A CRM system for Social Media

[Challenges and Experiences]

Jitendra Ajmera
IBM India Research
New Delhi
jajmera1@in.ibm.com

Hyung-il Ahn
IBM Research
Almaden, USA
hiahn@us.ibm.com

Meena Nagarajan
IBM Research
Almaden, USA
meenanagarajan@us.ibm.com

Ashish Verma
IBM India Research
New Delhi
vashish@in.ibm.com

Danish Contractor
IBM India Research
New Delhi
dcontrac@in.ibm.com

Stephen Dill
IBM Research
Almaden, USA
dill@us.ibm.com

Matthew Denesuk
IBM Research
Almaden, USA
denesuk@us.ibm.com

ABSTRACT

The social Customer Relationship Management (CRM) landscape is attracting significant attention from customers and enterprises alike as a sustainable channel for tracking, managing and improving customer relations. Enterprises are taking a hard look at this open, unmediated platform because the community effect generated on this channel can have a telling effect on their brand image, potential market opportunity and customer loyalty. In this work we present our experiences in building a system that mines conversations on social platforms to identify and prioritize those posts and messages that are relevant to enterprises. The system presented in this work aims to empower an agent or a representative in an enterprise to monitor, track and respond to customer communication while also encouraging community participation.

Categories and Subject Descriptors

H.3.1 [**Information storage and retrieval**]: Content Analysis and Indexing; H.3.3 [**Information storage and retrieval**]: Information Search and Retrieval—*Information Filtering*; H.3.4 [**Systems and Software**]: User profiles and alert services

General Terms

Social Media, Social CRM

Keywords

Social Media Analytics, Sentiment Mining, Intent Mining, Text Clustering

1. INTRODUCTION

Social media is rapidly evolving as a new channel for customer care due to its popularity among a vast majority of users and ever increasing importance among enterprises. Customers of various enterprises are using social media platforms such as Facebook, Twitter and blogs for expressing explicit opinions, concerns and complaints about products and services they use, and also expect companies to respond. Enterprises are finding it important to address and try to resolve any customer complaint or concerns posted on these social media sites in a timely manner to retain and improve their brand value. Customer Relationship Management performed on social media, is called Social CRM [11]. Figure 1 shows an example of a social CRM conversation between customers and enterprise agents on a Facebook page set up by an automotive company.

Current social media monitoring applications [15, 1, 3] use aggregate-level statistics such as number of unhappy customers[1], brand mentions and associated sentiment, trending topics etc., which are not suited in a social CRM setting where it is needed to identify individual posts that needed to be responded to. For example, posts such as *'Can you bring back Pontiac? I hate to see it go!'*, are high on topical relevance and contain a question but may not be actionable and hence may not be prioritized high by an enterprise in the list of posts to be responded to. Social CRM conversations are also unique as they involve threads containing multiple participants and topics. The entire conversation thread and its structure need to be analyzed in order to provide effective assistance to agents delivering the CRM functions. This is in contrast to the current social media monitoring applications that use a collection of individual posts to draw aggregate level insights.

In this paper we present our experiences towards building a system that can help enterprises manage their social CRM function by assisting customer service representatives in identifying, monitoring and responding to actionable conversations in an effective and efficient manner. The system mines the conversation threads on social media to identify

---

[1]we will be using the terms "customers" and "users" interchangeably, although "users" is a superset of people using social media but not necessarily customers of the enterprise in question
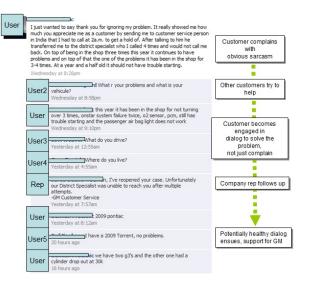
Figure 1: A CRM conversation between customers and a customer service representative

threads which are actionable, *i.e.*, warrant response by a customer service representative of the enterprise, and ranks them in the order of priority. This is achieved by taking into account several important considerations such as, the intent behind the post, its emotional charge, nature of the user who posted it, urgency of issues mentioned in the post, etc. It also provides a comprehensive view of all the *actionable* threads across multiple social media sites in the form of a dashboard arranged by various dimensions such as, categories relevant to the enterprise, authors, sentiments, etc. and gives a single interface to view and respond to the individual threads.

This work identifies new research challenges, such as, identifying conversation threads that are actionable from the ones which are simply just relevant but may not require a response, determining the category or the agent queue to which the thread should be assigned to and assigning a priority level for it. We also touch upon some other challenges which require further investigation such as, determining issue resolution state and customer satisfaction (CSAT) corresponding to the scenarios where a customer service representative has responded or intervened.

2. SOCIAL CRM BACKGROUND

Social media has been used extensively by the enterprise in recent past to get insights about what users think about their products or services [15, 3] and about the users themselves. This is typically achieved in a "listening" mode, *i.e.*, a large amount of data from multiple social media sites is analyzed in offline mode to extract aggregate level business insights.

However, an increasingly important requirement is being expressed by enterprises to go beyond this listening mode and actively engage with the users' conversations in real time, and directly on the social media platform. Towards this pursuit, enterprises have started to set-up their websites on popular social media channels such as, facebook and twitter, called "brand pages". Brand pages help enterprises capture relevant discussions among customers and provide

an opportunity to engage with them for customer care as well as marketing purposes.

Customers engaging in a conversation on the social media brand page of an enterprise expect attention and resolution to their concerns from the enterprise just as they would on a traditional CRM channel such as phone or email. However, social media brand pages, like the ones on Facebook and Twitter, differ from traditional CRM channels in several following ways.

- Every conversation on a social media brand page is public and hence visible for anyone to see. This has the advantage that other customers can read through posts and learn from them, but the disadvantage that a negative conversation can become viral in no time. This requires that enterprises act upon conversations in close to real-time.

- Another significant difference from the traditional CRM channel is that on a brand page, the enterprise needs to identify a relevant and actionable post (or thread) from a large number of ongoing conversation threads, all of which may not be relevant or actionable. While someone calling a traditional help desk is typically looking for help, many posts on a brand page are not directly relevant to customer care. Posts irrelevant to a CRM function may include compliments, marketing messages, or spam.

- In yet another important difference, customers can participate in the conversations on a brand page, helping to solve problems of a complaining customer or answer the queries posted by another. Companies may want to encourage this kind of participation, because it helps increase customer loyalty, and also reduces the burden on the customer service representatives.

These differences impose some key challenges with respect to what kind of analytics is required to support the enterprise perform the social CRM functions in an efficient manner.

Figure 1 shows an example of a CRM conversation on the brand page of an automotive enterprise created on Facebook. First, a customer posts a bitter complaint about the company. Next, other customers respond, to learn more details and try to help solve the problem. The initial complainer, not expecting a response, gets engaged to solve the problem. Finally the company and other customers continue the conversation until things are better. This example shows how social CRM can leverage the customer community to solve a customer's problems, and turn bad sentiment into good support. However, it does require that the company be involved in the conversations, or they are more likely to turn negative.

2.1 Salient Features of a Social CRM System

In the recent past, we have worked with social media data, especially data from brand pages, from a variety of industry verticals including automotive, finance and telecom, and interacted with domain experts to identify some important features of a social CRM system. We enumerate them below.

1. **Seamless data capture and ingestion:** A social CRM system needs to capture data (posts or conversation threads) from a variety of relevant social media

sources, ingest them into a common format, index and provide unified access to it

2. **Analytics to support customer issue identification and resolution, in an effective and efficient manner:** A social CRM system needs to sift through piles of conversations and identify what posts are actionable, *i.e.*, need a response, what topic the conversation is about and which business function (or category) within the enterprise a post should be routed to, such as marketing, product support, customer service, etc. A CRM system may also assist an agent in constructing responses for frequently asked queries or complaints or able to identify similar posts from other users.

3. **Leveraging community participation:** Communities can act as effective proxies for an enterprise by resolving issues from other consumers. One of the most important features of a social CRM system therefore is to promote community engagement by suggesting whether an agent should respond or let the community itself resolve the issue being discussed.

4. **Unified response interface:** A social CRM system should allow an agent to respond from a single interface regardless of the actual source/site where the data was captured from with an option to visit the site of origin. Response options should include individual messages, emails or chat, in addition to write a post in public.

In this paper we focus on item 2 above, *viz.*, the core analytics component to support the customer service representatives in resolving customer issues effectively and efficiently. Data ingestion and unified response interface components are also discussed in this paper but in lesser detail. Studying and leveraging community participation is an ongoing effort and preliminary insights are presented.

2.2 Research Challenges in Building a Social CRM System

Here we describe some of the challenges in realizing the goals of a social CRM system as outlined in the previous section. Obviously, many of these challenges may not exist in building a traditional CRM system and hence require special attention and addressal by the research community.

- **Informal and heterogeneous nature of content:** The heterogeneous protocols on social media platforms (e.g., Twitter has a text limit while blogs do not) and the informal setting in which participants converse, result in conversations that are highly unstructured, noisy, filled with misspellings and incorrect grammar [12]. Analytics for relatively well-structured CRM touchpoints such as email, do not translate effectively to this informal medium. It is imperative that a CRM system is able to process such user-generated content in an effective manner.

- **Identification of relevant posts:** Even though the brand-pages of various enterprises are set-up with focus to capture only the content relevant to the enterprise, a lot of content posted by the users could be about friendly greetings, topics such as weather and latest news and often times spam as well. Just keyword or simple rule-based techniques are not enough to deliver high precision and recall for relevant posts due to the presence of deep context due to the threaded structure, sarcasm and informal nature. On more generic social media channels this issue is simply much larger.

- **Identification of actionable posts:** While identification of relevant posts or threads is a necessary requirement towards engaging with the customers on social media, it is not a sufficient condition since not all posts relevant to the enterprise may be actionable. A simple example of such a post could be, "Company X's profits are going to be fall like rock this quarter". Although this post is very relevant to the enterprise X, it doesn't require any response from its customer service representative, at least not in near real-time. Another important aspect of actionability is that although a post may be actionable in itself it may no longer be actionable due to the response posted to it by another person as the problem/query may already be resolved by a fellow customer or user.

- **Priority Determination:** Due to a large number of potentially actionable posts or threads, it is important to assign a priority to each one of them to achieve optimal utilization of the customer service representatives' time. Unlike traditional CRM channels, where the identity of the customer is often used to prioritize, it may not be available in this case. Moreover, in addition to the importance of the customer to the enterprise, in this case the social influence of the author needs to be utilized to determine the priority their posts. In addition to author influence, intent of the post, sentiment expressed, urgency of the issues mentioned, etc. are some of the other important factors which determine the priority of the post. A combination of text mining and social network mining techniques are required to determine each of these factors automatically from the content which high accuracy, scalability and efficiency.

- **Bootstrapping challenges :** Most of the analytical techniques identified above need labeled data in order to learn key features to achieve high level of accuracy. Unlike email or call-centers, social media is a relatively new CRM channel and finding significant amount of labeled data is a big challenge. A Social CRM system should be able to deal with such a limitation.

2.3 Relevant Literature and Our Contributions

Significant amount of work has been done on using social media data for getting insights about customers and their views about an enterprise. Monitoring and predicting user behavior and events over social media data has been explored in detail in [15, 4]. There has also been some recent work on analyzing social media content on brand pages for monetization purposes [12, 10]. Work like [17] focuses on the former CRM process of using social data to identify potential leads for enterprises. The surge of social data has recently prompted researchers to envision conceptual models for social CRM systems that incorporate Web 2.0 technologies and user driven collaborative paradigms [11]. Using social media data for business intelligence [1, 3] etc. are also relevant in this context.

Analyzing social media data is challenging as it is inherently noisy due to informal use of language and hence poses a challenge for the current machine learning and natural language understanding techniques [14]. On the other hand, it also opens up opportunities for diverse research problems due its characteristically different structure. A number of interesting research problems around social media have been approached in recent times including attention prediction [5], author influence or authority determination [16] and event detection [6].

Whereas the common goal among most of the social media applications is to identify posts that are of interest to the users in the community, the central goal in social CRM applications is to identify posts that an enterprise needs to take action on. This requires achieving high recall while maintaining sufficient precision of the actionable social media content, i.e., ensuring that important customer posts are not missed, while not overloading agents with non-actionable posts. New social-media focused CRM vendors are emerging, such as, SugarCRM, RightNow, Genesys, PeopleBrowser, eGain and Attensity, just to name a few. While these offerings seem to provide basic analytical capabilities for obtaining, selecting, and routing posts, it is difficult to ascertain advanced analytical capabilities such as, detection of actionability and prioritization among actionable posts.

In this paper, we share our experiences and insights towards building a social CRM system which enables an enterprise to engage with users on social media platforms in an effective and efficient manner. We present analytical methods to identify posts or threads which are actionable for an enterprise by analyzing the content of the posts, relevance to the enterprise and social influence of the author. We present novel features, such as, user intent [13] and severity of issues in a user complaints, to determine the relative priority of a post among the actionable posts. We also determine the overall performance of the end-to-end system in terms of accuracy and time efficiency. Finally, we share our insights from the real life usage of the system and enhancements required for future.

3. SCION SYSTEM COMPONENTS

In this section we present details about SCION, our ongoing effort in building a Social CRM analytics engine that mines conversational data to enable enterprises to identify and respond to issues raised by customers. The social media CRM platforms that are of interest to us are those that are actively managed by corporations, for example their Facebook pages, Twitter handles or corporate forums and blogs to which consumers direct conversations (for example of a conversation relevant to a CRM agent, see Figure 1).

The SCION system comprises of four key modules, Data Ingestion Component, Core Analytics framework and the User Interface module as shown in Figure 2. These modules are explained in detail in the following sections.

3.1 Data Collection and Ingestion

An integral part of the SCION system is the ability to continually extract information from relevant data sources and process it in near real-time in order to facilitate comprehensive and timely responses by agents. The first step towards enabling this feature involves extracting data and metadata from data sources with minimal latency.
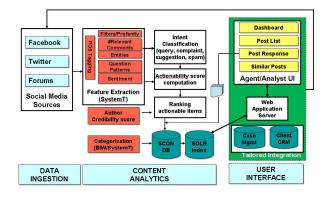

Figure 2: SCION system components

As shown in Figure 2, our primary data sources comprise of company brand pages from Facebook and Twitter. Handles to these pages are obtained during the setup process from the enterprise themselves. Both these social networking sites have gained a lot of popularity in recent times as enterprise customer interaction channels.. These sites provide REST APIs that allow applications to obtain data in the form of json objects. SCION employs a java-based framework that involves the invocation of these APIs in order to gather data. Our framework also facilitates processing of the json objects to retrieve relevant attributes from the data.

A post forms a basic unit of the data being crawled from these sites. The crawling mechanism involves periodic polling of relevant brand pages to check for new posts. When a new post arrives on a brand page hosted either on Facebook or Twitter, the crawling mechanism detects its absence in the SCION database and extracts information such as time_posted, user_id of the post author, text of the post, parent_id to which this post is a reply, number of likes/retweets that the post has gathered. SCION also captures the screen id of the user making the posts. We do not collect the user name or other profile information.

Posts on Facebook and Twitter are typically organized into conversations comprising of root posts and replies/comments. SCION is designed to capture this conversation level information. Each post stored in the database comprises of an attribute called the parent_id which identifies the post to which this post was a reply (or null if this is a root post). The JSON object returned by Facebook APIs allows the explicit capture of the root post (if any). Twitter APIs on the other hand, allow us to capture only the identifier of the previous post to which this post has been written as a reply. Tracking the root post that is the origin of a particular conversation is not as and simple on Twitter as it is on Facebook. In order to achieve this, SCION employs a specific logic. Whenever a new post arrives, the system looks up the parent id of the considered post and traces all parent ids back to the root post and assigns it as the parent to the new post.

As mentioned earlier, the data collected is ingested into a database. The interface between the data storage and the Java-based data collection framework is provided via IBM's DB2. At the end of the collection and ingestion process, the database is loaded with records in a form that can be readily used by the analytics framework.

Figure 3: Content Analytics components flow

3.2 SCION Content Analytics

The SCION system mines conversations occurring in a community for the end goal of supporting agents in responding to consumer issues in an efficient and effective manner. For every conversation post, the goal is to identify the intent of the post and then decide whether it needs a response and which business function can make the most appropriate response. Three components form the core of these analytics as illustrated in Figure 3: A topic categorizer that identifies the business function that is most suited to respond to a post, a module to identify if an incoming post is actionable or not based on its intent and finally a module that ranks all actionable posts for an agent to respond to. Here we describe each of the components in more detail.

3.3 Categorizer Module

The goal of this module is to group incoming posts so as to route them to appropriate business functions in a company. For the sake of clarity we will explain this module with respect to five business functions that are found in almost all industries; namely, *marketing and sales, product related, warranty issues,* and *customer care.* We will also assume that all other posts that do not belong to these categories is assigned to a category named *miscellaneous.*

During the setup process of SCION, enterprises are required to provide business functions that are relevant to them, in order of priority. In addition to categories and subcategories, they also provide us with models that are populated with seed words and phrases (between 5 and 10) that describe that particular function. For example, in the automotive industry the *Warranty* business function may include seed words such as warranty, coverage, liability, provider etc..

As mentioned earlier, the user generated content is typically noisy and simple rules derived from the seed words are not likely to suffice for the purpose of routing. Also, new concepts emerge periodically which are not likely to be captured with such simple rules. Therefore, our approach to identify the most relevant business function uses a combination of bottom-up unsupervised clustering and top-down rule-based matching techniques as illustrated in Figure 4. Only a rule based system with a list of terms for each business function would not have allowed us to capture all or evolving topics without significant manual intervention. A purely unsupervised approach to grouping posts on the other hand would yield clusters that do not directly translate to business functions.

The methodology can be summarized as follows (also see Figure 4). First, we generate clusters from a group of posts allowing a natural grouping of what the data contains. Next, we map the clusters to business functions by matching top discriminatory terms in the cluster with seed terms describing the business function.

Unsupervised Cluster Generation : The unsupervised clustering algorithm is borrowed from IBM's Busi-
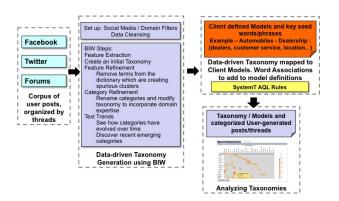

Figure 4: Two-step process for categorizing posts

ness Intelligence Workbench [2] that automatically performs the following steps to generate taxonomy of clusters from a group of posts.

- A dictionary of frequently occurring terms (words and phrases) is created, in our case provided by the enterprise analyst.

- A feature vector is created for each document containing the counts of the dictionary terms for that document.

- Text Clustering is applied to the feature vectors. Documents with similar term content are put together in categories that maximize intra-cluster similarity while minimizing inter-cluster similarity.

- Each cluster (category) is named based on the most frequently occurring word (or words). If no one word covers most of the documents in a category, then it is given a multi-word name, with the words separated by commas.

- In some cases a *Miscellaneous* category will be created containing all documents that could not be readily categorized.

- The final view is a list of cluster names, their sizes (number of documents) and a list of documents that belong to the cluster.

In the pre-processing stages BIW also allows for custom definition of stop words and to generate synonyms of frequently occurring terms. In SCION we find that it is useful to characterize common domain words such as cars, truck, auto in the Automotive domain as stop words since they are not generally discriminative across posts in the domain. BIW also optionally allows analysts to refine feature sets and clusters.

Mapping clusters to business function areas : Generated clusters are mapped to business functions or enterprise provided models by looking for a match between the top X discriminatory words in the clusters and business function dictionary terms. Functions with the majority match are chosen as relevant for all documents that lie within the cluster. In case of a tie, all functions are presented to an agent. An agent also verifies top X discriminatory words from every cluster to cautiously add to the business function dictionary of terms.

Real-time business function identification : While the above 2-step process is ideal when we have volumes of data, a rule-based technique is preferred when we have to classify incoming posts in real-time. In the current implementation of SCION, we run the above 2-step process periodically (every 24 hours), identifying top discriminatory words in clusters to enhance seed dictionaries. To identify

business functions in near real-time, we use terms in the enhanced seed dictionaries as inputs to a rule-based system [8]. The business function that matches a majority of terms is chosen as the route for an incoming post. In case of a tie, an agent decides which business function responds to a post.

The business function thus assigned to a post is used for routing the post to an agent assigned to the function. This metadata is also used in ranking actionable posts as we will describe in later sections.

3.4 Intent identification

The categorization as explained above categorizes the posts mainly based on presence of certain keywords or key-phrases and identifies the most relevant business function. However, just finding the business function is not enough for the final goal of prioritizing these posts for an agent to respond. Within each category, people can ask questions (e.g. *where can I find the customer service contact number in Canada?*) or express dissatisfaction (e.g. *the customer service in city XXX is pathetic*) or just share information. Depending on the business objective, only some of these 'intents' could be of importance to an enterprise for responding. An empirical analysis of intent types indicated that we could group posts among the following intent-types:

- Information seeking posts (Questions and Queries)
- Posts expressing dissatisfaction (Complaints)
- Information sharing (suggestions, news, positive comments, pictures etc.)
- Spam and other irrelevant posts

Note that there are other intent-types typically present in such data such as *intent to buy*, which could be of great importance for enterprises. However, in the current implementation of SCION we only focus on the customer care aspect of CRM and not much on the marketing and other engagement types. Therefore, we limit the discussion in this paper, specially the analytics and evaluation, to only the four intent-types mentioned above. In order to identify these four intent types, we start with extracting following four features from every post:

- Entities
- Sentiment
- Information Seeking Patterns
- Relevant Comments

In the following, we define each of these features and explain how these features are extracted.

3.5 Feature Extraction for finding Actionable Posts

We used IBM's SystemText (also referred as SystemT) annotation engine [8]. This engine makes information extraction orders of magnitude more scalable and easy to use. SystemT is built around AQL, a declarative rule language with a familiar SQL-like syntax. AQL replaces multiple obscure languages typically used to build annotators. Because AQL is a declarative language, rule developers can focus on what to extract while SystemT's cost-based optimizer determines the most efficient execution plan for the annotator. SystemT's information extraction engine is currently

deployed in many IBM products (Lotus Notes, IBM eDiscovery Analyzer, etc.) and is being used in several ongoing research projects.

Entity and Concept Extraction: We extract various types of entities and concepts from the posts such as: 1. Person names, 2. Dates, 3. Location names, 4. Organization names, 5. Currencies, 6. Platform specific patterns such as #hashtag, 7. Product and product part names 8. Acronyms, 9. Grammatically formed patterns such as *gerund adjective followed by a noun, e.g. promising results*.

A set of rules are created and written in AQL for extraction of these entities and concepts. These rules are primarily based on combination of some seed dictionaries and part-of-speech (POS) tags. An evaluation of such entity extraction module in terms of precision and recall is presented in the experiments section. Entities can be scored for their importance in a number of ways, including a manual override (by an agent) of what the system suggests. Our current implementation scores these entities based on their *tfidf* values within the entire dataset.

Sentiment Mining: Determining the sentiment orientation of a given post is one of the factors in judging the urgency with which a post needs to be responded to. Mining sentiment in text has been an active area of research in recent years and has been found to be a challenging task. A majority of previous work in this area formulate the problem as a binary classification problem (negative versus positive sentiment). [9] used a joint topic sentiment model based on latent Drichlet association (LDA) and reported an accuracy of 85% on this binary classification task. The datasets used for these studies also support this formulation in that most of the documents are opinionated either positively or negatively.

On the contrary, most of the posts on social media forums are not opinionated. On a sample of data that we labeled, we found that approximately 60% of the posts were adjudged not opinionated (neutral) and these mostly belong to the information sharing and the irrelevant/spam intent-types. Therefore, instead of considering it a binary classification problem, we categorize each post as expressing one of the three, neutral, positive or negative sentiments. This makes the problem more challenging. Consider for example this post: *hey XXX....does it mean that you are still having a lot of problems with your car?*. This post was adjudged neutral by human labelers although it does contain an opinion phrase. The problem is made even more challenging if we consider that most posts are short, do not have full context, and are often grammatically incorrect.

The sentiment mining rules in the current SCION system are based on dictionaries containing negative and positive polarity words and the neighboring context captured in terms of POS tags. The rules are written in AQL which are processed by the SystemT engine. These rules encode the fact that presence of a polarity word alone is not enough and that the surrounding context such as the presence of negation or blocking words (e.g. not, hardly) sarcasm, the nearest entity, the previous and the next sentence are all important to characterize the sentiment orientation of a post.

After all the negative and positive opinion phrases have been extracted from a post, the final sentiment is inferred based on the relative count of the two polarities. Higher the negative opinion phrase count (N_{ns}), more severe it is

considered for prioritization scoring as explained in the next section.

Information seeking pattern extraction: Presence of patterns such as *'how much is'* or *'where is the'* or *'I am looking for'* is indicative of the 'information-seeking' nature of a post. Our current implementation makes use of 11 such rules defined over the POS tags and dictionaries containing specific terms such as looking, searching etc. to detect the presence of these information seeking patterns. Following are some examples of such rules:

Questions: *"Can someone [please] tell...", "do you know..."*
Pattern: <Modal/Auxiliary Verb> <Noun/Pronoun> [0 or 1 token] $< Verb >$
Question: *"What can I....","when could [the] dealer..."*
Pattern: <Wh-question> <Auxiliary verb> [0 or 1 token] <Noun/Pronoun>

Community Participation: One of the characterizing aspects of the social CRM is the 'community' aspect. Any activity, be it from a customer or from the enterprise, is seen and shared by a lot of other community members. More people get involved in a discussion following a post is potentially indicative of something important and should be considered for human review. However, simply observing the volume of comments is not reflective of an actionable conversation. In accounting for community effect on actionable posts, we only account for comments (N_{comm}) that also score high on relevant concept indicators (as described above). The intuition being that an actionable post is more likely to receive responses that mention related concepts.

After these features have been extracted, we characterize the intent of a post as follows:

- Information seeking posts: Posts containing at least one information seeking pattern and at least one relevant entity

- Posts expressing dissatisfaction (Complaints): Posts containing negative opinion phrases (more than positive opinion phrases) and at least one relevant entity

- Information sharing: Posts containing at least one opinion phrase and at least one relevant entity.

- Spam and other irrelevant posts: Everything else.

Since the focus of SCION system is on the *customer care* aspect of CRM, in the following we assume that the first two intents (information seeking and expressing dissatisfaction) define the actionable set of items. The next section explains how these posts are scored to create a prioritized queue of posts for the agent to respond.

3.6 Ranking Actionable Posts

For scoring the actionable set of posts, we consider following three major factors:

- Actionability Score ($AScore$)
- Author credibility score ($AuthorScore$)
- Business Function Importance (BFI)

These three factors are explained in detail in the following.

Actionability score of a post: The actionability score ($AScore$) of a post is computed as a weighted combination of the features extracted in the previous section.

$$AScore = \ W_{ent} \cdot N_{ent} + W_{sent} \cdot (N_{ns} - N_{ps})$$
$$+ W_{cmt} \cdot N_{comm} + W_q \cdot (IfQ) \quad (1)$$

where, the subscripts *ent*, *sent*, *ns*, *ps*, *cmt* and *q* correspond to entities, sentiments, negative sentiment, positive sentiment, comments and information seeking patterns, respectively. $N_{\{.\}}$ denotes the number of times a particular feature was detected in the post and $W_{\{.\}}$ denotes the weight (importance) assigned to that feature. These weights can be adjusted to realize any prioritization or routing mechanism. IfQ denotes the presence of at least one information seeking pattern. The choice of IfQ instead of the total number of information seeking patterns was made based on the observation that most genuine queries were typically short and contained not more than one such pattern.

Author credibility score: Some users are naturally helpful participants while others engage in inflammatory responses. Some users digress conversations or routinely post non-actionable posts such as sharing pictures of their car on social media pages. This score reflects the intuition that posts of authors who are generally helpful participants, or who typically post relevant, specific content should be ranked relatively higher than those of authors who are not. In SCION, we model an author's credibility score as:

$$AuthorScore = (Avg(N_{comm}) \cdot Avg(AScore) \cdot N_{rep})^{\frac{1}{3}} \quad (2)$$

where, $Avg(N_{comm})$ is the average number of relevant comments received on the author's posts so far, normalized to 1.0. $Avg(AScore)$ is the average actionability score ($AScore$) assigned to author's main posts so far. N_{rep} is the normalized number of interactions that the author has had with the company representatives in past. These factors are initiated and updated with some minimum default values to avoid the values becoming an absolute zero for any author.

Business Function Importance (BFI): In any industry, different business functions assume priority at different points in time. For example, posts relevant to the marketing and sales function receive higher priority during a new product release cycle. SCION uses a tunable ranking of business functions as provided by an enterprise during the setup of the system. In the absence of an enterprise provided ranking, one can also simulate the score based on actionable content found in a category. Equation 3 shows one method of arriving at a business function prioritization:

$$BFI(category) = \frac{(N_{actionable}|category)}{(N_{total}|category)} \quad (3)$$

where $N_{actionable}$ is the number of actionable posts (complaints and queries) in a given category and N_{total} is the total number of posts in that category. Note that this is only one realization of such function. Any guidance of importance provided by the enterprise can be encoded in any of the functions including the actionability score ($AScore$) computation. The final ranking function incorporates all of the above parameters as follows:

$$RankScore = (AuthorScore * BFI * AScore)^{\frac{1}{3}} \quad (4)$$

The $RankScore$ essentially accounts for three independent factors which can influence the priority of a post: 1. The relevance score $AScore$ is computed from the content of a single post and determines how relevant and urgent is the post for responding, 2: The $AuthorScore$ considers the 'value' of an author by considering its previous activity. It is

possible, for example, that a given post does not score high on relevance but the same author has made a lot of valuable points in past and therefore should be considered with priority and 3: The *BFI* determines the urgency of a particular business function and it helps in determining the priority of a post in two ways. First, If a lot of complaints are being posted in a given category then that category should be responded to with the most immediate attention. Secondly, the relevance function *AScore* is determined from factors (intent, sentiments etc.) which are mostly rule based. The rule based factors have a very high precision but are likely to miss out on many relevant posts which do not adhere to the designed rules. The bottom up clustering used for categorization can alleviate some of these concerns.

4. FINDINGS AND COMPONENT EVALU-ATIONS

To assess the performance of our system, we did an evaluation test for each module. The results from our evaluation tests indicate that overall the system integrating the three basic modules, the categorizer, the actionable post identifier and the ranker is very effective in helping agents identify what posts to respond to. In addition, since our ranking function is based on a linear combination of multiple features (annotations), even if one of the annotations is wrong, the ranking function is still very likely to prioritize the post.

Setup: All the data used for evaluating our modules was from Facebook Pages (serving as social CRM channels) of companies in Automotive Industry. We collected a total of 10,385 conversations with a total of 63,593 posts including 16,334 participants over a period of time ranging from May 2009 to August 2011. For the purpose of evaluating different annotators in our system, six human coders were employed to label two datasets for various ground truth parameters and a pair-wise agreement by each group of two coders was obtained for their labels. In other words, when two coders disagreed on their labels, they consulted with each other to reach an agreement [7]. A labeled set of 1000 main posts created by randomly selecting conversation threads whose initial post was written by a customer was used for evaluating the categorizer module and the actionable post identifier (Evaluations 1 and 2 below). Also, a smaller dataset of total 149 main posts that were randomly selected from 1000 main posts was prepared for evaluating the performance of the ranking function (Evaluation 3 below).

Evaluation 1: Categorizer Module : The rule-based portion of the categorizer module was employed to classify the main post of each conversation thread into four business functions in the Automotive industry: *marketing and sales, product related, warranty issues*, and *miscellaneous*. We compared the categorizer's annotations with human labels on the labeled set of 1000 main posts for an accuracy of 76.2%. As is common for rule-based systems, posts that had insufficient or ambiguous contexts fared poorly. Given the new form of this medium, we are also faced with bootstrapping challenges in getting relevant posts for a business function in order to automatically construct rich dictionaries. This is an area of continued investigation for us. We are looking into methods of bringing agents into the loop by allowing them to add terms to dictionaries when they encounter them in posts and also vote on incorrect classifications so that the system can learn from it.

Annotation	Precision	Recall
Questions	0.77	0.49
Complaints	0.63	0.78
Non-actionable Posts	0.74	0.70

Table 1: Precision and recall rates for post-type annotations

Annotation	Precision	Recall
Concept Extraction	0.76	0.68
Negative Sentiment	0.53	0.65

Table 2: Precision and recall rates for concept extraction and sentiment annotators

Evaluation 2: Actionable Post Identifier : Of the 1000 labeled main posts, 142 were question posts, 316 complaint posts, and 500 were non-actionable in nature. Note that there were 330 main posts to which automobile company agents had already responded with their personal Facebook account names. These were easy to identify because of their signatures, such as *Customer Representative*. We evaluated our actionable post identifier module for identifying queries and complaints against this dataset. Table 1 shows the precision and recall rates for the post-type annotators; Questions, Complains, and Non-actionable Posts. In particular, the good performance in terms of both precision and recall rates on non-actionable posts implies that we are able to distinguish non-actionable posts from actionable posts very well.

In addition, using the dataset of 149 main posts, we tested the performance of the concept and sentiment annotators as shown in Table 2. To evaluate the concept annotation, human coders examined if each extracted concept from the annotator is actually relevant or not to the post. Also, we counted how many relevant concepts the annotator did not find. For the sentiment evaluation, we compared the human-labeled sentiment (positive, neutral, negative) with the annotated sentiment for each post. While the performance of the concept extraction module as shown in Table 2 is reasonably good, we found that our sentiment extraction module must be improved further. In further investigation we found that the sentiment extraction at the phrase level is quite precise but the problem appears when we extend the phrase level decision to the post level. Currently this is achieved by comparing the number of positively opinionated phrases to that of negatively opinionated phrases but a simple comparison is not sufficient. When more labeled data is available, this could be improved, for example, by using a regression function.

Evaluation 3: Ranking Actionable Posts: We tested the performance of the ranking function using the dataset of 149 main posts. There were two human-coded labels for each main post in the dataset: the first label is called *actionability label* obtained from human coders as answers to question like *Do you think that the main post has an issue that needs a response or resolution?* The second label is *issue-severity label* obtained using the question like *If actionable, what is the degree of severity of the issue of the main post? (High/Moderate/Low)*.

Actionability /Severity	Number	Mean Score	St. Dev
Actionable / High	21	0.23	0.14
Actionable / Moderate	22	0.17	0.11
Actionable / Low	17	0.13	0.11
Non-actionable	89	0.09	0.09

Table 3: Mean and standard deviation of the ranking scores for each issue-severity degree and non-actionable posts.

In this evaluation test, we are examining the effects of human-coded actionability and issue-severity labels on the ranking score from our ranking function. Note that the ranking score is used for prioritizing posts presented to an agent. Our hypotheses are: first, actionable posts will get higher ranking scores compared to non-actionable posts. Second, posts with high issue-severity will get higher ranking scores compared to those with moderate or low issue-severity.

From Table 3, we see that there is a significant effect of the actionability label on the ranking score at the $p < .001$ level, $F(1,147)=26.2$. Actionable posts tended to achieve higher ranking scores (Mean=0.18) compared to non-actionable posts (Mean=0.09). Moreover, the issue-severity label had a significant effect on the ranking score at the $p < .05$ level, $F(2,57)=3.51$. Post-hoc comparisons (Tukey) showed that issues with high severity (Mean=0.23) tended to get significantly higher ranking scores than issues with low severity (Mean=0.13) ($p < .05$). Although the table above shows that the score resulting of our ranking function largely approximates the issue severity in the case of the actionable posts and that it can also be used to filter out non-actionable posts, it does not provide an objective evaluation of the ranking function score.

For this purpose, we formulate the problem as that of a classification problem. In this formulation, we consider a classification correctly done by the system if it predicts a higher ranking score for a higher severity post compared to the score assigned to a lower severity post. This formulation also allows us to evaluate the scoring function on a significant number (6533) of severity degree pairs where a pair is defined as a set of two posts such that they belong to different severity classes (e.g. low and moderate). We obtained an accuracy of 77% with such formal evaluation.

All these results confirm that the ranking module can be successfully applied to prioritize main posts according to the actionability and the issue severity.

5. EXPERIENCE AND INSIGHTS

In this section we describe our experience in building the IBM SCION system and its usage towards social CRM. We also suggest direction for future work and improvements.

We found that a crucial feature that governs the performance of a social media based analytics system is the "nature" of the data itself. For example, if the data is crawled from a social media "stream" then methods for filtering the posts to make them relevant to the task, need to be employed. In addition, word senses and ambiguity need to be dealt with, which is not as necessary if a social media profile page or a hashtag/keyword based search query is used to crawl the data. SCION easily integrates custom filters

into the data pipeline before running its content analytics engine.

A key characteristic about social media data is the presence of highly opinionated information with a very low shelf life. CRM experts we worked with highlighted the importance of near-real time processing for the effective use of a social CRM system. Users posting a minor complaint about a product can begin to express more negative sentiments, if their queries are not addressed quickly enough. In addition, due to the public nature of this medium, a delayed response from an agent may cause other users to express further negative opinions and make the problem more severe from a CRM point of view. Therefore, social CRM systems should be able to process vasts amount of posts very efficiently and accurately, providing near-real time insights to customer agents about their brands/products. SCION uses the state of the art SystemT based annotators for its analytics and a single instance of the annotator engine can process over 950 social media posts per second.

Opinion detection in social media is difficult and is an active field of research. Existing techniques have reported varying accuracies based on the amount of noise present in the data as well as the "complexity" of the opinions expressed. For example, posts with sarcasm or humour are much harder to detect than posts expressing simple positive and negative sentiments. Further, in the context of social media CRM, it is also important to be able to associate sentiment with the entity of interest. A post expressing negative sentiments about a competitor should not be tagged as a high priority or actionable post, while a positive sentiment about a competitor with complaints about a product offering requires redressal. SCION uses rule based methods to associate sentiment to concepts, and as future work we plan to make this more robust by employing probabilistic methods.

CRM practitioners interested in SCION, expressed the need for being able to configure the criteria used for identifying and ranking actionable posts. This requirement stems from the need for social CRM systems to be able to provide useful insights regarding a brand apart from being a channel for customer engagement. CRM practitioners required configurability, not only based on the business functions of their enterprise, but also based on seasonal sales patterns. For example, during the holiday season when customer purchases tend to increase, enterprises may want to rank posts expressing intent to purchase higher than those related to feedback. However, if a company has released a new product into the market, it may be more interested in gauging the response to it's offering. Thus, keeping this requirement in mind, we allow the weights defined in Equation 1 to be configurable, and higher weights can be assigned to different scoring features for the actionability score. E.g. Posts expressing negative sentiment (complaints) can be given higher importance by assigning a higher weight factor to the sentiment component of the score. Another feature that attracted interest by CRM practitioners was SCION's ability to categorize similar posts. This has two advantages - first, posts that require a mass response can be immediately responded to (Eg. Responding to queries asking about product updates etc). Secondly, the categorization helps ensure consistency in responses thus, reducing the chances of erroneous or conflicting information form getting dispersed.

Lastly, the configurable weights used for scoring are set empirically based on the requirements specified by CRM users. With the deployment of SCION, we will be able to collect a rich source of manually assigned weights and posts, which could be used to build semi-supervised methods for automatic assigning weights for different CRM scenarios.

6. REFERENCES

[1] Cognos business intelligence and financial performance management. In *http://www-01.ibm.com/software/analytics/cognos*.

[2] A. Behal, Y. Chen, C. Kieliszewski, A. Lelescu, B. He, J. Cui, J. Kreulen, J. Rhodes, and W. S. Spangler. Business insights workbench: an interactive insights discovery solution. In *Proceedings of the 2007 conference on Human interface: Part II*, pages 834–843, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] M. Biere. Business intelligence for the enterprise. In *IBM Press*. IBM Press, 2003.

[4] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 78–87, New York, NY, USA, 2005. ACM.

[5] T. Hogg and K. Lerman. Stochastic models of user-contributory web sites. *ICWSM*, pages 50–57, 2009.

[6] Y. Hu, A. John, D. D. Seligmann, and F. Wang. What were the tweets about? topical associations between public events and twitter feeds. In *6th Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*, 2012.

[7] H. il Ahn, W. Geyer, C. Dugan, and D. R. Millen. "how incredibly awesome!" - click here to read more. In W. W. Cohen and S. Gosling, editors, *ICWSM*. The AAAI Press, 2010.

[8] R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, S. Vaithyanathan, and H. Zhu. SystemT: a system for declarative information extraction. *SIGMOD Rec.*, 37(4):7–13, Mar. 2009.

[9] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 375–384, New York, NY, USA, 2009. ACM.

[10] P. Melville, V. Sindhwani, and R. D. Lawrence. Social media analytics: Channeling the power of the blogosphere for marketing insight, 2009.

[11] S. Mohan, E. Choi, and D. Min. Conceptual modeling of enterprise application system using social networking and web 2.0 "social CRM system". In *Proceedings of the 2008 International Conference on Convergence and Hybrid Information Technology*, ICHIT '08, pages 237–244, Washington, DC, USA, 2008. IEEE Computer Society.

[12] M. Nagarajan. Understanding user-generated content on social media. In *Doctoral Dissertation*. Wright State University, 2010.

[13] M. Nagarajan, K. Baid, A. Sheth, and S. Wang. Monetizing user activity on social networks - challenges and experiences. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '09, pages 92–99, Washington, DC, USA, 2009. IEEE Computer Society.

[14] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. pages 1524–1534, 2011.

[15] S. Spangler, Y. Chen, L. Proctor, A. Lelescu, A. Behal, B. He, T. D. Griffin, A. Liu, B. Wade, and T. Davis. COBRA - mining web for corporate brand and reputation analysis. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 11–17, Washington, DC, USA, 2007. IEEE Computer Society.

[16] J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank: Finding topic-sensitive influential twitterers. *WSDM*, pages 261–270, 2010.

[17] B. Wu, Q. Ye, S. Yang, and B. Wang. Group CRM: a new telecom crm framework from social network perspective. In *Proceedings of the 1st ACM international workshop on Complex networks meet information and knowledge management*, CNIKM '09, pages 3–10, New York, NY, USA, 2009. ACM.