Extracting Problem and Resolution Information from Online Discussion Forums

Preethi Raghavan [*, 1], Rose Catherine[2], Shajith Ikbal[2], Nanda Kambhatla[2], Debapriyo Majumdar[2]

[1]Dept. of Computer Science and Engineering , The Ohio State University, USA,
[2]IBM Research - India, Bangalore
raghavap@cse.ohio-state.edu, {rosecatherinek, shajmoha, kambhatla, debapriyo}@in.ibm.com

Abstract

The ability to obtain quick solutions to problems is an important requirement in many practical applications such as help desks and technical support. In this paper, we describe an approach that will enable easy identification of potential solutions to a given problem. The proposed approach involves extraction of useful problem- and resolution-related information from online discussion forums. We specifically focus on identifying important parts of a discussion thread by classifying message posts in the thread as problem- or resolution-related. We cast this problem as a sequence labeling task and train a discriminative Conditional Random Field for supervised learning. Results are presented from classification experiments done at a sentence as well as phrase-level. The structure and information flow pattern in a typical discussion thread helps generate effective features for classification. We also discuss the effect of inducing different features on the precision and recall of the sequence labeling task. Sentence level classification with feature induction yielded F1 scores of 66% for problem related information and 58.8% for resolution-related information, which is a significant improvement over our baseline scores.

1 Introduction

Social media and user-generated content have grown rapidly over the past decade. A key characteristic of online community-generated content includes conversational media, where users create new content as well as comment on content generated by others. Examples of conversational media include blogs, social networking sites, online discussion forums and any other website that offers an opportunity for the user to share their knowledge and familiarity with a product or experience.

Online discussion forums are web-based communities that allow users to share ideas, post problems, comment on posts by other users and obtain feedback. This often leads to development of active communities of enthusiastic contributors such as technical troubleshooting forums like SAP forums[1], Ubuntu forums[2], Microsoft TechNet support forum[3] etc. Such discussion forums are replete with user-generated content consisting of multiple discussion threads that span across many pages.

Consider the information flow pattern in a technical troubleshooting forum discussion thread shown in Figure 1. We can observe that a discussion thread starts with a message containing the problem description with some background information, followed by a series of replies with various suggestions towards resolving the problem. The flow of information is mostly sequential, with multiple responses to the original question including responses where new questions are posed and answered. Discussion threads may also be hijacked by users with a similar problem, who interrupt the thread and start posting their queries. Content-wise, messages in the thread may contain various ideas that may or may not help solve the problem, as well as other information such as links to other threads. The thread may conclude with the message author acknowledging successful resolution of the problem.

Thousands of such archived discussion threads with problem and resolution information are a storehouse of useful information for several applications. However, the unstructured nature of message posts that are characterized by issues of relevance, comprehensibility and subjectivity, make it challenging to retrieve relevant threads of interest. A traditional keyword search to identify potentially useful threads tends to be inadequate as many distinguishing characteristics of discussion forums are ignored by traditional information retrieval techniques. Searching over the discussion forum archive by making use of Boolean queries, proximity analysis and text relevance, results in a long list of possibly relevant discussion threads. However, this search engine approach of retrieving multiple results forces the user to browse multiple discussion threads. Moreover, useful messages may be hidden deep in the dis-

* This work was done while Preethi was an intern at IBM Research Labs -India

16th International Conference on Management of Data
COMAD 2010, Nagpur, India, December 9–12, 2010

[1]http://forums.sdn.sap.com
[2]ubuntuforums.org
[3]http://technet.microsoft.com

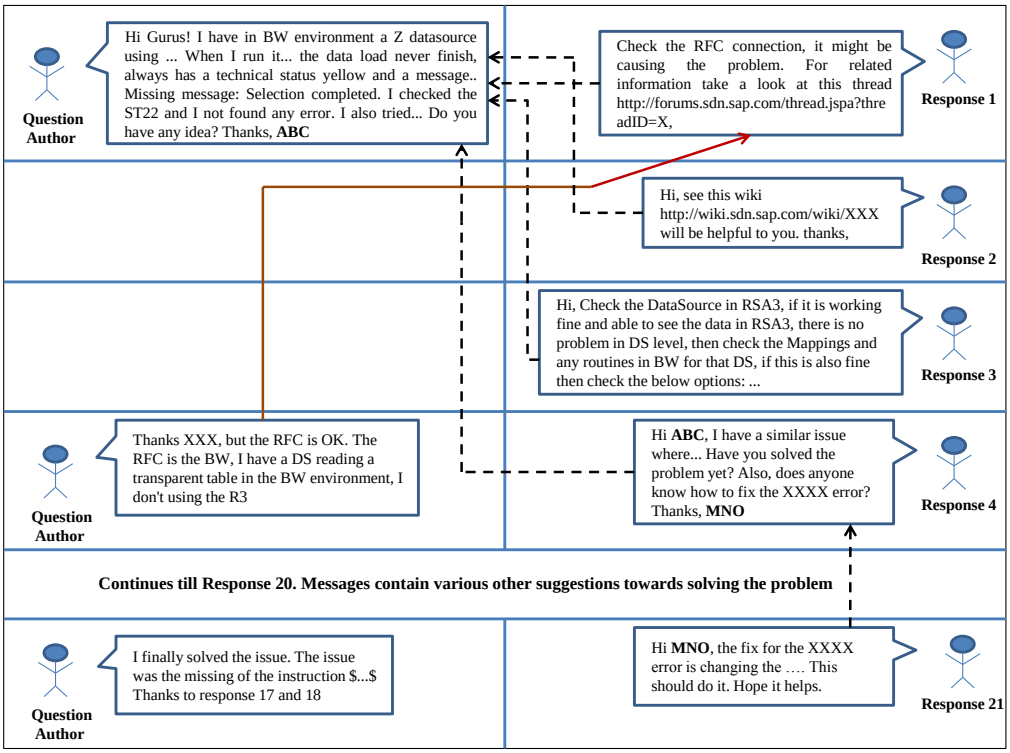

Figure 1: Example of the structure and the information flow pattern in a typical discussion thread

cussion threads, as seen in Figure 1. Besides searching for relevant threads, identifying parts of the thread that have relevant information is of tremendous benefit to the retrieval task.

Identifying and extracting problems and their corresponding steps that solved the problem, helps structure the discussion thread. Information retrieval over such structured discussion forum archive will be more effective, as it not only improves the accuracy of identifying the most relevant discussion thread, but also provides the exact resolution steps used to solve the problem whenever available. The ability to retrieve exact resolution steps for incoming problems that are similar to ones in the problem archive serves as an important asset in technical support. The problem resolution time can be optimized by providing help desk personnel easy access to solution steps of similar problems solved in the past. The solutions steps could be extracted from the help desk archive or from online discussion forums dedicated to the same domain.

In this paper, we investigate extraction of problems and resolution steps from discussion threads from an archive of unstructured discussion forum data. We address this problem by first identifying important parts of the thread that describe the problem and corresponding resolution related information. The problem is modeled as a sequence labeling task using Conditional Random Fields (CRFs) for supervised learning. We exploit distinguishing characteristics specific to the structure of discussion threads and the nature of data in message posts to generate features for the learning task.

2 Related Work

Public online communities like discussion forums, mailing lists and social networking sites have been the focus of several data mining studies. There has been a notable amount of work in knowledge extraction from discussion forums and mailing lists to build question-answering and summarization systems. Online forums and blogs have also been recognized as fertile ground for mining product discussion [7]. A system for mining online discussion for the purposes of monitoring popular opinion about brands or products is presented in [8]. Classification of message posts based on relevance, originality and other forum-specific features is used to automatically identify high quality posts in large discussion boards [20]. Researchers have also exploited the hierarchical thread structures in community sites to help retrieval in online forums and email archives [16]. While they focus on improving search in community sites like discussion forums by exploiting thread structure, we leverage forum-specific characteristics to help enable identification of solution steps for a given problem.

In [5], the authors propose a general framework that applies linear chain CRFs to detect answers and extract their context from discussion forum threads on TripAdvisor. Their work is motivated by the need to highlight important content, make searching easier by better orga-

nization of threads as well as to enrich the knowledge base of community-based question and answering such as Yahoo! Answers. They assume the question-answer pairs have been identified in a forum thread using the sequential patterns based classification method to detect questions and a graph based propagation method to detect answers for questions in the same thread as detailed in [4]. They also extend the basic model to 2D CRFs to model the dependency between contiguous questions in a forum thread for context and answer identification. Context information is utilized in answer detection by capturing the dependency between contexts and answers using the Skip-chain CRF model.

On a similar note, our work is motivated by the need to highlight important information from across the multiple message posts in each discussion thread, thus making information access much easier and the overall search experience effective. We specifically focus on classifying parts of the messages in the thread as both question and resolution-related or as information that is not relevant and to be ignored.

There are also several models proposed for thread retrieval in online message boards [6] [16]. This previous work on retrieval in online forums has focused on the discussion thread as the primary unit of retrieval. In the past, researchers have also experimented with classifying threads from web user forums in order to improve information access to problems and solutions described in the threads [1]. They classify threads based on the amount of problem related information, general discussion, whether the problem was finally resolved and completeness of the thread. In contrast, we work on extracting useful parts of the discussion thread.

Email threads found in mailing lists are similar to discussion forum threads, however the nature of data found in discussion forums is far more heterogeneous. This is because any user can post and reply while email has a fixed set of participants. Previous work in email thread information extraction includes [18] who present a method to detect question-answer pairs in an email conversation for the task of email summarization. The problem of extracting signature and reply lines in email messages using features in sequential (CRF, CPerceptron) as well as non-sequential (SVM, Naive Bayes) methods is addressed in [3]. Previous work [2] also utilizes thread information derived from annotated message posts to create a graph based on labels and analyze interaction patterns in a mathematical problem-solving scenario.

There has also been considerable interest in knowledge extraction from problem ticket archives to automate and optimize the ticket resolution process and workflow. TroubleMiner [13] can automatically organize trouble tickets into a hierarchy based on the problem raised in the ticket. However, it does not investigate segmenting and labeling parts of the ticket such as resolution, question etc. Another system, EasyTicket [17] examines the routing sequence of the ticket and uses a Markov model to capture the likelihood that a ticket is transferred to a group, given the past group transfer information and recommends the best routing for a new ticket. They also predict parameters such as possibility of escalation, maximum severity during creation, value of severity etc. during the life-cycle of an active ticket by reconstructing the evolution of each ticket from a corpus of closed tickets. Thus they demonstrate that machine learning can be successfully applied to optimize the required resources for solving network failures as managed by Trouble Ticket Systems, contributing to a new and automated approach to a process which has been traditionally solved by means of expert knowledge. Previously, CRFs have been applied to automatically structure problem tickets by viewing them as unstructured data and labeling each line with tags such as problem, resolution step etc [21].
In this paper, we present results on extraction of problem resolution information from the SAP community network discussion forum[1] using CRFs.

CRFs are probabilistic discriminative models for segmenting and labeling sequence data [10]. They predict the probability of a label sequence conditioned on the input and can capture long-range dependencies in the data and avoid the need for restrictive independence assumptions. Linear-chain CRFs have been shown to perform well for information extraction and other language modeling tasks due to their ability to capture arbitrary, overlapping features of the input in a Markov model. CRFs have previously been used for several natural language processing tasks such as named entity recognition, information extraction and chunking. Examples include POS tagging [10]and extracting headers from research papers [14].

The nature of messages found in discussion threads is distinctly different from that of a problem ticket or mailing lists. Threads in a discussion forum are heterogeneous and often fragmented by topic and by aspects of temporality, pace and sequence, frequency, and duration of discussion as contributions build over time [9]. We propose an approach to highlight important parts of a discussion thread and classify them as problem related or resolution related. This acts as a step towards generating a comprehensive summary of resolution steps for a problem found in a discussion thread.

3 Problem Description

Our work is motivated by the need to improve information access from a discussion forum archive by extracting problems and associated resolution steps from each discussion thread. Easy access to solution steps for similar problems in the archive is of tremendous benefit to technical support groups and help desks. In order to enable this, the specific task addressed in this paper is classification of parts of discussion threads found in online discussion forums as problem- or resolution-related. In doing so, we take advantage of certain structural characteristics

[1] http://forums.sdn.sap.com

unique to such discussion forums. Online forums are typically organized hierarchically with several high-level topical forum categories, which are split into finer grained categories. Each of these contains many threads, collections of user-contributed messages. In this section, we describe certain unique characteristics of discussion forums and proceed to outline the challenges in extracting problem- and resolution-related information from discussion threads.

3.1 Online Discussion Forums

A discussion forum is a place where participants can engage in text-based conversation organized into topic-based discussion threads. In technical discussion or troubleshooting forums, users discuss various technical problems, problem related details, potential solutions to the problems, and usefulness of various solutions suggested. A typical technical discussion forum contains several threads with each thread corresponding to some discussion about a single problem or a topic. Each thread contains multiple posts from various authors, with each post typically corresponding to a sub-component of a discussion. Some of the sub-components are problem statement, requesting clarification, providing clarification, providing problem related information, solution suggestion, solution related clarification, solution related information, solution suggestion feedback, and statement of similar problems. Each discussion is typically initiated by an author facing a problem and seeking potential solutions for it from the community members and domain experts.

In a discussion thread, all the authors try to solve the problem systematically in a collaborative fashion. There is usually an organized flow in the way the discussion is taken forward by the involved authors. Essentially, participants are engaged in conversation but they can take time to reflect on others' messages and to compose their own replies carefully. One example of the discussion flow typically observed in threads is follows.

1. An author creates a new thread by posting the technical problem he/she faced seeking potential solution from people who might be experts in the domain and/or might be familiar with the problem. Besides the actual problem, this post may also contain some background information on how the author encountered the problem, the problem environment and what steps he/she has already tried in order to fix the problem.
2. Looking at this problem post, a second author from the community (possibly a domain expert) responds with a follow-up post asking further technical details about the problem faced in order to get a better understanding of the problem.
3. The first author responds to it by generating a follow-up post with further technical details of the problem. Besides adding details to the original problem, he might also ask new questions in the context of the original problem environment.
4. The second author in response posts a potential solution to the original problem and also inquires about the new questions posted by the original message author.
5. As seen in Figure 1, a third author might hijack the thread by stating he has a similar problem with different errors and go on to describe his problem in detail.
6. A fourth author looking at the discussion so far generates a new post suggesting his/her own experience/opinion/solution.
7. The first author tries out the proposed solution and respond with a post saying whether his/her problem is solved or not.

Figure 2 shows one example discussion thread from public SAP community network discussion forum (in this figure, the discussion thread is sanitized to mask sensitive information).

As described previously in the introduction, each discussion thread comes with varying data quality, multiple levels of noise, troll message posts and unstructured informal language. We proceed to highlight in detail, the various challenges and benefits associated with information extraction in the online discussion forum domain.

3.2 Challenges and Benefits

One major challenge in automatically extracting useful problem-resolution information from discussion threads is in processing the human generated noisy text data. This is because many a times, the core useful information is hidden inside a bulk of non-important, heterogeneous, and some times irrelevant, text data of varying quality. On the other hand, the structure and manner of data flow in discussion threads offers various benefits and clues in facilitating effective extraction of useful information.

By examining the information flow in a typical discussion thread, we can conclude that there tends to be a systematic and sequential flow of information in the thread. The thread usually begins with the problem description, followed by various suggestions to solve the problem. Thus, by normalizing the position of the posts within the thread, we can predict parts of the thread that are more likely to contain problem or resolution-related information. The sequential flow of discussions in a thread also provides a great deal of contextual information when extracting useful information from a message post.

In addition, a message post generally begins with a greeting, followed by the description of the problem or resolution and end with note of thanks. This structure of a message post can also be leveraged in identifying important parts of the message. Moreover, discussion threads also have additional structured information including message author user names, time of posting and message title indicating which post the author has replied to in the thread that could potentially be utilized for efficient extraction. However, in this paper, we are not using such meta-information for extraction.

Since discussion forums primarily center around a specific domain, with discussion threads on topics in that domain, it is also possible to leverage a domain specific ontology or dictionary, to help identify parts of threads with useful information. For instance, we could leverage the Linux Kernel Glossary[4] or the SAP Glossary[5] to generate features for classification based on the presence or absence of these glossary terms, when extracting information from each of these discussion forums respectively.

In the next section, we outline our approach to extract problem and resolution related information.

3.3 Problem Formulation

We approach the problem of extracting problem and resolution-related information from discussion forums by studying the basic structure of the threads and noting their general composition and characteristics. The formal description of the problem is as below.

A discussion forum thread T, is composed of message posts M_1 through M_n where n is the length of the discussion thread. This can be represented as,

$$T =< M_1, M_2, M_3, M_4, M_5, ...M_n >$$

Each message post M_i where i takes values in the range of 1 to n, and is generally composed of a greeting, message description and a note of thanks. The part of the message that is important with respect to extraction of problem and resolution-related information is the message description. The greetings may help determine which post the message author is replying to in the discussion thread. Thus, a message post can be represented as M_i, where,

$$M_i =< m_1, m_2, m_3, m_4, m_5...m_j >$$

where m_j is a sentence and the values of j are in the range of 1 to the total number of meaningful sentences that add up to form the message. Of these meaningful sentences, only some are actually relevant to the problem or resolution. Thus, the problem or resolution-related information is a subset of the sentences in M_i.

Now, let M_i' be the set that is composed of all the problem or resolution-related information in a message post, such that $M_i' \subset M_i$ and $m_k \in M_i'$ is the sentence that has potentially relevant and useful problem or resolution-related information.

Also, every sentence in a message post is made up of multiple text snippets or ngrams, of which some maybe relevant to the problem or resolution and some may not. This can be defined as,

$$m_k =< mp_1, mp_2, mp_3, mp_4, mp_5...mp_q >$$

where q is the total number of ngrams of a predetermined length in m_k. Out of all these parts, only some may be relevant to the problem or resolution. Thus we have, $m_k' \subset m_k$, and $mp_r \in m_k'$ is the text snippet that has relevant or useful information.

Our task can now be defined as follows:

Sentence-level classification:
For each T, find every m_k that belongs to M_i' and classify it as problem- or resolution-related. In most cases, each m_k that belongs to a M_i' will be either problem- or resolution-related. However, there could be instances where M_i' is a combination of both problem- and resolution-related sentences m_k.

Phrase-level classification:
For each T, find every mp_r that belongs to m_k' and classify it as problem- or resolution-related.

We leverage contextual information resulting from specific discussion flows within discussion threads for extracting useful information from them. The sequential order of message posts in a thread, along with the fact that, in most cases, a message post is dependent on the previous message post allows us to create effective features for classification. In fact, the availability of contextual information is key to modeling the problem as a sequence labeling task, where the aim is to identify sections of useful information in the thread, in the context of other information that is present in the thread. We now proceed to describe the CRF model and its application to classify as problem or resolution-related, each m_k in case of sentence-level classification, and each mp_r in case of phrase-level classification.

4 Sequence Labeling using Conditional Random Fields

There are several established techniques to perform sequence labeling of text data. In this paper, we propose to use a supervised approach using CRFs for sequence labeling of the discussion threads. CRFs are discriminative models, basically an extension of logistic regression and a special case of log-linear models. They compute probability of a label sequence given an observation sequence, assuming that the current label (state) depends only upon the previous label (state) and the observation, as given below:

$$P(Y|X,W) = \frac{exp(\sum_j \sum_k w_j f_j(y_{k-1}, y_k, X)}{Z(X))} \quad (1)$$

where $Y = \{y_1, y_2, ..., y_m\}$ denote the label sequence and $X = \{x_1, x_2, ..., x_n\}$ denote the input sequence, $f_j(.)$ denote j^{th} feature function over its arguments and w_j denote the weight for the j^{th} feature function. $Z(X)$ is a normalization factor over all label sequences, defined as follows:

$$Z(X) = \sum_{y \in Y^T} exp(\sum_j \sum_k w_j f_j(y_{k-1}, y_k, X))$$

where Y^T is the set of all label sequences.
Inference to find the most likely state sequence given the observation sequence, given as below, can be performed

[4]http://kernelnewbies.org/KernelGlossary
[5]http://help.sap.com/saphelp/glossary

Sentence in the thread	True Label	Collapsed Label	Predicted Label
Hi,	GREETING	IGNORE	IGNORE
I run the mdm workflow (5.5) plugin in MDM and Visio professional 2003.	RELATED INFO	PROBLEM	PROBLEM
When opening the workflow the next time the MDM workflow stencil doesn't appear.	PROBLEM QUERY	PROBLEM	PROBLEM
X. X. XXXX	MSG AUTHOR	IGNORE	IGNORE
Hi,	GREETING	IGNORE	IGNORE
Install your Workflow Engine i.e. WorkflowInstall Ver setup and restart your machine	RESOLUTION SUGGESTION	IGNORE	IGNORE
Regards, YYY YYYYY	GREETING	IGNORE	IGNORE
Hi	GREETING	IGNORE	IGNORE
For the first time you need to enable macro settings in VISIO and enable checkboxes for trusted publications.	RESOLUTION SUGGESTION	RESOLUTION STEP	RESOLUTION STEP
best regards, ZZZZ	GREETING	IGNORE	IGNORE
Hi,	GREETING	IGNORE	IGNORE
thanks for the answer. I've reinstalled visio and the plugin locally, but no result.	PROBLEM	PROBLEM	IGNORE
Do I also need to restart the MDM server for the changes to take effect?	RELATED INFO	PROBLEM	PROBLEM
X. X. XXXX	MSG AUTHOR	IGNORE	IGNORE
Hi Again	GREETING	IGNORE	IGNORE
Listing the steps to be followed	RESOLUTION START	IGNORE	RESOLUTION STEP
1) Open Visio and go to the Tools > Macros > Security. The 'Security Level' tab should be set to 'Low' and in the 'Trusted Publishers' tab the 'Trust access to Visual Basic Project' should be checked	RESOLUTION STEP	RESOLUTION STEP	RESOLUTION STEP
2) In Visio, click Tools->Options.	RESOLUTION STEP	RESOLUTION STEP	RESOLUTION STEP
In the Options dialog, click Security tab. In the Security tab, check the Enable Automation Events button	RESOLUTION STEP	RESOLUTION STEP	RESOLUTION STEP
Once this has been done you will need to close the Visio and then launch it again via the Data Manager	RESOLUTION STEP	RESOLUTION STEP	RESOLUTION STEP
regards, ZZZZ	GREETING	IGNORE	IGNORE
Thanks for the advice	IGNORE	IGNORE	IGNORE
I've tried it including macro settings in MS visio, but I still can't get it to run. Can I have missed something else?	RELATED INFO	PROBLEM	PROBLEM
X. X. XXXX	MSG AUTHOR	IGNORE	IGNORE
Hi,	GREETING	IGNORE	IGNORE
Hope you have installed MDMWorkflowInstall Ver from the same installable. MDM Server and workflow engine should be of same version.	RELATED INFO	IGNORE	IGNORE
Regards, YYY YYYYY	GREETING	IGNORE	IGNORE

Figure 2: Example of a typical thread in a forum, showing the manual annotations at a sentence-level, the corresponding collapsed labels and the labels predicted by the CRF model.

using the Viterbi algorithm:

$$Y^* = argmax_Y P(Y|X, W) \quad (2)$$

CRFs have advantage that they do not require modeling effort for the observation, unlike other generative models for sequence labeling task such as hidden Markov model (HMM) [15]. Another key advantage is that there is no constraint that various feature components and observation should be independent of each other, thus providing flexibility to include wide variety of arbitrary number of non-independent features computed from observation for use along with observation, during labeling task.

Few of the additional features that we additionally extracted from observation for use along with words based features are as follows.

1. Part of speech (POS) tags as extracted from the text of observation to include additional information about the grammatical structure and word category of the sentences, in addition to the plain set of words.

2. Normalized position (NP) of posts within threads: Index of every post within thread is normalized with the total length of the corresponding thread and those normalized length are bucketed into one of the 3 features namely BEGIN, MIDDLE, and END. Each sentence in the post is added with the normalized position feature of the corresponding post. We believe this would help in learning and utilizing some trends as observed in the training data such as problem description would most likely appear in a post at the top of the thread, and resolution suggestion typically appear in post at later part of the thread.

4.1 CRF Training

For the CRF training, we used a manually annotated set of discussion forum threads in order to generate training data. We perform classification experiments on data annotated both at a sentence as well as phrase-level. The annotations done at various levels and granularities allow us to analyze the performance of CRF on different types of discussion forum labeling. In the first case, we performed sentence-level labeling of threads with only 3 labels repre-

senting useful/non-useful part of the text namely:

- PROBLEM: The actual problem being addressed in that thread. E.g.: As soon as we added SSL encryption it did not work anymore
- RESOLUTION STEP: Steps leading to problem solution. E.g.: Connect to port 504 for instance number 00
- IGNORE: Remaining part of text which is not relevant to the problem or resolution.

We also performed a sentence-level labeling with a richer set of labels. This not only the labels representing useful/non-useful part of information, but included additional labels characterizing various other aspects of a typical discussion thread. The aim here is to aid the CRFs in accurately modeling the data with richer contextual information. Some of the additional labels used in the richer set of labels are:

- PROBLEM RELATED INFO: Problem description/ related error information. E.g.: Here is what I get in the dispatcher log.
- PROBLEM END: Request for help usually at the end of problem post. E.g.: Any suggestions? Any ideas?
- RESOLUTION SUGGESTION: Suggestion towards solving the problem; but did not solve the problem.

Few other labels such as: SIMILAR PROBLEM (mention of a similar problem), PROBLEM FOLLOWUP (discussion of what happened after trial of a resolution suggession), MESSAGE AUTHOR and GREETING (usually at the beginning and end of every message post).

Experiments are also done with phrase-level annotations with the above set of labels. The aim here is to help CRFs capture the most distinguishing features representing a particular label. Figures 2 and 3 show examples of sentence-level labeling (with richer and reduced label sets) and phrase-level labeling.

With the above explained labeled training data we generated following set of CRF models:

- With richer label set (17 labels) at sentence-level and using only words as occurring in the thread as the features.
- With reduced label set (3 labels) at sentence-level and using only words as occurring in the thread as features.
- With reduced label set (3 labels) at sentence-level and using POS tags in addition to words as features.
- With reduced label set (3 labels) at sentence-level and using NP features (thread level normalized position information) in addition to words as features.

Phrase in the thread	True Label	Predicted Label
Hello Experts,	IGNORE	IGNORE
Am implementing a BAdI to	PROBLEM	IGNORE
achieve some customer enhancement for XD01 Transaction	PROBLEM	IGNORE
I need to confirm to customer that after the implementation and before implementation	PROBLEM	PROBLEM
what is the response time of the system	PROBLEM	PROBLEM
Response time BEFORE BAdI Implementation Response time AFTER BAdI Implementation	PROBLEM	IGNORE
Where can i get this.	PROBLEM	IGNORE
Help me in this regard	IGNORE	IGNORE
Best Regards	IGNORE	IGNORE
VVVV	IGNORE	IGNORE
Hi VVVV,	IGNORE	IGNORE
Please	IGNORE	IGNORE
have a look on this thread	RESOLUTION STEP	RESOLUTION STEP
it is helpful to	IGNORE	IGNORE
resolve the query you were facing.Hoping it resolves quickly. Link:Thread	IGNORE	PROBLEM
Have a best day ahead.	IGNORE	PROBLEM
Use Transaction ST05.	RESOLUTION STEP	IGNORE
UU UUUU	IGNORE	IGNORE

Figure 3: Example of a thread in the forum data, showing annotations at a phrase-level and the labels predicted by the CRF model.

- With reduced label set (3 labels) at sentence-level and using POS tags and NP features (thread level normalized position information) in addition to words as features.
- With reduced label set (3 labels) at phrase-level and using words as features.

Given a test discussion thread text, the above models are used to generate label sequence (after generating appropriate additional features as required). Since the main focus of this paper is extracting useful problem and resolution information from discussion forum threads, we only retain only the labels corresponding to problem (PROBLEM) and resolution (RESOLUTION STEP) from the output of the CRF. In the next section we describe the experimental setup used to evaluate our approach of extraction useful problem-resolution information from public discussion forums.

5 Experimental setup

5.1 Forum Data

To the best of our knowledge there are no publicly available standard corpora for discussion forums. Hence, the dataset used for the experiments is forum discussion data crawled from the SAP Community Network Forum[1]. This forum is a discussion place for SAP users - developers as well as customers of SAP products. The discussions in this forum are grouped into many categories, which include, SAP

[1] http://forums.sdn.sap.com

Message class	train	test
number of threads	34	45
number of labeled sentences	977	1057
PROBLEM	221	174
RESOLUTION STEP	268	139

Table 1: Distribution of the message classes in the training and test sets

Solutions, SAP NetWeaver, Business Intelligence, ABAP Development, etc.

The discussions on various topics, within the forum, are organized into threads and can be arbitrarily long, often spanning many pages. To train the CRF, each thread of discussion is provided as an instance, after manually labeling it at a sentence-level. Manual labeling was done using the Stanford Manual Annotation Tool[5].

Table 1 gives the number threads and labeled instances in the training and tests sets that were used, along with the distribution of the labels.

5.2 CRF implementation - MALLET

To conduct our experiments, we use the CRF implementation in the MALLET toolkit [12] which is trained with limited-memory quasi-Newton. MALLET is a Java-based package for machine learning applications to text, including sequence tagging. For training our data set, each labeled example is represented by a line, which has the format:

$\texttt{feature}_1\ \texttt{feature}_2\ \ldots\ \texttt{feature}_n\ \texttt{label}$

While testing, for each data point, the features constructed are entered in a single line, which is then passed as input to the trained model.

Furthermore, the CRF model used is trained using feature induction. Feature induction for CRFs includes automatically creating a set of useful features and feature conjunctions. McCallum [11] describes an implementation, where feature induction works by iteratively considering sets of candidate singleton and conjunction features that are created from the initially defined set of singleton features as well as the set of current model features. The log-likelihood gain on the training data is then measured for each feature individually. The candidates included in the current set of model features are only the ones that cause the highest gain. Our experiments indicated that using feature induction improved performance over just using all defined singleton features.

5.3 Evaluation metric

Precision:

Precision is a measure of the accuracy provided that a specific class has been predicted. Precision for $\texttt{class}_i$:

[5]http://nlp.stanford.edu/software/stanford-manual-annotation-tool-2004-05-16.tar.gz

	Precision	Recall	F1
PROBLEM	40.0	15.4	22.2
RESOLUTION STEP	57.1	27.9	37.5

Figure 4: Phrase-level model: Baseline - Results

$$Precision_i = \frac{TP_i}{TP_i + FP_i}$$

where TP_i and FP_i are the number of True Positives and False Positives respectively, for $\texttt{class}_i$.

Recall:

Recall is a measure of the ability of a prediction model to select instances of a certain class from a data set. Recall for $\texttt{class}_i$:

$$Recall_i = \frac{TP_i}{TP_i + FN_i}$$

where FN_i is the number of False Negatives for $\texttt{class}_i$.

F measure:

The F measure can be interpreted as a weighted average of the precision and recall. It is defined as:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

6 Results and discussion

This section describes the experiments done on the forum data using both phrase-level and sentence-level annotations.

We provide empirical evidence that using sentence-level annotations and labeling sentences by performing sequence tagging using CRFs provides better results than the phrase-level model. For sequence tagging, we use the MALLET implementation of CRFs and limited-memory quasi-Newton training (L-BFGS) [12]. In addition, we utilized MALLET's feature induction capability. Treating the problem as a sequence tagging task allows us to utilize contextual information present in preceding messages in the thread during training.

We annotate the sentences in each thread in the data set with multiple annotations described in Section 4.1. We distinguish two levels of annotation, viz., phrase-level and sentence-level. The phrase-level annotation involves assigning labels only to logical chunks of a sentence which generally span a couple of words. On the other hand, sentence-level annotations involve labeling the entire sentence.

To start with, we annotated a subset of data with all 17 annotations described in Section 3.3 in a sentence-level model. We trained a linear chain CRF on 977 sentences and

Features used	PROBLEM			RESOLUTION STEP		
	Precision	Recall	F1	Precision	Recall	F1
Baseline	57.2	59.2	58.2	41.5	61.2	49.5
Baseline + Normalization	61.6	67.8	64.6	40.5	76.3	52.9
Baseline + POS tags	61.5	70.7	65.8	49.7	69.1	57.8
Baseline + Normalization + POS tags	61.0	71.8	66.0	51.6	68.3	58.8

Figure 5: Sentence-level model: Accuracy values for different features

Problem	Predicted Negative	Predicted Positive
Actual Negative	TN: 803	FP: 80
Actual Positive	FN: 49	TP: 125

Resolution Step	Predicted Negative	Predicted Positive
Actual Negative	TN: 749	FP: 169
Actual Positive	FN: 44	TP: 95

Figure 6: Confusion Matrix for PROBLEM and RESOLUTION STEP

tested the obtained model on 1057 sentences. Results for this experiment gave an average accuracy of 34.7%. The precision and recall values were high for some labels and very low for others. For example, high recall values were observed for labels like RESOLUTION STEP (77.5%), GREETING (73.6%) and MESSAGE AUTHOR (49.1%).

In order to simplify the labeling task we then reduced the number of labels to just three: PROBLEM, RESOLUTION STEP and IGNORE. All the experiments described henceforth are on data annotated with just these three labels.

For the experiments for phrase-level classification, we annotated only the main parts (phrases) of the sentences with the PROBLEM and RESOLUTION STEP labels. The model was then trained on 395 phrases and tested on 168. Even though the classification task could achieve respectable precision, since the recall is quite low, the F1 values are not acceptable, as given in Figure 4.

For the experiments for sentence-level classification, the training and test set with the 17 labels were collapsed to contain only the main labels, viz. PROBLEM and RESOLUTION STEP, with the rest of it labeled as IGNORE. The baseline for this classification task used just the words in the sentence as features. Accuracy values for this experiment is given in Figure 5. As can be observed from this and the values in Figure 4, there is a substantial improvement in precision, recall and F1 measures for the sentence-level model when compared to the phrase-level model.

The second experiment at the sentence-level classification used thread length normalization feature in addition to the baseline. Length normalization essentially quantizes the position of the sentence within the thread and encodes it as BEGIN, MIDDLE or END in the input list of features for the sentence. From Figure 5, it can be seen that, adding length normalization tags improve the accuracy values for both PROBLEM and RESOLUTION STEP.

The next experiment was done by Part-of-Speech (POS) tagging the threads using only noun, verb, adverb and adjective forms. The POS tagging was done using Stanford NLP Group's Log-linear POS Tagger [19]. Figure 5 shows that using POS tags as feature improves the classification accuracy for both the labels.

In the final experiment for sentence-level classification, we combine all the three features i.e., words, length normalization tags and POS tags in the input feature vector. This leads to improvement in the accuracy values for both the labels when compared to the previous experiments, as seen in the precision, recall and F1 values given in Figure 5.

A confusion matrix presented in Figure 6 allows us to study the True Negatives, False Positives, False Negatives, and True Positives for the final experiment using the combination of all three features.

In the classification of instances as PROBLEM, we can observe that the False Negatives are 49. Out of this 49, 23 are instances of PROBLEM that were incorrectly classified as RESOLUTION STEP. In this case, the number of False Positives are 80 with 51 instances of IGNORE getting classified as PROBLEM and 27 instances of RESOLUTION STEP being classified as PROBLEM.

Similarly, in classification of instances of RESOLUTION STEP, the total number of False Negatives are 44. Out of the 44, 27 are instances of RESOLUTION STEP being incorrectly classified as PROBLEM. Furthermore, in this case the False Positive rate is 169 out of which 66 are instances of IGNORE being classified as RESOLUTION STEP and 23 instances of PROBLEM being classified as

RESOLUTION STEP.

In general, the False Positives seem to be high, thus degrading the precision. This could be because of the category IGNORE. This category contains instances which may contain irrelevant background information about the problem or personal experience of the authors, greetings, certain irrelevant suggestions etc. Since the category is not clearly distinguished by specific features from the instances labeled as PROBLEM or RESOLUTION STEP, it mainly contributes to the increase in number of False Positives.

From the experiments described above, the best results are obtained when using all of the three features in the input vector to train a linear CRF with the feature induction provided by MALLET [12]. The sentence-level model performs better compared to the phrase-level model as it is able to gain better contextual information from the entire sentence leading to increased accuracy during sequence tagging.

7 Conclusions

Online discussion forums are web communities where users collaborate to solve problems, and discuss and exchange ideas. A large fraction of discussions are problem solving threads where many members of the forum collaborate to solve an issue faced by a member. Archived discussion forum threads document a history of multiple problems solved in the past along with resolution details. They provide an excellent source of information for users seeking solutions to the same problem in the future.

In this paper, we studied the basic structure and information flow pattern in a discussion thread and proposed a method using CRFs to extract problem and resolution-related information by modeling the task as a sequence labeling problem. The extracted information enables easy access to solution steps for problems, thus providing improved information access in help desk scenarios. We presented results from classification experiments at a sentence as well as phrase-level. Our experiments showed improved accuracy while using a combination of features, such as words, POS tags and normalization based on the discussion thread structure, in the sentence-level model. Sequence labeling of problem, resolution and ignore, using CRFs, gave us F1 scores of 66% for problem related information and 58.8% for resolution-related information. This is an improvement over the baseline of 58.2% and 49.5% respectively, where the model is trained using only the words.

Future work to improve the extraction, is to include the meta-data associated with discussion threads like author specific information, and use of domain specific ontologies. Also, studying the usefulness of our method in a real help desk scenario will be an interesting direction of work.

References

[1] T. Baldwin, D. Martinez, and R. B. Penman. Automatic thread classification for linux user forum information access. In *Proceedings of the 12th Australasian Document Computing Symposium*, pages 72–79, 2007.

[2] M. Cakir, F. Xhafa, N. Zhou, and G. Stahl. Thread-based analysis of patterns of collaborative interaction in chat. *Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, 2005.

[3] V. R. Carvalho and W. W. Cohen. Learning to extract signature and reply lines from email. In *Proceedings Conference on Email and Anti-Spam*, 2004.

[4] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *Proceedings of SIGIR, International Conference on Research and Development in Information Retrieval*, 2008.

[5] S. Ding, G. Cong, C.-Y. Lin, and X. Zhu. Using conditional random fields to extract contexts and answers of questions from online forums. In *Proceedings ACL-HLT, Association for Computational Linguistics*, pages 710–718, 2008.

[6] J. L. Elsas and J. G. Carbonell. It pays to be picky: an evaluation of thread retrieval in online forums. In *Proceedings of SIGIR, International Conference on Research and Development in Information Retrieval*, 2009.

[7] J. L. Elsas and N. Glance. Shopping for top forums: Discovering online discussion for product research. In *1st Workshop on Social Media Analytics*, 2010.

[8] N. Glance and M. Siegler. Deriving marketing intelligence from online discussion. In *Proceedings of the 11th ACM SIGKDD, International conference on Knowledge Discovery in Data Mining*, 2005.

[9] A. C. Graesser, M. A. Gernsbacher, and S. R. Goldman. *Handbook of Discourse Processes*. Lawrence Erlbaum Associates, 2003.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML, International Conference on Machine Learning*, 2001.

[11] A. McCallum. Eciently inducing features of conditional random elds and Preethi. In *Conference on Uncertainty in Articial Intelligence*, 2003.

[12] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[13] A. Medem, M.-I. Akodjenou, and R. Teixeira. Troubleminer: Mining network trouble tickets. In *Proceedings 1st IFIP/IEEE international workshop on Management of the Future Internet*, 2009.

[14] F. Peng and A. McCallum. ”accurate information extraction from research papers using conditional random fields”. In *”Proceedings of HLT-NAACL, North American Association for Computational Linguistics”*, 2004.

[15] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[16] J. Seo, W. B. Croft, and D. A. Smith. Online community search using thread structure. In *Proceedings of CIKM, International Conference on Information and Knowledge Management*, 2009.

[17] Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis. ”easyticket: A ticket routing recommendation engine for enterprise problem resolution”. In *Proccedings of VLDB, International Conference on Very Large Data Bases*, 2008.

[18] L. Shrestha and K. McKeown. Detection of question-answer pairs in email conversations. In *Proceedings International Conference On Computational Linguistics*, 2004.

[19] K. Toutanova and C. D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 2000.

[20] N. Wanas, M. El-saban, H. Ashour, and W. Ammar. Automatic scoring of online discussion posts. In *Proceedings of the 2nd ACM Workshop on Information Credibility on the Web*, 2008.

[21] X. Wei, A. Sailer, R. Mahindru, and G. Kar. ”automatic structuring of it problem ticket data for enhanced problem resolution”. In *IFIP/IEEE International Symposium on Integrated Network Management*, 2007.