically defined as follows:

$$M_{mav} = \frac{1}{n}\sum_{i=1}^{n}\frac{n_{+_i}M_{+_i} + n_{-_i}M_{-_i}}{n_{+_i} + n_{-_i}} \tag{3.1}$$

where n_{+_i} and n_{-_i} are the number of positive and negative posts in the test set in the i^{th} fold . M_{+_i} and M_{-_i} are the values of metric M for the positive and the negative classes, respectively, in the i^{th} fold. I used $n = 10$ in my experiments. I use F-1 score to compare performances of two classifiers. A naive baseline that classifies all the threads in the majority class will have a macro-averaged precision, recall and F-1 measure of 0.467, 0.684 and 0.555, respectively. For co-training, I used 2000 unlabeled instances in U, randomly sampled from all the unlabeled data, and 100 instances in S, sampled from U (see Algorithm 1). I experimented with different values for the number of iterations and the number of high confidence

unlabeled data added to the labeled data after each iteration (i.e., the value of p and n). I present these experiments in Section 3.4.4.2.

Model	**Pr.**	**Re.**	**F-1**	**ROC**	**Ac.**
Domain-dependent features (DD)					
BoW	0.714	0.724	0.717	0.786	0.723
$Uni + POS$	0.711	0.72	0.714	0.759	0.720
$Uni + Bi$	0.702	0.717	0.706	0.732	0.716
$Uni + Bi + POS$	0.711	0.72	0.714	0.718	0.720
Domain-independent features (DI)					
DI^{nl}	0.782	0.788	0.783	0.832	0.786
DI^{l}	0.79	0.795	0.792	0.812	0.792
Combining DD and DI features					
$FeatureComb(BoW + DI^{l})$	0.798	0.802	0.799	0.758	0.802
$Ensemble(BoW + DI^{l})$	0.785	0.740	0.786	0.828	0.788
Classification model proposed by Qiu et al. [46]					
Qiu et al. [46]	0.781	0.780	0.781	0.832	0.788
Co-training $(BoW + DI^{l})(p = 2, n = 1, K = 95)$					
Co-training	**0.851**	**0.85**	**0.85**	**0.858**	**0.849**

Table 3.3: Performance of different classification models.

3.4.3 Baselines

I use the following classification models as baselines:

1. **DD**: Classifiers trained on domain-dependent features (four classifiers corresponding to four types of features: BoW, $Uni + POS$, $Uni + Bi$, and $Uni + Bi + POS$).

2. **DI**: Classifiers trained on domain-independent features.

3. $featureComb$: Classifiers trained on the feature set created by combining the domain independent and the best performing domain dependent sentiment feature sets among the four listed above.

4. *Ensemble*: In this model, I train two classifiers C_I and C_D on the two views of the data (DD and DI) in a supervised setting. For a test instance, the final prediction is computed by taking the average of the predictions of the two classifiers. More precisely, for a test instance T, the final prediction is calculated as follows:

$$P_{ens}(+|T) = \frac{1}{2}(P_{C_I}(+|T) + P_{C_D}(+|T)) \tag{3.2}$$

where $P_{C_I}(+|T)$ and $P_{C_D}(+|T)$ are the probability estimates given by classifiers C_I and C_D, respectively, for T belonging to the positive class. $P_{ens}(-|T) = 1 - P_{ens}(+|T)$. For classification, I used a threshold of 0.5 on the model prediction.

5. **Qiu et al.**: Classifiers developed by Qiu et al. [46].

3.4.4 Experimental Results

In this section, I present the performance of the proposed co-training sentiment classification approach and compare it with the five baseline models.

3.4.4.1 Comparison of various classification models

Table III presents the results of comparing the proposed co-training approach with the different classification baselines described above. As can be seen in the table, the co-training approach outperforms all the other models with an F-1 score of 0.85. I performed experiments with different parameter settings for co-training, as explained in the next section. The first four rows of Table III show the results of the classification using DD features. As can be seen, BoW outperformed all the other DD features with respect to all the metrics. This is consistent with previous observations [47]. The next two rows in Table III show results of domain-independent features (DI) which outperform domain-dependent features. To see the impact of the last sentence on the sentiment classification, I first build a model using DI features extracted only from posts and not separately from their last

sentences [2] (denoted as DI^{nl}) and then I build the DI^{l} model in which I extract the DI features from posts and their last sentences separately. Including the last sentence improves the performance over the DI^{nl} model. In fact, DI^{l} model is the third best performing model (after co-training and $featureComb$) outperforming all the other baselines. For $featureComb$, I combined the best performing DD features (BoW) with the best performing DI features (DI^{l}). Combining the two types of features improves the performance over BoW features and the DI^{l} model. For $Ensemble$, I trained two classifiers on the best performing DD features (BoW) and the best performing DI features (DI^{l}) and combined the two classifiers using Equation 3.2. The performance of this model is worse than that obtained using both BoW and DI^{l} features. This observation suggests that the predictions of these two classifiers are quite different from each other. The conflicts between the two classifiers in terms of predictions of labels results in a lower accuracy of the combined ensemble model. However, in the co-training experiment, the two classifiers "guide" each other in the learning process, i.e., only instances predicted with high confidence in one class or the other by both classifiers are used to enrich the labeled training set. The conflicting instances (i.e., instances classified by the two classifiers in opposite classes) are discarded, since they can introduce noise to the labeled data. Finally, we have results of the comparison of my models with the model proposed by Qiu et al. [46]. As can be seen, all my models except those built on DD features outperform the Qiu et al. model.

3.4.4.2 Effect of number of iterations (K) and number of unlabeled instances added to the labeled set after each iteration (n, p)

Both, the number of iterations and the number of positive and negative instances added to the labeled data (after each iteration) have effects on the performance of co-training algorithm [58, 53, 61, 5]. To explore these effects, I conduct additional experiments. I plot the performance of the proposed co-training approach (in terms of F-1 score) as a function of the number of iterations K for different numbers of unlabeled instances added to the labeled data during the prediction step (n, p). Figure 3.2 shows this experiment. Here, p and n denote the number of *positive*

[2] That is I do not extract those features separately from the last sentence. I do not discard the last sentence and extract the features from the remaining sentences of the post.

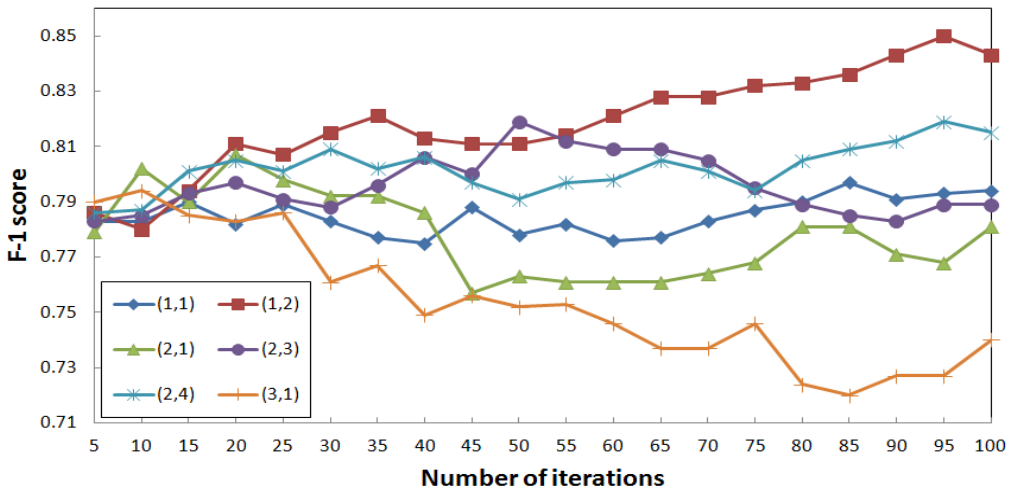

Figure 3.2: Co-training performance as a function of number of iterations (K) for different numbers of negative and positive instances (n, p) added from the unlabeled to the labeled data after each iteration.

and *negative* instances, respectively, added from the unlabeled data to the labeled data after each iteration. I experiment with twenty values of K, starting from 5 and going until 100 in steps of 5. That is, for a particular value of (n, p), I conduct 20 co-training experiments corresponding to the 20 values of K. We see that for all the settings (except $(n, p) = (3, 1)$), the proposed co-training approach outperforms the best performing baseline (of 0.799 F-1 score) at or after 20 iterations. We also note that the best performance of the proposed algorithm occurs at $K = 95$ and $(n, p) = (1, 2)$, which is same as that of the ratio of the number of negative and positive instances in the underlying training data. I, then, change (n, p) with respect to the underlying data distribution in the following ways:

(a) Altering the ratio of n and p such that $n >= p$: $(n, p) = (1, 1); (2, 1); (3, 1)$.

(b) Altering the ratio of n and p such that $n < p$: $(n, p) = (2, 3)$.

(c) Increasing n and p proportionately: $(n, p) = (2, 4)$.

From Figure 3.2, we see that for $(n, p) = (1, 1)$, the F-1 score remains almost constant with number of iterations. When we alter the ratio by increasing the number of negative instances with respect to positive instances $((n, p) = (2, 1); (3, 1))$, we see that the performance decreases with increasing the number of iterations. For

$(n, p) = (2, 1)$, we get the best F-1 at $K = 20$ after which the performance decreases. When the alteration in the ratio is more ($(n, p) = (3, 1)$), the decrease is higher (from 0.79 at $K = 5$ to 0.74 at $K = 100$). For the second setting ($(n, p) = (2, 3)$), we see that the performance increases with increasing the number of iterations until $K = 50$ with F-1 of 0.82 and decreases after that. For the third setting, the performance increases with K (from F-1 of 0.788 at $K = 5$ to 0.82 at $K = 95$) but the increase is not as high as with $(n, p) = (1, 2)$ (from 0.788 at $K = 5$ to 0.85 at $K = 95$). The proposed co-training approach performs the best when the underlying distribution of positive and negative instances (in the training data) is maintained. This observation is consistent with the setting used in the original co-training paper [58].

	Pr.	**Re.**	**F-1**
Postive class	0.894	0.886	0.89
Negative class	0.755	0.772	0.763
Overall	0.851	0.85	0.85
Best performing baseline ($FeatureComb(BoW + DI^l)$)	0.798	0.802	0.799

Table 3.4: Classification performance of the proposed model for positive and negative classes.

3.4.5 Performance on Individual Classes

Table 3.4 reports the performance of the best performing co-training model on the positive class and the negative class. We see that the performance on positive class is better than that on negative class. This behavior can be attributed to the unbalanced distribution of posts in positive and negative classes with positive class having significantly more instances than the negative class. Hence, the classifier learns more patterns for the majority (positive) class and performs better on it.

3.5 Chapter Summary

In this chapter, I performed sentiment classification of user posts of an online cancer support community using co-training, a semi-supervised learning algorithm. I utilized unlabeled posts to augment a small training data using co-training. I used

domain-specific and general information about sentiment expressions and combined them in the co-training setting. The experiments showed that co-training is an effective way to combine the two information sources with respect to sentiment classification performance. I, also, find that the last sentences of the posts play an important role in the sentiment classification.

Chapter 4

Identifying Emotional and Informational Support in Online Health Communities

4.1 Introduction

Increasingly more people turn to online health communities (OHCs) to seek social support during their illnesses [62, 42]. When people suffering from a serious disease such as cancer or AIDS *interact* with other people who have experienced similar medical conditions, they feel emotionally supported. In addition, through these interactions, people can obtain important information about the disease, e.g., about various medications, symptoms, and side-effects. Although authoritative health-related web sites contain the information they search for, obtaining this information directly from people in OHCs adds substantial value to it. Previous studies showed that obtaining social support in OHCs can help people feel better [43, 44, 42, 45, 46].

As a result of online interactions in OHCs, a huge volume of user-generated content exists today on various issues/problems related to specific diseases. This content comprises of important information such as people's experiences with diseases, recommendations and feedbacks about certain medications or medical procedures, and emotional support in the form of encouragement, sympathy, and success sto-

ries. Mining this content can prove to be very useful in obtaining crucial insights into community dynamics such as identifying dominant health issues or the effects of social support on community members, identifying influential members, as well as designing smart information retrieval systems for users.

In this study, I focus on an online cancer support community, the Cancer Survivors Network[1] (CSN) of the American Cancer Society. I analyze user messages of CSN to identify the two most important types of social support present in them: informational and emotional support [41]. Emotional support comprises of seeking or providing caring/concern, understanding, empathy, sympathy, encouragement, affirmation and validation. In contrast, informational support comprises of seeking or providing knowledge such as advice, referrals, and suggestions [40]. I further explore the relation between the type of support present in messages and users' influence in the community.

Identifying the type of support in user messages in an OHC can potentially be used in many important applications including the following:

1. **Identify influential members in OHCs**: Every community has a set of members who influence (a much larger set of) other members in the community. These members are called leaders or influential members. The attributes of a leader in a community depends upon the community's nature (QA, Twitter, OHC, forum, blogsite, etc.). For example, high activity may not be an indicator of high influence in the blogosphere [63] and high popularity does not necessarily imply influence in Twitter [64]. In OHCs, bringing positivity in the community and answering members' concerns effectively by posting messages that contain certain type of support (informational or emotional) may be an indicator of influence.

2. **Improve information search in OHCs**: Interactions in OHCs contain valuable information in the form of people's experiences, advice, referrals, pertaining to diseases, medications, side-effects, etc. Users embed this information often in messages containing other types of support, of which emotional support constitutes a major part. To efficiently search OHCs for this information, it must be separated from emotional support. Hence, identify-

[1]http://www.csn.cancer.org

ing the type of support in user messages can help improve search and retrieval in OHCs.

3. **Understand social relationships in OHCs**: Emotional support is one of the dimensions of social tie strength between members in a social network [65]. Previous studies have shown that members receiving emotional support in OHCs are more likely to remain in the community for a longer period of time as compared to members receiving informational support [66]. Identifying emotional and informational support can help understand the social dynamics of an OHC. For example, it may help determine if there is a correlation between the social tie strength of members and the type of support present in their interactions.

Hi X, I had a bilateral with radical on the right and prophylactic on the left. I think all you can do is gentle exercises to strengthen your back (yoga). There are also herbal painkillers that work well too. I just tolerate the pain and consider it a signal of my new limit and go down to rest. You want to talk, anytime! We are all there with you.

Table 4.1: A user message. Sentences in grey and black fonts are informational and emotional, respectively.

I model the task of identifying the two types of supports as a binary classification problem. Specifically, I classify each sentence in a user message as containing either emotional or informational support [2]. Table 4.1 shows a user message containing emotional and informational supports. I use several features computed from sentences of messages such as unigrams, part-of-speech tags, lexicon-based features and word patterns for the classification. After building the classification model, I predict the amounts of the two supports in all CSN messages and explore the following research question:

RQ: Do influential members of CSN post one of the two types of supports significantly more compared to regular members?

I analyze messages posted by regular members and messages posted by certain members, identified as *influential* by the CSN community managers and two staff

[2]Although a sentence may belong to both the classes, I did not find such cases in the data.

members who monitor the contents of the CSN on a full time basis, for the type of support (informational and emotional) present in them. Using the classification model, I calculate the amounts of the two supports posted by influential members and regular members and compare them across the two populations (For details, see Section 4.3.1).

Previous works on analyzing social support in OHCs have mainly been in the field of social science [67, 68, 69, 70, 42, 71, 72]. These works used manual techniques for labeling the type of support in user messages and hence, are limited to a small number of messages as compared to the real world data. In contrast, the current work builds machine learning classifiers that can automatically predict the type of support in messages. Also, to the best of my knowledge, there have been no reported works on analyzing the relationship between users' influence and the type of support present in their messages in OHCs. Next, I review related works.

4.2 Related Work

Many studies in social science have focused on analyzing social support in user messages of OHCs [73, 72, 70], finding impacts of social support on users [67, 68, 71, 69], identifying information needs of users in OHCs [74], etc. Among various types of social supports in OHCs, emotional support and informational support have received major attention. In this section, I first review some social science works on analyzing online social support, discuss works on identifying the type of social support, and, compare the current problem with **subjectivity analysis**. Finally, I will discuss works on identifying influential members in online communities.

4.2.1 Emotional and Information Support

LaCoursiere [62] presented an integrated theory conceptualizing online social support. She defined three channels through which online social support occurs: 1) *perceptual*: individual feeling the need of social support arising due to emotional states such as stress, etc, 2) *cognitive*: individual seeking information about certain medical entities such as procedures, medication, etc, 3) *transactional*: individual evaluating the received social support. In this work, these channels correspond to

emotional support and informational support. Høybye et al. [69] conducted a qualitative study to analyze the effects of online social support by interviewing women with breast cancer who use an online support group and find that the women were empowered by the exchanges of knowledge and experience within the online support group. Rodgers et al. [68] conducted a longitudinal content analysis of messages of participants in a breast cancer discussion board to analyze changes in affect/sentiment of the participants towards breast cancer and found that a positive shift in sentiment occurred over the period of time. Pfeil and Zaphiris [70] analyzed messages of SeniorNet forum to extract language patterns used to provide empathic support. Buis [71] studied messages in an online hospice support community and found that emotional interactions were more frequent that informational interactions. Han et al. [72] analyze the effects of *reception* of empathy (a type of social support) on cancer patients and showed that both expression and reception of empathy are responsible for attaining the benefits of empathy in online support groups. Budak and Agrawal [75] interviewed participants of group chats in Twitter and found that informational support is more important than emotional support in educational Twitter chats.

All the above works used manual methods of data preparation such as interviews with users of support groups, manual coding of messages to identify emotional and informational supports, etc and performed further qualitative and/or quantitative analyses based on that data. Since, manual methods have serious limitations in terms of scalability, the number of messages used for analysis in these studies is too small compared to the real world data which contains millions of messages. To address these limitations, I develop automatic methods for identifying the type of support in user messages in an online cancer support group using machine learning techniques. I develop a classifier that learns on a smaller set of labeled messages and makes predictions on a much larger set of messages with a very high accuracy.

A recent work by Wang et al. [66] is close to the current work. They used a linear regression model to predict the amount of informational and emotional supports present in messages of a cancer forum. For a test message, the trained model predicts the amount of the two supports on a scale of $1-7$. Since a message may contain both types of support, it is generally difficult for human annotators

to assess the amount of each support in an entire message on a particular scale for model training. In contrast, I label each sentence as belonging to either informational or emotional support class and identify the two types of support at sentence level in messages (using binary classification). Note that it is much easier and less ambiguous for a human annotator to identify the type of support present in a sentence (of a message) compared to giving a score to an entire message based on the amount of the two supports present in it.

4.2.2 Relationship with Subjectivity Analysis

Subjectivity analysis is an active area of research in computational linguistics. It essentially deals with separating subjective parts (e.g., expressing opinion, emotion, speculation and other private states of mind) from objective parts (presenting facts, verifiable information) of a text [7, 76, 77, 78, 79]. Though the current work has some relation with subjectivity analysis in the sense that both are text classification, there are important differences between the two problems. The two classes in subjectivity analysis (subjective and objective) are different from the two types of support that I identify. While emotional support is subjective in nature, informational support is not necessarily objective as it also contains opinions of users. Also, social support in OHCs encompasses several types of supports such as understanding, caring, concern, sympathy, empathy, knowledge about medications, etc. which are generally not provided by users in other sites such as product reviews, question-answering sites, etc. These differences make the two problems different in both the nature and the approaches that can be used to address them. For example, I use certain word patterns to identify sympathy and affirmation and use the presence of terms related to cancer medications, procedures and side-effects for computing features for classification. These features have not been used in subjectivity classification.

4.2.3 Identifying Influential Members

Every community has a set of members who influence (a much larger set of) other members in the community. These members are called leaders or influential members. Identifying influential members in online communities such as blogosphere,

twitter and Facebook has gained a lot of attention. Broadly, the methods used for identifying influential members in online communities or social networks can be divided into two categories: (i) content-based, and (ii) network-based. Content-based methods identify influential members by analyzing the content posted by members in the community. Network-based methods analyze the properties of the social network (of the community) with nodes representing community members or their posts. Since influence is subjective in nature, there is no single way to measure it. Goyal et al. [80] identified leaders in Yahoo movies social network by mining certain patterns in the network. They defined leaders to be the people whose actions affect the other people in their social circle to perform similar actions (e.g., rating a movie). Agarwal et al. [63] identified influential bloggers in a blot site based on certain properties (recognition, novelty, eloquence) of their blogs and the flow of influence of their blog posts in the network. Romero et al. [64] use popularity twitter users in terms of followers, and the influence of tweets in terms of propagation in the tweet network by retweeting for identifying influential twitter users. In the current work, I analyze the content of the messages posted by users in CSN for the type of support (emotional and informational) present in them to find out if influential members post one of the two supports significantly more than the other in their messages.

4.3 Problem Formulation

Online health communities provide social support to its members of which emotional and informational supports constitute the major part and have received major attention as compared to other supports such as companionship, community building, network support, etc. [40, 81, 82, 66, 70]. I focus on the two supports and follow their definitions as given by Bambina [40] in their study of social supports expressed in a cancer support group. They define *emotional* messages as the messages that have the following supports: caring/concern, understanding, empathy, sympathy, encouragement, affirmation and validation. *Informational support* is defined as providing advice, knowledge and referrals. Since a user message often contains a mixture of these supports, I identify the two supports at sentence level. Table 4.1 contains a user message with sentences marked with the type of support

in them. Specifically, given a sentence s, in a user message, I want to classify it into one of the two classes: emotional support or informational support. I use machine learning methods for classification. After training the classifier, I use it to predict the support in the sentences of user messages in CSN and address the research question outlined in Section 4.1. I present the details of the features used for classification in Section 4.3.2.

4.3.1 Research Question

To address the research question **(RQ)**, we need to compute the amounts of the two supports in the messages of regular and influential members and then compare the two amounts. Let u denote a user and M be the set of messages posted by her such that $M = \{m_1, m_2, ...m_p\}$ where p is the total number of messages in the set M. For a message $m_k \in M$, I compute its *emotional index*, $e_{uk} = n_{ek}/(n_k)$ where n_{ek} and n_k are the number of sentences containing emotional support and the total number of sentences in m_k. Since a sentence can belong to either emotional support or informational support class, informational index of m_k, $i_{uk} = 1 - e_{uk}$. The overall emotional index of u (e_u) is the average of the emotional indices of her messages: $e_u = \frac{1}{p}\sum_{k=1}^{p} e_{uk}$. The informational index of u, $i_u = 1 - e_u$. Since, the informational index can be derived from emotional index, I compute only emotional indices for all regular and influential members and compare them between the two user populations (regular and influential). I compute the emotional indices of regular members, E_R, and emotional indices of influential members, E_I. I compare the means of the two populations of emotional indices (μ_{Re} and μ_{Ie}) and test the null hypothesis (H_0) and the alternate hypothesis (H_1) as follows:
H_0: The two populations have equal means, i.e., $\mu_{Re} - \mu_{Ie} = 0$.
H_1: The two populations have significantly different means, i.e., $\mu_{Re} - \mu_{Ie} \neq 0$.

For one of the population indices to be significantly more than the other, we should have the null hypothesis rejected. I use one-sided t-test to conduct hypothesis testing and report the results in Section 4.4.6. Next, I discuss the features used in the classification.

4.3.2 Features for Classification

4.3.2.1 Words and POS tags

Words and their part-of-speech tags capture basic lexical properties of text and have been extensively used in text classification problems such as subjectivity classification and sentiment classification [5, 4, 83]. I use frequency of words and their POS tags in a sentence as features in the classification model.

4.3.2.2 Lexicon-based Features

Emotional support expresses caring, concern, sympathy, and other kinds of sentimental support whereas informational support provides knowledge about cancer medications, cancer reports, referrals, and other kinds of information [40]. Due to this difference in the nature of these supports, a sentence expressing emotional support is likely to contain emotional words which are subjective in nature and a sentence containing informational support is likely to have cancer-related keywords such as drug names, name of cancer procedures, etc. To capture this difference, I use frequencies of subjective words and cancer-related keywords as features. Specifically, I design five features to code frequencies of weak subjective words (**numWeak**), strong subjective words (**numStrong**), cancer drugs (**numDrug**), side-effects of cancer medications (**numSide**), and cancer procedures (**numProc**) respectively in a sentence. I use the subjectivity lexicon compiled from the MPQA corpus [32] to get weak and strong subjective words. I compile lexicon of cancer drugs [3], and CSN staff members helped get a list of side-effects and cancer procedures. Some of the side-effects of cancer medications are hair loss, neuropathy, fatigue, fibrosis, etc.

4.3.2.3 Linguistic Features

I analyzed user messages to find patterns that are expressive of emotional and informational support. I found that members, generally, use certain word patterns to express similar feelings. For example, to provide affirmation and sympathy, people use positive verbs such as *know, feel, understand, sense, support*, etc. in

[3] http://www.cancer.gov/cancertopics/druginfo/alphalist

patterns *<I $posVerb>* and *<I $aux $posVerb>*, where *$posVerb* is a positive verb and *$aux* is an auxiliary verb from the set {can, could, do, would, will, may}. Some people use *"We"* instead of *"I"* in their messages to provide support such as "**we understand** *what you are going through*". To take into account these cases, I use the same patterns of emotional support by replacing *"I "* with *"We"*. Hence, we get four patterns for emotional support. For providing informational support, people often use patterns such as *<You $advice>*, *<I $opinion>*, *<I $aux $opinion>* to provide advice and opinions. *$advice* is an auxilliary verb from the set {should, must, need, might} used to give advice, *$opinion* is an opinion verb from the set {recommend, advise, suggest, advocate, request}, and *$aux* is an auxilliary verb. People also give information about their experiences using patterns such as *<I too>*, *<I also>* and *<I $pastVerb>* to tell their own experiences related to similar problems as that of the support seeker where *$pastVerb* is a past tense verb such as *underwent, undergone, experienced, had, found*, etc. So, we get six patterns for informational support. I use the presence of these patterns as binary features. Specifically, I design two features (**IsEmPattern** and **IsInPattern**) to encode presence (1) or absence (0) of the two types of patterns.

For a sentence, I also use its number of words (**numWords**) and its type, question sentence (**IsQues**) and/or exclamatory sentence (**isExclaim**), as features. To identify question sentences, I see if a sentence starts with any of the 5W1H words *(what, why, who, when, where, how)* or words in the set {do, does, did} or ends with a question mark. For exclamatory sentences, I look for exclamation mark(s) at the end of the sentence.

4.4 Experiments

I now describe the data and the experimental setting, and present results.

4.4.1 Data Preparation

I use data from a popular online cancer support community, the Cancer Survivors' Network (CSN), developed and maintained by the American Cancer Society. CSN is an online community for cancer patients, cancer survivors, their families and

friends. Its features are similar to many online forums with dynamic interactive medium such as chat rooms, discussion boards, etc. Members of CSN post in discussion boards for seeking and sharing information about cancer related issues and for seeking and providing emotional support. To conduct experiments, I used user messages in the discussion threads of the Breast Cancer sub forum of CSN that were posted between June 2000 to June 2012. Breast cancer is the largest among all the sub-forums of CSN. A dataset of $250,868$ messages posted by 5516 users in $22,297$ discussion threads is used in this study.

To prepare the evaluation dataset for classification experiments, I randomly sampled 240 messages from 27 discussion threads. Since, the focus is on the messages that provide support, I do not consider messages posted by thread starters in discussion threads as they seek support. I took help of three human annotators to tag all the sentences of all the messages in one of the two support classes. First, two annotators tagged all the sentences. The percentage agreement between them was 89%. For the remaining 11% sentences, majority vote was taken with the help of the third annotator. Following this tagging scheme, I obtained a total of 1066 sentences with 390 sentences in the informational support class and 676 sentences in the emotional support class. In many cases, members only write a few words, e.g., see you, bye, or their names at the end of a message. To deal with these situations, I filter out sentences that have less than four words.

4.4.2 Experimental Protocol

I experimented with various machine learning algorithms (Naive Bayes, Support Vector Machines, Logistic Regression, Bagging, Boosting, etc.) to conduct the classification experiments. Naive Bayes Multinomial gave the best performance with words & POS tags features, logistic regression with lexicon-based features and AdaBoost (with Decision Stump as the weak learner) with linguistic features. For combining the models built on the three types of features, I used the following three methods:

1. **Feature combination:** Classification model built on the feature set generated by combining the three types of features. It is denoted by **All**. I use Multinomial Naive Bayes for this model.

2. **Average confidence:** Ensemble of the three classifiers built on the three types of features respectively. The final confidence of the ensemble is calculated by taking average of the confidences outputted by the three classifiers. It is denoted by **AllAvgConf**.

3. **Highest confidence:** Similar to the **AllAvgConf** model but the final prediction of the ensemble is taken as the prediction of the most confident classifier of the three classifiers. More precisely, the prediction for an instance is given by the classifier that returns the maximum prediction confidence for one class or the other. It is denoted by **AllMostConf**.

I used Weka data mining toolkit [36] to conduct classification experiments. To evaluate the performance of the classifiers, I used macro-averaged precision, recall and F-1 score. I use F-1 score to compare performances of two classifiers and used 10-fold cross validation. A naive baseline that classifies all the instances in the majority class will have a macro-averaged precision, recall and F-1 score of 0.402, 0.634 and 0.492, respectively.

4.4.3 Classification Results

Table 4.2 presents the results of the support classification experiments. The table reports precision, recall and F-1 score of different classification models for the individual classes and the overall result. Words & POS tags are the best performing features followed by lexicon-based features and linguistic features. Further, combining all the features (model denoted as "**All**") improves the performance over individual feature types for both classes. We see that **AllMostConf** model is the best performing of all the models, particularly outperforming **All** and **AllAvgConf** models. This observation suggests that the three classifiers built on the three features types have different knowledge. For some instances, a particular classifier is more confident than the rest while for other instances, other classifiers are more confident. Hence, we see that taking prediction of the most confident classifier gives the best performance. It is interesting to note that combining the three classifiers' knowledge in this fashion is more effective than simply combining all the three types of features and train a single classifier on the combined feature set. We also note that all the models have better performance for the emotional

Model	Precision	Recall	F-1
Emotional support class			
Words & POS tags	0.855	0.858	0.857
Lexicon-based features	0.722	0.836	0.775
Linguistic features	0.698	0.837	0.761
All	0.862	0.861	0.862
AllAvgConf	0.848	0.893	0.87
AllMostConf	0.851	0.911	**0.88**
Informational support class			
Words & POS tags	0.753	0.749	0.751
Lexicon-based features	0.608	0.441	0.511
Linguistic features	0.569	0.372	0.45
All	0.76	0.762	0.761
AllAvgConf	0.797	0.723	0.758
AllMostConf	0.825	0.723	**0.77**
Overall			
Words & POS tags	0.818	0.818	0.818
Lexicon-based features	0.68	0.691	0.678
Linguistic features	0.651	0.667	0.647
All	0.825	0.825	0.825
AllAvgConf	0.829	0.830	0.83
AllMostConf	0.841	0.842	**0.84**

Table 4.2: Classification results.

support class than for the informational support class. This can be caused by the fact that there are significantly more number of instances in the former class and, hence, more patterns to learn for the class.

4.4.4 Informative Features

Next, I study the importance of individual features by measuring their chi-squared statistic with respect to the class variable. I, first, study the word features and then present rankings of the other types of features. Figure 4.1 shows a cloud of top 26 most informative words. The size of a word is proportional to its chi-squared statistic, i.e., bigger a word, more informative it is. We see that cancer specific keywords such as herceptin, tamoxifen, chemo, dose, stage, etc and words conveying emotions such as good, hope, glad, pain, hugs, etc are highly informative

for the support classification. Since, chi-square method gives feature ranking for the class variable and not for individual classes, I compute word rankings for individual classes using $tf - idf$ scores of words. Specifically, for a term t and a class c, I compute the term frequency of t by counting its number of occurrences in the instances (sentences) belonging to c and multiply the term frequency with the inverse document frequency of t (calculated from the entire corpus) to get the $tf - idf$ score of t for c. Using this method, I calculated $tf - idf$ scores for all the words and ranked them according to their scores for the two classes. Figure 4.2 shows top ten $tf - idf$ ranked keywords for the two classes. We see that cancer-related keywords and words expressing emotions are among the top ten most informative words for the informational and the emotional support classes respectively. We also note that most of the top ten words for the two classes in Figure 4.2 are in the word cloud of the top 26 words computed using chi-squared method except "keep" for the emotional support class and "after", "first", "because" and "cancer" for the informational class. These words have semantic relationships with the classes. For example, "keep" is often used by support providers in phrases such as "**keep** *you in prayers*", "*may god* **keep** *you in good health*", etc to provide emotional support and "after" and "first" are used in the context of providing one's own experience related to cancer procedures, medications, etc such as "**After** *my* **first** *chemo, I did not feel light*".

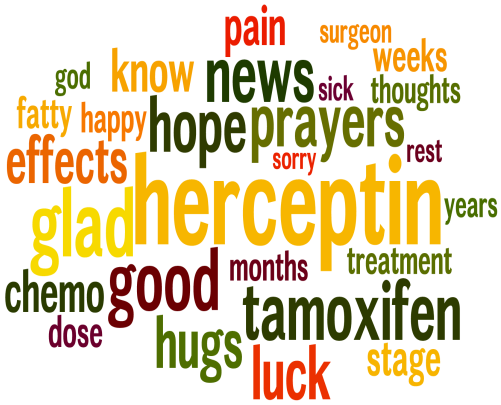

Figure 4.1: Top 26 words ranked by Chi-squared test.

Emotional support	**Informational support**
good	chemo
know	after
glad	radiation
news	first
hope	herceptin
keep	treatment
prayers	tamoxifen
luck	cancer
hugs	because
better	pain

Figure 4.2: Top ten words for the two classes ranked using tf-idf scheme.

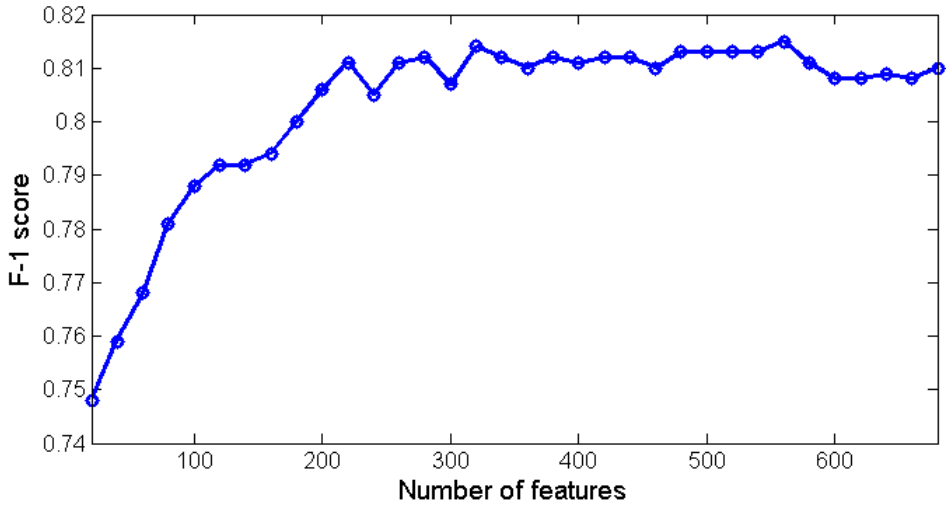

Figure 4.3: Plot showing classification performance with top informative features.

I, now, discuss the ranking of non-word features: POS tags, lexicon-based and linguistic features. The chi-squared ranking for the lexicon-based and linguistic features is as follows: numStrong > numWords > isExclaim > numDrug > numSide > numProc > isInPattern > isEmPattern > numWeak > isQues. The features on the right side of > have higher rank than those on the left side. We see that the number of strong subjective words in a sentence is the most informative feature followed by number of words in a sentence. Among cancer-related terms, drug names are more informative than side-effects and cancer procedures. Also, informational support word patterns are more informative than word patterns capturing emotional support. It is interesting to note that isQues is the least informative feature, maybe due to the fact that, while providing support, people generally do not ask questions. The top 5 most informative POS tags are: cardinal number (CD) followed by singular noun (NN), participle verb (VBN), past tense verb (VBD) and preposition (IN).

Further, I plot the classification performance by building classifier on top k most informative features at a time. I vary k from 20 to 680 (total number of features is 690) in steps of 20. Figure 4.3 reports the performance. We note that as we increase the number of features, the performance improves with the best F-1 score of 0.815 at number of features equal to 560. However, we see that feature selection does not improve the performance over the model built using all the

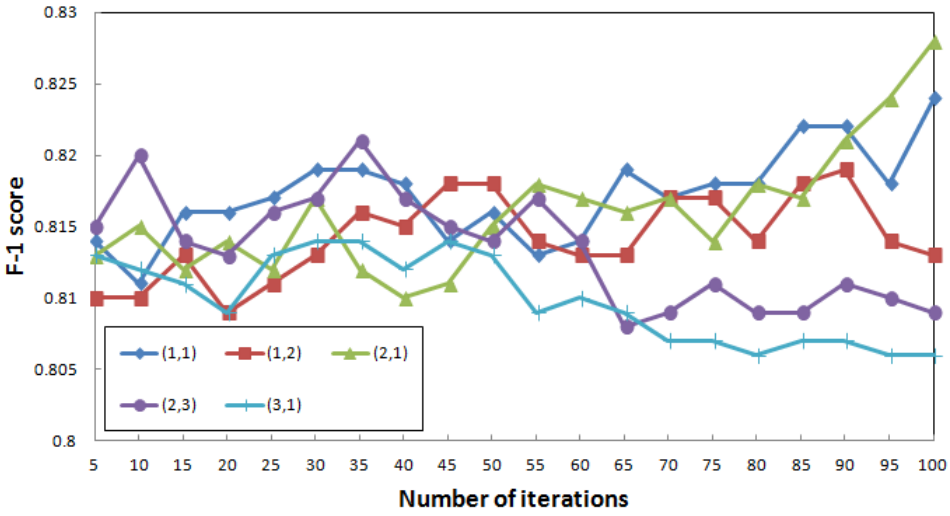

Figure 4.4: Co-training performance as a function of number of iterations (K) for different numbers of emotional and informational support instances (e, i) added from the unlabeled to the labeled data after each iteration.

features ("All" model in Table 4.2).

4.4.5 Co-training for Classification

To see if the classification performance can be improved using information contained in the unlabeled data, I tried co-training. I used words & POS tags as one view and lexicon and linguistic features combined as the other view for an instance (sentence), as required by co-training. Co-training is explained in detail in Section 3.3. For getting unlabeled data, I used 9848 sentences from randomly sampled 200 threads. Figure 4.4 shows co-training performance with varying number of iterations (5 to 100) and different number of emotional and informational support examples (e, i) added after each iteration. As we see, the best performance is attained for $(e, i) = (2, 1)$ (which is same as the distribution in the training data) and number of iterations equal to 100, with F-1 score of 0.828. However, we note that the AllMostConf ensemble model with F-1 score of 0.84 (Table 4.2) outperforms the co-training model.

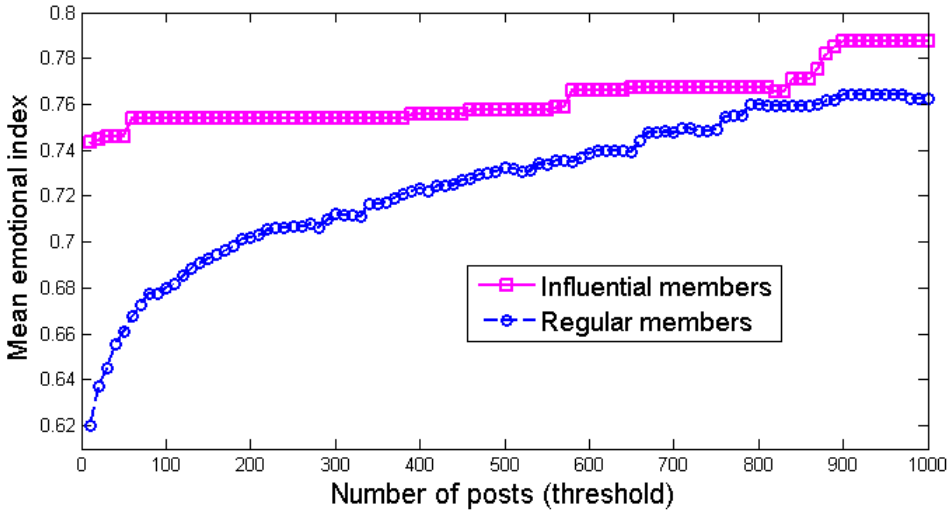

Figure 4.5: Plot showing the change in mean emotional indices of influential members (pink) and regular members (blue) with the threshold on the number of messages posted by them.

4.4.6 Influence versus Support type

CSN managers provided a list of 62 influential members (IMs) for the breast cancer forum. IMs posted a total of 340, 147 sentences in 85, 244 messages and regular members posted 825, 651 sentences in 165, 624 messages in the breast cancer forum. As described in Section 4.3, I conduct statistical hypothesis testing on the two populations of emotional indices (regular members and IMs) to understand if there is a significant difference in their posting behaviors in terms of providing one of the two supports more often than the other. To test the hypothesis, I conducted one sided t-test on the two populations. I found that the mean of emotional indices of IMs (0.713) is significantly larger than that of the regular members (0.542). We also note that the posting behavior of regular members in CSN follows a power law distribution with most of the members posting very few messages ($mode = 1$, $median = 2$, $mean = 30$) and only a few members posting very many messages. To verify that this behavior does not have impacts on the hypothesis testing, I conducted three more t-tests between the two populations using a threshold on the number of messages that a member has posted. I used three threshold values on the number of messages: 1, 2, and 30 (as mode, median and mean values). For all the three t-tests, the null hypothesis was rejected at p-value< 0.001, suggesting

that IMs posted significantly more emotional support than regular members. The values of Mean Emotional Indices corresponding to the three thresholds are 0.715, 0.719 and 0.746 for influential members and 0.564, 0.581 and 0.646 for regular members respectively.

In my analysis, an interesting behavior I observed was that, as I increased the threshold, the mean of emotional indices also increased. To further investigate this finding, I plotted the means of emotional indices of regular members and IMs as the function of the threshold on the number of messages posted by them. I increased the threshold from 10 to 1000 in steps of 10. Figure 4.5 reports the finding. We see that the mean of emotional indices of regular members increase with the threshold suggesting that members who are more active in the forum post more emotional support as compared to the less active members. We also see that the mean of emotional indices of IMs is higher than that of regular members for all the thresholds.

Above observations suggest that IMs significantly differ from regular members in terms of the amount of emotional support they provide in their messages. To see if these differences can be helpful in identifying IMs in OHCs, I conducted another set of experiments. I ranked CSN members on the basis of their overall emotional indices (OEI) and compute the number of IMs among the top K members in the ranked list. I vary K from 10 to 100 in steps of 10. Table 4.3 presents the number of IMs at different K. I also compare the performance of the proposed metric (OEI) with a recently proposed method metric called **Influential Responding Replies (IRR)** [84]. IRR is defined as the responding reply whose sentiment is aligned with the thread starter's change of sentiment probabilities from the starting post to the first self-reply. More precisely, an IRR satisfies the following two conditions:

- Temporally, an IRR is between the first post of the thread and the first self-reply of the thread starter.
- If the sentiment change of the thread starter from the first post to the first self-reply is positive then the sentiment of an IRR is also positive. Similarly, if the sentiment change of the thread starter from the first post to the first self-reply is negative then the sentiment of an IRR is also negative.

For more details about IRR, please refer to the original work. To calculate the

sentiment of user messages, I use the sentiment classifier which is explained in Chapter 3.

Metric used for ranking a user	Top10	Top20	Top30	Top40	Top50	Top60	Top70	Top80	Top90	Top100
IRR	8	15	19	25	30	33	35	38	41	41
OEI	6	11	11	13	15	16	18	19	20	23

Table 4.3: Number of influential users in top k users ranked by their total IRRs and total emotional index.

From Table 4.3, we note that OEI does help identifying influential users with 60% precision in top 10. For higher ranks, performance deteriorates with 23% precision in top 100. Also, we see that IRR outperforms OEI. One of the reasons for this can be the fact that IRR aligns with the change of sentiment which is a strong indicator of influence. It would be interesting to see how OEI performance changes if only emotional messages that change sentiment of thread initiator are considered in its calculation. Also, a hybrid model that combines IRR and OEI would be worth looking into. I leave analysis of such cases for future work.

4.5 Chapter Summary

In this chapter, I identified two types of social support, emotional and informational, provided by users in their messages of an online cancer support community using machine learning classification models. I used ensemble of classifiers built on three types of features. The model achieved strong results with over 80% accuracy. Using the trained model, I predicted the type of support in all the user messages in CSN and found that influential members provide significantly more emotional support to the community as compared to regular members.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

This dissertation focused on extracting important information from large amounts of user generated data in online forums using automatic machine learning techniques. For that, I addressed three problems: (i) subjectivity analysis of online forum threads, (ii) sentiment analysis of messages in an online cancer forum, and (iii) identifying the type of social support in messages of an online cancer forum. The specific contributions towards solving these problems are summarized below:

- **Subjectivity Analysis of Online Forum Threads and its Use in Thread Retrieval:** Internet users search online forum archives for discussion threads relevant to their information needs. Often, finding relevant threads becomes difficult due to a large number of threads discussing lexically similar topics but differing in the type of information they contain (e.g., opinions, facts, emotions). Hence, forum search facilities need to take into account the match between users' *intent* and the type of information contained in threads in addition to the lexical match between user queries and threads. I proposed a machine learning framework for matching user intent with the type of information in threads and used it to improve thread retrieval. I developed a binary classifier that predicts whether a thread's topic is subjective (seeking opinions, emotions, other private states) or non-subjective (seeking factual information) by taking inference from several novel thread-specific

features. I use the classifier to predict subjectivity of all the corpus threads. Finally, I incorporated the subjectivity information of threads along with manual subjectivity labels for a set of user queries in a probabilistic thread retrieval model and showed that retrieval performance improves by matching subjectivity of user queries and threads.

- **Sentiment Analysis of an Online Cancer Support Community Using Co-training:** Identifying sentiments expressed by members in an online health community can be helpful in understanding the community and its features, e.g., dominant health issues, emotional impacts of interactions on members, finding influential members, etc. I performed sentiment classification of user posts in an online cancer support community (Cancer Survivors Network). I used a small amount of labeled data (prepared by a previous research) to provide initial supervision to the classification model and then used a semi-supervised machine-learning algorithm, co-training, which uses information contained in unlabeled data, to perform sentiment classification. I used domain-specific and generic sentiment features as the two views of user posts (as required in co-training). I showed that using the unlabeled data significantly improves sentiment classification performance over supervised models that were built using only the labeled data.

- **Identifying Emotional and Informational Support in Online Health Communities:** Interactions in Online health communities contain messages providing and seeking social support of which informational and emotional support constitute major part. I identified these two types of social support in user messages of an online cancer support community (CSN) at sentence level using classification. I used three types of features and got the best results by using ensemble of the three classifiers built on the three individual feature types. The model achieved strong results with over 80% accuracy. Using the trained model, I predicted the type of support in all the user messages in CSN and explored whether influential members provide a particular support significantly more than the other as compared to the regular members. I found that influential members provide significantly more emotional support to the community as compared to regular members. The finding can

be helpful in identifying properties of influential members in online health communities.

5.2 Future Work

- For subjectivity analysis of online forum threads, I assume that a thread has a single topic of discussion specified in its title and initial post. However, it is common for users (participating in a thread) to talk about topics that are different from the thread topic. In such cases of topic drifts, subjectivity of thread topics would not be same throughout a thread. Detecting topic drift is a challenging task. It would be interesting to investigate if subjectivity analysis at post level can be used to detect topic drifts. By identifying change in subjectivity across posts in a thread, drifts from subjective to non-subjective topics and vice versa can potentially be detected. Further, post level subjectivity analysis can help developing post retrieval models.

- I performed sentiment classification of user messages of Cancer Survivor Networks (CSN). CSN users talk about various medications, side-effects, social support, and other cancer related topics. It would be interesting to analyze opinions (positive/negative/neutral) of CSN users with respect to various cancer medications and procedures. A summary of opinions of different users on various cancer medications and procedures can be generated and presented to users. This will enable users to get an idea of the overall experiences of other users with those medications and procedures and will save a lot of their time used in browsing through relevant threads.

- Sentiment classification of user messages in OHCs can also be used by community managers and nurse navigators for making necessary interventions. For example, with consent, a nurse may observe the sentiment and the change of sentiment in posts by a patient to get an advance warning of the deterioration of the sentiment of the patient. This may, in turn, result in interventions that may save the life of the patient or improve the patients quality of life.

- Interactions in OHCs contain valuable information in the form of people's experiences, advice, referrals, pertaining to diseases, medications, side-effects,

etc. Users often embed this information in messages containing other types of support, of which emotional support constitutes a major part. It would be interesting to use the binary support classifier for separating this information from emotional support to improve search in OHCs.

Bibliography

[1] Seo, J., W. B. Croft, and D. A. Smith (2009) "Online community search using thread structure," in *CIKM 2009*, ACM, New York, NY, USA, pp. 1907–1910.

[2] Bhatia, S. and P. Mitra (2010) "Adopting Inference Networks for Online Thread Retrieval," in *AAAI 2010, Atlanta, Georgia, USA, July 11-15*, pp. 1300–1305.

[3] Duan, H. and C. Zhai (2011) "Exploiting thread structures to improve smoothing of language models for forum post retrieval," *Advances in Information Retrieval*, pp. 350–361.

[4] Yu, H. and V. Hatzivassiloglou (2003) "Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, pp. 129–136.
URL http://dx.doi.org/10.3115/1119355.1119372

[5] Li, B., Y. Liu, and E. Agichtein (2008) "CoCQA: co-training over questions and answers with an application to predicting question subjectivity orientation," in *Proceedings of The Conference on Empirical Methods in Natural Language Processing*, pp. 937–946.

[6] Aikawa, N., T. Sakai, and H. Yamana (2011) "Community QA Question Classification: Is the Asker Looking for Subjective Answers or Not?" *IPSJ Online Transactions*, **4**, pp. 160–168.

[7] Wiebe, J., R. Bruce, and T. O'Hara (1999) "Development and use of a gold-standard data set for subjectivity classifications," in *ACL*, ACL, pp. 246–253.

[8] Bruce, R. and J. Wiebe (1999) "Recognizing subjectivity: a case study in manual tagging," *Natural Language Engineering*, **5**(2), pp. 187–205.

[9] Wiebe, J. and E. Riloff (2005) "Creating subjective and objective sentence classifiers from unannotated texts," *Computational Linguistics and Intelligent Text Processing*, pp. 486–497.

[10] Su, F. and K. Markert (2008) "From words to senses: a case study of subjectivity recognition," in *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 825–832.
URL http://dl.acm.org/citation.cfm?id=1599081.1599185

[11] Mihalcea, R., C. Banea, and J. Wiebe (2007) "Learning Multilingual Subjective Language via Cross-Lingual Projections," in *Proceedings of the 45th annual meeting of the Association for Computational Linguistics*, pp. 976–983.

[12] Banea, C., R. Mihalcea, J. Wiebe, and S. Hassan (2008) "Multilingual subjectivity analysis using machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 127–135.
URL http://dl.acm.org/citation.cfm?id=1613715.1613734

[13] Banea, C., R. Mihalcea, and J. Wiebe (2010) "Multilingual subjectivity: are more languages better?" in *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 28–36.
URL http://dl.acm.org/citation.cfm?id=1873781.1873785

[14] Mukund, S. and R. K. Srihari (2010) "A vector space model for subjectivity classification in Urdu aided by co-training," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 860–868.
URL http://dl.acm.org/citation.cfm?id=1944566.1944665

[15] Liu, B. (2010) "Sentiment analysis and subjectivity," *Handbook of natural language processing*, **2**, pp. 627–666.

[16] Hu, M. and B. Liu (2004) "Mining and summarizing customer reviews," in *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pp. 168–177.

[17] Ly, D. K., K. Sugiyama, Z. Lin, and M.-Y. Kan (2011) "Product review summarization from a deeper perspective," in *Proceedings of ACM/IEEE Joint Conference on Digital Libraries*, ACM, pp. 311–314.

[18] Bhatia, S., P. Biyani, and P. Mitra (2012) "Classifying User Messages for Managing Web Forum Data," in *Proceedings of the 15th International Workshop on the Web and Databases*, pp. 13–18.

[19] Li, B., Y. Liu, A. Ram, E. Garcia, and E. Agichtein (2008) "Exploring question subjectivity prediction in community QA," in *SIGIR*, ACM, pp. 735–736.

[20] Gurevych, I., D. Bernhard, K. Ignatova, and C. Toprak (2009) "Educational question answering based on social media content," in *Proc. of the 14th International Conf. on Artificial Intelligence in Education*, pp. 133–140.

[21] Stoyanov, V., C. Cardie, and J. Wiebe (2005) "Multi-perspective question answering using the OpQA corpus," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 923–930.

[22] Somasundaran, S., T. Wilson, J. Wiebe, and V. Stoyanov (2007) "QA with attitude: Exploiting opinion type analysis for improving question answering in on-line discussions and the news," in *ICWSM*.

[23] Li, F., Y. Tang, M. Huang, and X. Zhu (2009) "Answering opinion questions with random walks on graphs," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, Association for Computational Linguistics, pp. 737–745.

[24] Moghaddam, S. and M. Ester (2011) "AQA: aspect-based opinion question answering," in *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, IEEE, pp. 89–96.

[25] Oh, J.-H., K. Torisawa, C. Hashimoto, T. Kawada, S. De Saeger, J. Kazama, and Y. Wang (2012) "Why question answering using sentiment analysis and word classes," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, pp. 368–378.

[26] Hassan, A., V. Qazvinian, and D. R. Radev (2010) "What's with the Attitude? Identifying Sentences with Attitude in Online Discussions," in *EMNLP 2010*, ACL, pp. 1245–1255.

[27] Zhai, Z., B. Liu, L. Zhang, H. Xu, and P. Jia (2011) "Identifying evaluative sentences in online discussions," in *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

[28] Walker, M. A., P. Anand, R. Abbott, J. E. F. Tree, C. Martell, and J. King (2012) "That is your evidence?: Classifying stance in online political debate," *Decision Support Systems*, **53(4)**(4), pp. 719–729.

[29] Abbott, R., M. Walker, P. Anand, J. E. Fox Tree, R. Bowmani, and J. King (2011) "How can you say such things?!?: Recognizing disagreement in informal political argument," in *Proceedings of the Workshop on Languages in Social Media*, Association for Computational Linguistics, pp. 2–11.

[30] Jeong, M., C.-Y. Lin, and G. G. Lee (2009) "Semi-supervised speech act recognition in emails and forums," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1250–1259.
URL http://dl.acm.org/citation.cfm?id=1699648.1699671

[31] Joty, S., G. Carenini, and C.-Y. Lin (2011) "Unsupervised modeling of dialog acts in asynchronous conversations," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, AAAI Press, pp. 1807–1813.
URL http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-303

[32] Wiebe, J., T. Wilson, and C. Cardie (2005) "Annotating Expressions of Opinions and Emotions in Language," *Language Resources and Evaluation*, **39(2)**, pp. 165–210.

[33] Thelwall, M., K. Buckley, and G. Paltoglou (2012) "Sentiment strength detection for the social web," *J. Am. Soc. Inf. Sci. Technol.*, **63**(1), pp. 163–173.
URL http://dx.doi.org/10.1002/asi.21662

[34] Ponte, J. M. and W. B. Croft (1998) "A Language Modeling Approach to Information Retrieval," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 275–281.

[35] Zhai, Chengxiang and Lafferty, John (2001) "A study of smoothing methods for language models applied to Ad Hoc information retrieval," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 334–342.

[36] Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten (2009) "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, **11**(1), pp. 10–18.

[37] Porter, M. F. (1980) "An algorithm for suffix stripping," *Program*, **14(3)**, pp. 130–137.

[38] Manning, C., P. Raghavan, and H. Schütze (2008) *Introduction to information retrieval*, vol. 1, Cambridge University Press Cambridge.

[39] Petrie, K. J. and J. Weinman (1997) *Perceptions of health and illness: Current research and applications*, vol. 1, Taylor & Francis.

[40] Bambina, A. D. (2007) *Online Social Support: The Interplay of Social Networks and Computer-Mediated Communication*, Cambria Press.

[41] Davison, K. P., J. W. Pennebaker, and S. S. Dickerson (2000) "Who talks? The social psychology of illness support groups." *American Psychologist*, **55**(2), p. 205.

[42] Beaudoin, C. E. and C.-C. Tao (2007) "Benefiting from social capital in online support groups: An empirical study of cancer patients," *CyberPsychology & Behavior*, **10**(4), pp. 587–590.

[43] Dunkel-Schetter, C. (1984) "Social support and cancer: Findings based on patient interviews and their implications," *Journal of Social Issues*, **40**(4), pp. 77–98.

[44] Maloney-Krichmar, D. and J. Preece (2005) "A multilevel analysis of sociability, usability, and community dynamics in an online health community," *TOCHI*, **12**(2), pp. 201–232.

[45] Vilhauer, R. P. (2009) "Perceived benefits of online support groups for women with metastatic breast cancer," *Women & health*, **49**(5), pp. 381–404.

[46] Qiu, B., K. Zhao, P. Mitra, D. Wu, C. Caragea, J. Yen, G. Greer, and K. Portier (2011) "Get Online Support, Feel Better – Sentiment Analysis and Dynamics in an Online Cancer Survivor Community," in *Proceedings of the 3rd IEEE International Conference on Social Computing*, pp. 274–281.

[47] Pang, B., L. Lee, and S. Vaithyanathan (2002) "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the 2002 conference on Empirical methods in natural language processing*, EMNLP '02, Stroudsburg, PA, USA, pp. 79–86.

[48] Stavrianou, A., J. Velcin, and J.-H. Chauchat (2009) "Definition and measures of an opinion model for mining forums," in *Proceedings of 2009 International Conference on Advances in Social Network Analysis and Mining*, IEEE, pp. 188–193.

[49] Neri, F., C. Aliprandi, F. Capeci, M. Cuadros, and T. By (2012) "Sentiment Analysis on Social Media," in *ASONAM' 12*, pp. 919–926.

[50] Jiang, L., M. Yu, M. Zhou, X. Liu, and T. Zhao (2011) "Target-dependent twitter sentiment classification," in *Proceedings of The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 151–160.

[51] Pang, B. and L. Lee (2004) "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *ACL '04*, p. 271.

[52] McDonald, R., K. Hannan, T. Neylon, M. Wells, and J. Reynar (2007) "Structured models for fine-to-coarse sentiment analysis," in *Proceedings of The 45th Annual Meeting of the Association for Computational Linguistics*, pp. 432–439.

[53] Wan, X. (2009) "Co-training for cross-lingual sentiment classification," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing: Volume 1-Volume 1*, Association for Computational Linguistics, pp. 235–243.

[54] Li, S., S. Ju, G. Zhou, and X. Li (2012) "Active Learning for Imbalanced Sentiment Classification," in *Proceedings of The Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 139–148.

[55] Kumar Pal, J. and A. Saha (2010) "Identifying Themes in Social Media and Detecting Sentiments," in *ASONAM*, pp. 452–457.

[56] Bermingham, A., M. Conway, L. McInerney, N. O"Hare, and A. Smeaton (2009) "Combining Social Network Analysis and Sentiment Analysis to Explore the Potential for Online Radicalisation," in *ASONAM '09*, pp. 231–236.

[57] Turney, P. D. (2002) "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 417–424.

[58] Blum, A. and T. Mitchell (1998) "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th annual conference on Computational learning theory*, COLT' 98, ACM, New York, NY, USA, pp. 92–100. URL http://doi.acm.org/10.1145/279943.279962

[59] Balcan, M.-F., A. Blum, and K. Yang (2004) "Co-Training and Expansion: Towards Bridging Theory and Practice." in *NIPS*, vol. 17, pp. 89–96.

[60] Wang, W. and Z.-H. Zhou (2007) "Analyzing co-training style algorithms," in *ECML*, Springer, pp. 454–465.

[61] Nigam, K. and R. Ghani (2000) "Analyzing the Effectiveness and Applicability of Co-training." in *Proceedings of the 2000 ACM CIKM International Conference on Information and Knowledge Management*, pp. 86–93.

[62] LaCoursiere, S. P. (2001) "A theory of online social support," *Advances in Nursing Science*, **24**(1), pp. 60–77.

[63] Agarwal, N., H. Liu, L. Tang, and P. S. Yu (2008) "Identifying the influential bloggers in a community," in *Proceedings of the 2008 international conference on web search and data mining*, ACM, pp. 207–218.

[64] Romero, D. M., W. Galuba, S. Asur, and B. A. Huberman (2011) "Influence and passivity in social media," in *Machine learning and knowledge discovery in databases*, Springer, pp. 18–33.

[65] Gilbert, E. and K. Karahalios (2009) "Predicting tie strength with social media," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, pp. 211–220.

[66] Wang, Y.-C., R. Kraut, and J. M. Levine (2012) "To stay or leave?: the relationship of emotional and informational support to commitment in online health support groups," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, ACM, pp. 833–842.

[67] Eriksson, E. and S. Lauri (2000) "Informational and emotional support for cancer patients relatives," *European Journal of Cancer Care*, **9**(1), pp. 8–15.

[68] Rodgers, S. and Q. Chen (2005) "Internet community group participation: Psychosocial benefits for women with breast cancer," *Journal of Computer-Mediated Communication*, **10**(4), pp. 00–00.

[69] Høybye, M. T., C. Johansen, and T. Tjørnhøj-Thomsen (2005) "Online interaction. Effects of storytelling in an internet breast cancer support group," *Psycho-Oncology*, **14**(3), pp. 211–220.

[70] Pfeil, U. and P. Zaphiris (2007) "Patterns of empathy in online communication," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, pp. 919–928.

[71] Buis, L. R. (2008) "Emotional and informational support messages in an online hospice support community," *Computers Informatics Nursing*, **26**(6), pp. 358–367.

[72] Han, J. Y., D. V. Shah, E. Kim, K. Namkoong, S.-Y. Lee, T. J. Moon, R. Cleland, Q. L. Bu, F. M. McTavish, and D. H. Gustafson (2011) "Empathic exchanges in online cancer support groups: distinguishing message expression and reception effects," *Health communication*, **26**(2), pp. 185–197.

[73] Coursaris, C. K. and M. Liu (2009) "An analysis of social support exchanges in online HIV/AIDS self-help groups," *Computers in Human Behavior*, **25**(4), pp. 911–918.

[74] Rozmovits, L. and S. Ziebland (2004) "What do patients with prostate or breast cancer want from an Internet site? A qualitative study of information needs," *Patient education and counseling*, **53**(1), pp. 57–64.

[75] Budak, C. and R. Agrawal (2013) "On participation in group chats on twitter," in *Proceedings of the 22nd international conference on World Wide Web*, International World Wide Web Conferences Steering Committee, pp. 165–176.

[76] Biyani, P., S. Bhatia, C. Caragea, and P. Mitra (2012) "Thread Specific Features are Helpful for Identifying Subjectivity Orientation of Online Forum Threads." in *Proceedings of the 24th International Conference on Computational Linguistics*, pp. 295–310.

[77] ——— (2014) "Using non-lexical features for identifying factual and opinionative threads in online forums," *Knowledge-Based Systems*.

[78] Biyani, P., C. Caragea, A. Singh, and P. Mitra (2012) "I want what i need!: analyzing subjectivity of online forum threads," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 2495–2498.

[79] Biyani, P., C. Caragea, and P. Mitra (2013) "Predicting subjectivity orientation of online forum threads," in *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 109–120.

[80] Goyal, A., F. Bonchi, and L. V. Lakshmanan (2008) "Discovering leaders from community actions," in *Proceedings of the 17th ACM conference on Information and knowledge management*, ACM, pp. 499–508.

[81] Meier, A., E. J. Lyons, G. Frydman, M. Forlenza, and B. K. Rimer (2007) "How cancer survivors provide support on cancer-related Internet mailing lists," *Journal of Medical Internet Research*, **9**(2).

[82] Himle, D. P., S. Jayaratne, and P. Thyness (1991) "Buffering effects of four social support types on burnout among social workers," in *Social Work Research and Abstracts*, vol. 27, Oxford University Press, pp. 22–27.

[83] Biyani, P., C. Caragea, P. Mitra, C. Zhou, J. Yen, G. E. Greer, and K. Portier (2013) "Co-training over domain-independent and domain-dependent features for sentiment analysis of an online cancer support community," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 413–417.

[84] Zhao, K., J. Yen, G. Greer, B. Qiu, P. Mitra, and K. Portier (2014) "Finding influential users of online health communities: a new metric based on sentiment influence," *Journal of the American Medical Informatics Association*, http://jamia.bmj.com/content/early/2014/01/21/amiajnl-2013-002282.full.pdf+html.
URL http://jamia.bmj.com/content/early/2014/01/21/amiajnl-2013-002282.abstract

Vita

Prakhar Biyani

Prakhar Biyani completed his Bachelor of Technology (B.Tech) in Electrical Engineering from Indian Institute of Technology (IIT) Roorkee, India, in May 2010. Thereafter, he joined the PhD program in the College of Information Sciences and Technology at The Pennsylvania State University. During his PhD, he worked with the Cancer Informatics Initiative (CANI) group at Penn State and developed algorithms for analyzing sentiment and identifying the type of social support present in user messages of an online cancer support group. During his PhD, he worked as an intern at IBM Research Bangalore and Yahoo Labs, Sunnyvale. His research interests lie in the field of computational linguistics, applied machine learning and information retrieval.

Selected Publications

1. Sumit Bhatia, Prakhar Biyani and Prasenjit Mitra. Identification of Dialogue Acts in User Messages posted in Online Forums and its Application in Improving Thread Retrieval. Journal of the American Society for Information Science and Technology (to appear) (2014).

2. Prakhar Biyani, Sumit Bhatia, Cornelia Caragea, and Prasenjit Mitra. Using Non-lexical Features For Identifying Factual and Opinionative Threads in Online Forums. Elsevier Knowledge-Based Systems (2014) .

3. Prakhar Biyani, Cornelia Caragea, Prasenjit Mitra and John Yen. Identifying Emotional and Informational Support in Online Health Communities. In: 25th International Conference on Computational Linguistics, 2014 (to appear).

4. Prakhar Biyani, Cornelia Caragea, Prasenjit Mitra, John Yen, Greta E Greer and Kenneth Portier.Co-training Over Domain-dependent and Domain-independent Features for Sentiment Analysis of an Online Cancer Support Community. In: proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2013: 413-417.

5. Prakhar Biyani, Sumit Bhatia, Cornelia Caragea, and Prasenjit Mitra. Thread Specific Features Are Helpful For Identifying Subjectivity Orientation of Online Forum Threads. In: Proceedings of the 24th International Conference on Computational Linguistics, 2012: 295-310.

6. Prakhar Biyani, Cornelia Caragea, Amit K. Singh, Prasenjit Mitra. I want what I need! Analyzing Subjectivity of Online Forum Threads. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, 2012: 2495-2498.