Statistical Affect Detection in Collaborative Chat

Michael Brooks[1], Katie Kuksenok[2], Megan K. Torkildson[1], Daniel Perry[1], John J. Robinson[1], Taylor Jackson Scott[1], Ona Anicello[1], Ariana Zukowski[1], Paul Harris[1], Cecilia R. Aragon[1]

[1]Human Centered Design & Engineering, [2]Computer Science & Engineering
University of Washington, Seattle

ABSTRACT

Geographically distributed collaborative teams often rely on synchronous text-based online communication for accomplishing tasks and maintaining social contact. This technology leaves a trace that can help researchers understand affect expression and dynamics in distributed groups. Although manual labeling of affect in chat logs has shed light on complex group communication phenomena, scaling this process to larger data sets through automation is difficult. We present a pipeline of natural language processing and machine learning techniques that can be used to build automated classifiers of affect in chat logs. Interpreting affect as a dynamic, contextualized process, we explain our development and application of this method to four years of chat logs from a longitudinal study of a multi-cultural distributed scientific collaboration. With ground truth generated through manual labeling of affect over a subset of the chat logs, our approach can successfully identify many commonly occurring types of affect.

Author Keywords

Computer-mediated communication; scientific collaboration; affect; emotion; machine learning; chat; spontaneous text communication; social media.

ACM Classification Keywords

H.5.3. Group and Organization Interfaces: Computer-supported-cooperative work.

General Terms

Human Factors; Measurement; Algorithms.

INTRODUCTION

Geographically distributed collaboration is increasingly common across many work domains, and understanding the expression of emotion in computer-mediated communications is crucial to understanding team interactions and processes.

Against prior views of emotional expression as merely inappropriate disturbances within a work setting, an increasing number of studies in the last twenty years have documented a renewed interest in understanding emotion and affect in the workplace [4,5,11]. Numerous studies have shown that affect and emotion influence performance and interactions in cooperative work environments [1,4,11,16,17]. Further, this "awakening of interest in emotion in CSCW" [16] can support the design of affect-aware information systems in the workplace.

There are many definitions of emotion, affect, sentiment, and related concepts; many definitions are overly restrictive given the wide range of emotional or affective phenomena that may be of interest in understanding cooperative work environments. In this paper we draw on Russ's broad definition of *affect* to refer to an inclusive concept spanning emotions and feelings distinct from cognition [26], and more pervasive than the neurophysiological experiences of emotions [20]. We seek to better understand how instances of this broader notion of affect manifest in collaborative text-based communication.

Text-based chat within collaborative scientific work provides a rich body of data for understanding affective processes within groups. The increasing volume of text-based communication available for study, combined with a growing awareness of the importance of affect in the workplace, have led to an upsurge in research on affect detection in text, including work in fields as diverse as sentiment analysis, affective computing, linguistics, and psychology, among others [11].

Manual coding for affect expression in chat logs can yield rich and reliable data, but this process does not scale well to larger data sets. Research on the automated detection of the overall positive or negative sentiment of long, relatively well-formatted blogs, articles, and online posts has achieved promising results with statistical classification methods based on frequencies of term occurrence, e.g. [10,14,30]. More recently these methods have been applied in more informal settings, focusing on classifying messages on social network sites, blogs, and discussion forums, which are characterized by irregular grammar and spelling practices, e.g. [26, 33].

However, more work is needed to develop classification methods robust to the varied and dynamic context of affect in collaborative and distributed online chat environments [25]. Recent work in this area has applied rule-based techniques to the detection of specific types of affect, e.g. [15,21]. Effective classification of very short, informal

CSCW '13, February 23–27, 2013, San Antonio, Texas, USA.

texts, such as chat messages, remains a challenging problem for current text classification methods.

In this paper, we address the problem of detecting subjective, non-mutually-exclusive labels of affective state (*e.g.*, *joy*, *excitement*, *confusion*, *frustration*, *anger*, and *annoyance*) in workplace chat logs. We contribute (1) a novel approach to affect classification in chat logs based on an interpretation of affect as a dynamic process, (2) a novel combination of features for use in classification of affect expression in chat, and (3) *ALOE*[1] – open source software for classifying coded chat messages. Our technique was applied to a large data set of nearly 500,000 lines of chat collected over four years of an international scientific collaboration. We were able to successfully identify 13 common types of affect expression from a taxonomy developed via collaborative open coding. Automated techniques to identify affect in chat messages are a powerful addition to the analytic toolkit of researchers studying affect in distributed teams.

RELATED WORK

Previous work includes a diverse set of approaches to affect classification, differing both in the evidence (features) used as well as the classification method. The variety of ways that the problem has been studied and the many techniques that have been developed make comparison between these studies challenging.

The features used for affect classification include everything from statistics on the frequencies of word sequence occurrences (word counts, n-grams) to more linguistically informed features (e.g. part of speech). These features have been combined with countless classification methods, ranging from rule-based methods to more flexible probabilistic methods. Depending on characteristics of the data, different configurations can have different levels of success. Furthermore, different granularities of classification – whether to classify affect merely as positive or negative, or to use a finer set of categories – not only impact the success of approaches, but make results even more difficult to compare between studies.

A significant amount of research on affect or sentiment detection in text has focused on lexicon-based approaches, in which pre-determined dictionaries of words that are associated with the target affect categories are used to generate features for machine learning algorithms. The Linguistic Inquiry and Word Count tool (LIWC) uses a predefined lexicon, counting words in specific psychological categories in order to measure characteristics of a text [30]. This tool has been successfully used to study the prevalence of emotion or affect in documents [10]. Taboada et al.'s Semantic Orientation CALculator [29] used manually compiled lexicons for sentiment analysis, with consistent results across a variety of types of product reviews.

Many sentiment analysis techniques, such as these, are most successful when applied to carefully authored, lengthier content, but often struggle when faced with informal online communication. There have been efforts to adapt some of these techniques to work with such content. Thelwall et al. detected positive, negative, and neutral emotions in MySpace blog posts, which use more informal language including nonstandard spellings and grammar. Classifying the strength of both negative and positive sentiment independently on 5 point scales, their algorithm, SentiStrength, performed well relative to other machine learning approaches because of its ability to correct misspellings and its sentiment strength lexicon, in combination with other features [31].

The Affect Analysis Model uses a database of emoticons, abbreviations, interjections, and other words that had been manually associated with nine emotions to drive a rule-based affect classification system that analyzes affect at the word, phrase, and sentence level [21]. This system was developed specifically for informal online communications, including hundreds of popular abbreviations and emoticons in its database. Over two blog post data sets, the Affect Analysis Model reached 72% and 77% accuracy, and outperformed other systems on news headlines. Although work such as this is promising, "spontaneous text communication" such as collaborative online chat presents other unique problems, with extremely short, hastily written messages and rapidly shifting topics and affective states.

There are noted challenges in applying lexicon-based approaches, such as the LIWC tool, to naturalistic or jargon-ridden language [25]. Some word associations that lexicon-based approaches rely on break down in specialized domains. In techniques developed for more formal documents, punctuation, nonstandard capitalizations, grammar, and misspellings are often discarded as noise. This makes LIWC and other existing tools inappropriate in this work. In our data set, messages often communicate affect through grammar and spelling modification, capitalization, and punctuation: e.g. "in there???" and "WHAT WHO DID THAT" (*annoyance* and *frustration*). While these issues have been studied before in the context of blog posts, very little previous work has focused on affect classification in chat messages.

In addition to lexicons, a variety of other approaches have been studied. Liu et al.'s EmpathyBuddy made use of real-world knowledge, obtained from the Open Mind Common Sense knowledgebase, to power an affective email client, which displayed Chernoff face-style feedback alongside email messages. Participants in a user study perceived the email client as more intelligent than a version which displayed random faces [15].

[1] http://depts.washington.edu/sccl/tools

Other work has had some success with n-gram features, or short phrases derived from the data set itself, to inform classification. Rather than a predefined lexicon of words with known associations, the significance of the n-grams is learned from the available data. Aman and Szpakowicz experimented with lexicon-based features from the General Inquirer and WordNet-Affect [28] in addition to other features, such as emoticons, exclamation points, and question marks, for classifying blog post sentences according to a taxonomy of six basic emotions. The combination of all of these types of features was found to be the most successful configuration, compared to using any one group of features in isolation [2].

Mishne supplemented word counts with punctuation, emoticons, and length of blog post. While providing an important exploration of a feature-enhanced analysis approach, the results yielded a modest 8% improvement over the 50% baseline on average, stating that further improvement could be obtained by increasing the amount of training data [18].

Gilbert explored the relationship between the words written in emails and the rank of the email recipient in the workplace hierarchy. Recipients were ranked as higher, lower, or the same as the sender within a dataset of 2,000 messages from the Enron email corpus. A logistic regression model was used to determine the most predictive unigrams, bigrams, and trigrams from the emails. Support vector machine (SVM) classification based on these phrases achieved 70% accuracy with three-fold cross-validation [9].

Learning the terms of interest from the data set under study can provide both advantages and disadvantages. Mohammad experimented with both lexicons and n-gram features to classify affect in text. Findings showed that the efficacy of word-level lexicons (WordNet-Affect and NRC) was correlated with the size of the lexicon, with the larger lexicon (NRC) showing significant improvements over the use of n-gram features alone for sentence level affect classification. The study also found that the classification performance of n-gram features was domain specific; namely that n-gram features trained on data from one domain were unable to classify affect as well as lexicon features when transferred to a new domain [19].

Most work in this area has focused on texts written by an individual, but there has also been work on affect detection in collaborative contexts, e.g. investigating applications of modern text classification techniques for analyzing computer supported cooperative learning environments [25]. There remains a need for affect classification methods suited to chat messages from distributed collaborative work.

CLASSIFYING AFFECT IN TEXT CHAT LOGS

Our goal was to automate the process of affect labeling in a specific set of chat logs produced by an extended scientific collaboration. We begin by describing the corpus and the manual labeling process we used to generate truth data for machine learning. Then, we describe the results of experiments on different feature selection and classification configurations.

The Supernova Factory Chat Dataset

Our dataset is comprised of chat logs collected from the Nearby Supernova Factory [3], an international astrophysics collaboration of approximately 30 core members; about half of the scientists were located in the U.S. and the other half in France. The scientists were monitoring the occurrence of Type Ia supernovae, a specific type of stellar explosions that have a consistent brightness that allows them to be used to effectively measure the distances to other galaxies and trace the expansion history of the universe. The scientists, distributed across multiple time zones, operated their telescope remotely three nights per week using chat as the primary means of communication; during such operation, numerous technical and scientific decisions involving the operation of the telescope had to be made quickly and collaboratively.

There are a total of 485,045 chat messages in the corpus. The top 32 human participants contributed over 500 messages each, or 300,684 messages total, accounting for the majority of the data. Most of the rest were produced by automated programs ("bots") using chat to relay critical changes to the environment (sunrise/sunset; weather; telescope settings, etc.) [24]. Individual chat messages are very short, with the vast majority between 5 and 10 words in length. The chat logs span 1,319 days (nearly four years).

Coding for Affect

We prepared training data through a manual coding process. A subset of the data was annotated with any number of non-mutually-exclusive affect codes by between 1 and 5 undergraduate and graduate students in our research group. Below, we describe in more detail the mechanics and justification of the manual labeling process.

Because our main goal in this work was to facilitate a rich analysis of the dynamics of distributed work in a specific data set, we constructed a taxonomy of affect [27] through a combination of open, axial, and selective coding grounded in the data [6] and affective terms from Plutchik's taxonomy of emotion [22,23]. This approach was our solution to the problem of translating a large body of existing work on affect and emotion into a more appropriate and useful analytic tool for our particular data set. The resulting taxonomy allows us to examine the specific types of affective expression present in our data because it accounts for the distinct ways in which these expressions are molded by the text-based medium.

Over the course of several months we conducted iterative, open coding of the chat messages. Coders were allowed to label messages with as many affective terms as they believed applied to each message, and they were free to add new terms to the taxonomy. Messages could be coded as

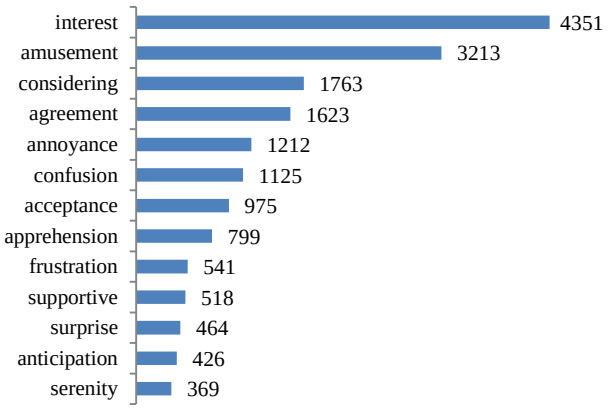

Figure 1. Number of times each of the top 13 affect codes was applied.

time	*speaker*	*message*
05:58:41	Alice	ok, so where was the **** SN on the image? #1: interest / anger #2: annoyance / confusion#3: interest / frustration
05:58:55	Alice	was it the bright blob? #1: interest / anger #2: considering #3: interest
05:59:03	Ben	5876 absorption is much wider than the H alpha in v space #1, #2, #3: no affect
05:59:18	Ben	Oh hmmm. #1, #2, #3: considering
05:59:28	Ben	Lemme see what [the] coordinates were... #1, #2, #3: no affect

06:13:07	Charlie	is it "well-developed"? #1: interest
06:13:18	Alice	Should be an interesting experiment. #1, #2: anticipation #3: interest
06:13:19	Dana	yes #1, #3: agreement #2: no affect
06:12:20	Dana	big!! #1: excitement / agreement #2, #3: excitement

Figure 2. Two anonymized examples of conversations from our dataset. Each segment was coded by three members of the research team; these annotations are shown below each line.

"no affect" to distinguish messages that had been identified as lacking affect from those which were not yet coded.

The interpretation of affect in any recorded communication is inherently subjective. In deciding which affect labels to apply to a given chat message, coders were asked not to attempt to guess what emotion the speaker was feeling, nor the affect that the speaker may have intended to express. Instead, they were asked to focus on the affect that they believed was communicated by the message in the context of the conversation. Coders were encouraged to trust their

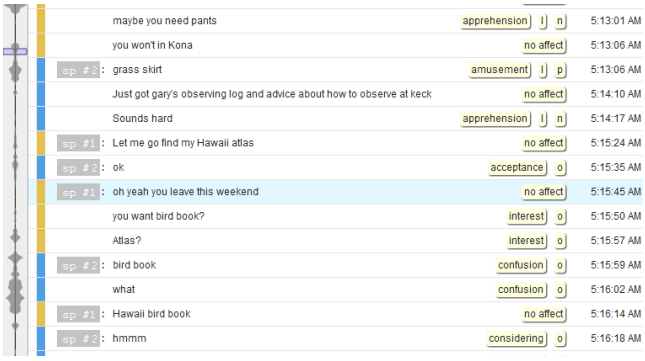

Figure 3. A screenshot of the coding tool developed by team members, as seem by an individual coder applying codes. (Codes *o, l, h, n,* and *p* refer to neutral, low, and high intensity, and negative and positive valence, respectively.)

own instincts as experienced human readers of chat messages from this data set.

Given the nature of this task, we did not find that any existing qualitative data analysis software was suitable. We needed to allow up to 10 human coders to efficiently work together on thousands of short, sequential lines of chat data, and this data needed to be easily accessible and transformable for machine learning procedures and statistical analysis. Thus, as part of preparing the ground truth data, we developed and used our own web-based coding tool, pictured in Figure 3, which we plan to present in greater detail in a future paper and release publicly.

As the affect taxonomy stabilized, we integrated the categories we had developed with Plutchik's taxonomy of emotion to facilitate comparison to other work and ensure that our taxonomy captured the breadth of possible expressions of emotion. We also added codes for intensity (low, neutral, and high) and valence (positive and negative) to allow for more coarse analysis and comparison to prior sentiment analysis research, which typically focuses on positive and negative sentiment. We do not address these codes in this paper. The resulting taxonomy includes 40 different categories of affect. Some of these categories – such as *interest*, *considering*, and *agreement* – are cognitive aspects of interaction that are closely linked with an affective component in the way that they are communicated or expressed by members of the group. This inclusive and flexible aspect of our taxonomy ensures that it captures the broad range of affective expressions that influence group dynamics.

With the taxonomy solidified, a team of three primary coders and five additional coders, all part of the research team, coded about 5% of the data set over a period of about 8 weeks. Of 27,344 messages coded, 15,942 (58%) were coded as 'no affect' by at least one person. About 18,000 messages were coded by exactly one rater, the rest by up to 5 raters. Most of the affect codes were used too infrequently for successful machine learning: Figure 1 shows the most

commonly applied 13 affect codes, which we focus on throughout this paper.

In many cases, multiple affect codes may apply, such as *annoyance* and *frustration* applying to these three messages, sent by the same person: "Did I see a bunch of = vs = = in there??? / WHAT / WHO DID THAT". These coincident codes sometimes overlap in meaning. At other times, multiple codes with distinct meanings may apply: *anger and confusion* were applied to a conversation about error-prone software. Example messages where this expressiveness is especially useful are included in Figure 2. Of those messages coded, 1,599 were coded with more than one affect code by at least one coder (129 were coded with more than two affect codes by at least one coder).

Reliability

Several characteristics of our data make reliability calculation challenging. There are between 1 and 5 raters per message, which can be modeled as missing data. Since the 40 codes of interest are subjective, not mutually exclusive, and may be conceptually overlapping, the criteria for agreement are atypical: non-matching codes may sometimes reflect a degree of agreement instead of disagreement. If we are to estimate the reliability of the taxonomy as a whole, rather than the reliability for each code independently, then it becomes a problem that raters could apply more than one code per message.

One of the most widely used reliability statistics, Cohen's kappa [7], is not easily applied to this data because of the variable number of coders and the large number of codes. A multitude of variations exist, such as Fleiss' kappa [8] which can work with more than two coders, but cannot handle missing data (i.e. variable number of raters). Krippendorff's alpha [13], one of the most flexible reliability statistics, handles variable numbers of raters and is nearly appropriate, but does not work with non-mutually-exclusive categories.

After studying the available techniques, we decided to modify and extend Cohen's kappa. Instead of analyzing the entire taxonomy of affect with its overlapping categories at once, we examined the reliability of each code separately. This also had the advantage of providing useful code-level information. However, we wanted to carefully control the criteria for agreement and disagreement between coders. We also wanted to use all of the data coded by multiple raters, regardless of how many raters had coded it. Although Cohen's kappa has known weaknesses [13], it is also widely understood, so we developed a way to compute a version of kappa over our coded data.

In general, kappa represents the percent agreement over data points, corrected by the probability of chance agreement. To compute it, we needed to be able to calculate two quantities: the percent of the data points where coders are observed to be in agreement, and an estimate of the probability of coders agreeing by chance. We defined agreement about a particular code on a single chat message in the following way: if more than half of raters said that the code was present, or if all of them said it was absent, then they are in agreement. Otherwise they are in disagreement.

This allows the computation of the percentage of observed agreement, in the manner of Cohen's kappa. Computing the probability of coders agreeing by chance directly is more complex because of the variable number of raters and variable number of codes.

We developed an estimate of the probability of chance agreement based on a Monte Carlo method. Similar to the calculations for Cohen's kappa, we first calculate the marginal probabilities of choosing each code for each individual coder, and the marginal probabilities of applying specific numbers of codes for each individual coder.

Code	Obs. % Agreement	Prob. Chance Agreement	Kappa
interest	0.925	0.609	0.808
amusement	0.933	0.827	0.611
agreement	0.954	0.909	0.491
considering	0.931	0.864	0.49
confusion	0.906	0.755	0.615
acceptance	0.941	0.828	0.657
annoyance	0.929	0.693	0.77
apprehension	0.876	0.737	0.529
supportive	0.961	0.906	0.583
surprise	0.968	0.93	0.543
anticipation	0.942	0.9	0.424
serenity	0.923	0.808	0.602
frustration	0.971	0.935	0.55

Table 1. Kappa statistics showing the reliability of 13 affect codes in our data set.

Next, we randomly simulate the rating of a very large number of messages. For each simulated message, we randomly generate the codes that the raters will apply, sampling from the pre-computed distributions. Counting the number of these simulated messages where agreement occurred (according to our definition) allows us to estimate the probability of random agreement. The Monte Carlo simulation continues until all probability estimates are stable to within 0.0001, generally requiring about 2 million messages to be simulated.

Finally, we calculate kappa in the usual way, given the rate of observed agreement and the probability of chance agreement. Table 1 shows the kappa values for each of the 13 affect codes that we discuss in this paper. Our kappa values ranged from 0.424 to 0.808, which is generally comparable to reliabilities obtained in previous research coding for affect or emotion [2,21].

OUR APPROACH

Our purpose in classifying affect is to automatically apply affect labels to our entire chat dataset with reasonable accuracy. With a sufficient amount of chat data coded for affect, we began developing a pipeline of data processing steps and classifier configurations, seeking the configuration with the strongest precision and recall for as many different categories of affect as possible.

Because most of the affect codes in our taxonomy still had far too few examples to use for machine learning, we focused on only the 13 affect codes that were manually applied over 300 times (Figure 1). This included a mix of positive and negative codes, as well as some closely related codes (e.g. *frustration* and *annoyance*).

We have already mentioned several significant challenges to successful classification of chat messages using current text classification algorithms. Facing these problems required numerous choices throughout the classification pipeline, including preprocessing steps, features, and classification algorithms. In order to explore this vast space to find the most successful overall pipeline, we ran a series of experiments designed to test each aspect of the pipeline in isolation. Combining these results yielded the findings we present in this paper.

We describe the options considered at each step in the pipeline, explaining how we have dealt with the challenges specific to chat messages and which choices were supported by experimental results.

Experiment Setup

In order to maximize our efficient use of the limited truth data, we used 10-fold cross validation [32] for all experiments. This helps to avoid making decisions that work well on training data but do not generalize well. Here we also describe our preparation of truth data and our software architecture.

Data preparation

For classification, we transformed the manually-labeled chat messages into training sets, one for each code, with examples labeled "present" or "absent" for each code. In order to do this, we had to decide how to use labels that raters disagreed on, and how to account for the much larger number of negative examples rather than positive for each code.

As previously mentioned, raters frequently, and often justifiably, disagree about what affect is present in a given message. For the purposes of creating truth data for classification, we assigned a given affect label in the truth data to any messages where any of the raters applied that affect code. Reliability was low for some affect codes, reflecting low internal consistency within the raters, but not low validity (i.e. low inter-rater reliability does not imply that the codes fail to capture affect). The process by which the taxonomy was developed and applied is the best assurance that the codes we used actually reflect affect expression in the data.

For all of our affect categories, there are far more messages without a given affect code (negative examples) than there are messages where that code applies (positive examples). The imbalance in the labeled data is a common problem in machine learning tasks. In such situations, it is difficult for a classification algorithm to do better than could be obtained by simply guessing the majority class. For this reason, there are many established methods for balancing data sets. We experimented with different balancing strategies including up-sampling (randomly duplicating positive examples) and down-sampling (randomly removing more-common negative examples). We found that down-sampling led to more consistent results across different affect codes.

In the experiments reported below, we applied downsampling both to the training sets and to the held-out sets in cross validation. Thus, the performance would be different on real unlabeled data where the percent of positive examples is very low. In order to evaluate classifiers with realistic unlabeled data, the relative importance of minimizing false positives vs. minimizing false negatives must be determined. These decisions depend heavily on the purpose for which labels are needed, and will vary from one project to another. We decided to downsample the held-out sets for this paper because the results generated are agnostic of the specific project context and are more easily interpreted.

Software architecture

Key to our rapid exploration of different choices for the machine learning pipeline was the use of a relational database for data storage and manipulation. Our web-based chat coding application, unlike most commercial coding systems, uses a MySQL server for storing both messages and applied codes. Our software for running machine learning experiments connects directly to this database. It is implemented in Java and relies on the popular Weka library for implementations of classification algorithms and data processing [12].

Chat Segmentation

Two significant challenges in classifying chat messages are that the messages are extremely short and highly dependent on context. We developed a segmentation procedure that reduces the number of negligibly small data points and incorporates message context.

Sparsity of feature space

Many approaches to text classification rely on "bag of words" features. The ordered list of words in the raw text is simplified by discarding all the ordering information, leaving only the number of times each word occurred in the document. This approach has been successful for many text analysis problems [32].

A corpus of text documents generally uses a large number of different words, while each individual document uses relatively few. Thus, the feature space produced by bag of words features has a high dimensionality, but is very sparse since each document has a value of 0 for most of the features. Moreover, the overlap in words between any two documents is likely to be small. Under these conditions, many machine learning algorithms are prone to overfitting. That is, they might detect and learn patterns in the data that do not generalize beyond the specific group of documents used for training.

With chat messages, this effect is even more pronounced. The average length of messages in our data is 26 characters, or about 5 words. This means that the bag of words created from each data point probably contains only a miniscule fraction of the total vocabulary in the corpus. There are likely to be many spurious patterns created by the random coincidence of these words in messages, making the patterns actually relevant to classification more difficult to detect.

Contextual dependence

Much of the information about individual messages is not present in the messages themselves, but rather in the surrounding context. Messages are often not understandable without reading many lines before (and sometimes after) in the logs, posing a challenge to message-by-message classification.

In classification problems, the data points being classified are typically assumed to be independent of one another. Yet, chat messages rarely stand on their own. Approaches which do support learning of labels in context of their surroundings, such as hidden Markov models (HMMs) or more general graphical models, are available but not commonly used for affect identification in text. We discuss our approaches to accounting for context in the following sections, leaving experiments with graphical models like HMMs for future work.

The segmentation procedure

To deal with small message lengths and to help capture message context, we split the data into segments, combining messages based on their proximity in time.

Combination of messages was determined by a simple time threshold: if two messages were separated by less than the threshold, they were grouped together. We evaluated the classification performance achieved with different segmentation configurations. Because many consecutive messages in our data set were separated by less than 25 seconds, we tested nine different time thresholds from 5 to 45 seconds at 5 second intervals.

We also developed and evaluated two different formulations of the segmentation procedure. One formulation grouped together messages by different speakers, reflecting an assumption that affective state is distributed among all of the chat participants. The second formulation did not combine messages by different speakers, presuming that affective state is bound to individuals.

Regardless of the time threshold selected, segmentation resulted in a significant reduction in data set size due to the combination of data points: a conservative threshold of 10 seconds, not separating by participant, halved the number of data points. The effect was less pronounced when we maintained separation between participants: at 10-second-segmentation, the dataset only shrank to two-thirds of its original size. In general, reducing the amount of training data makes machine learning more difficult.

However, we also observed that segmentation did have the desired effect: the number of negligibly small messages (40 characters or less, spanning only a few words) in the data set decreased, having been combined with other messages. We hypothesized that the higher word-count per data point would improve classification results. Additionally, the segmentation procedure makes contextual information available, because the messages in the immediate context of each data point are pulled in and combined.

In our experiments, the effectiveness of segmentation procedures varied from one affect code to another. For some codes, performance differences between different time threshold settings were as great as 10%. For most of the 13 affect codes we tested, the best classification performance was achieved with a time threshold of 30 seconds, keeping messages from different participants in different segments. We believe that this threshold balances the harmful effects of reducing the data set size against the benefits of increasing the size of data points.

Enriched Feature Space

Face-to-face communication relies on facial expressions and tone to communicate affect. Without these channels, chat participants use other means to communicate affect. We developed a rich set of features to help capture these aspects of chat messages, including pronoun categories, punctuation, emoticons, spelling changes, and the words in the message.

Linguistic challenges

Our data set, and chat communication generally, is rife with informal language and atypical spelling and punctuation. Unfortunately, many successful techniques for automatically analyzing text, such as LIWC [30], rely on one or more characteristics of standard written language, such as a reasonably correct vocabulary, correct grammar, and predictable punctuation.

For example, in converting a text document into a sequence of words, the text is usually split up at spaces and punctuation characters (although other techniques do exist). Indiscriminate use of this technique on chat communication risks obliterating much of the interesting content. Emoticons and other nonstandard punctuations (*e.g.*

The telescope is stuck! >:(	The exclamation point and emoticon suggest *frustration*.
The telescope is stuuuuuuuuuck...	Repetition of the letter "u" suggests *annoyance*.
The telescope is stuck??	Multiple question marks suggest *confusion.*

Figure 4. The addition of one of our features dramatically changes the meaning of the phrase.

Message Information (4 features)
duration: the length of the segment in seconds
length: the number of characters in the segment
characters per second: length / duration
rate: the average rate of messages in the segment
Pronouns (7 features)
of **1st person singular** pronouns: I, me, my, mine…
of **2nd person singular** pronouns: you, your, yours…
of **3rd person singular** pronouns: she, he, hers, his…
of **1st person plural** pronouns: we, us, ours…
of **2nd person plural** pronouns: you all, yourselves
of **3rd person plural** pronouns: they, them, theirs, their…
of **interrogative** pronouns: who, whom, whose
Punctuations (8 features)
and length of **ellipses**
and length of **question marks**
and length of **exclamation points**
and length of **?!s and !?s**
Special strings (3 features)
of **negation words**: no, not, cannot, aren't, can't…
of **swear words**
of **known people names** (list of about 18)
Low-level spelling features (8 features)
and length of **capital letters**
and length of "**hmmm**"-variants: hm, hmm…
length of **laughter** phrases: lol, hehe, heehee, haha…
and length of **repeated letter sequences** 3 or longer
Emoticons (varies, around 8-15 features)
Counts the **number of each emoticon** in the message.
Vocabulary of up to about 2200 (marshall.freeshell.org/smileys.html)
Emoticons must occur in at least 10 data points to be counted
Bag of Words (varies, around 200-300 words usually) Stemmed (Porter) and lowercased

Figure 5. A detailed list of features used.

"??????!") carry a great deal of meaning in chat, but would typically be removed or distorted during this process.

As noted earlier, lexicon-based approaches to affect detection and sentiment analysis in text [10,19,28,31] are difficult to adapt to communications with frequent nonstandard spellings and abbreviations. Because of these irregularities and because of their short length, typical chat messages may contain few if any words that are recognized by these tools. In our data, there is also a large amount of jargon and a mix of multiple languages, making the application of lexicon-based techniques especially challenging.

Features

We included duration, length, and rate of messages as features to capture this aspect of chat communication. A conversation over a short period of time, with a high rate of messages, could signal urgency or anger, such as a problem with the telescope. Whereas a conversation over a longer period of time and a lower message rate could signal one of the participants explaining a process to another, or a less stressful event.

Grammatical markers, such as personal pronouns, and punctuation, unusual spelling, and emoticons may also communicate affect. For example, the statement "*the telescope is stuck*" can have a markedly different character when expressed using various non-verbal cues embedded in the text (Figure 4).

In addition to the above features, we included more traditional bag-of-words features, based on Weka's *StringToWordVector* filter. Certain words occur more often with specific affect codes. For example, *confusion* is very often linked with phrases containing variations of "confuse," such as "confusing" and "confused." *Amusement* is often paired with phrases containing "haha," as in "You should just live there hahaha" or "Did you have enough coffee this morning? Haha."

Evaluation of feature sets

We experimented with two different configurations of the bag-of-words features: one using the Porter stemming algorithm (reducing each word to its base form), and one removing stopwords (words like "and" or "the"). Word stemming had no noticeable effect on classification performance, but the removal of stopwords consistently *decreased* performance by 2 to 3% for most affect codes we tested. Based on these results, we decided to use a stemming algorithm, but no stoplist, to generate the bag of words.

In order to determine the value of the other features outside of the bag of words, we measured the performance difference between a data set prepared with standard bag-of-words features (a large set of about 1.5k words), and similar data sets that were augmented with sets of additional features (such as punctuation, pronouns, or emoticons). Performance improvements of a few percent from each of these new types of features in isolation prompted us to continue developing and improving them, and to combine them into the extensive set of rich features summarized in Figure 5. We also applied additional reduction techniques to the bag-of-words features, such as a minimum frequency threshold and lowercasing.

Classifier Configuration

Aside from the data preparation procedure and the set of features to be used in classification, we considered a variety of options for the classification algorithm itself.

Our taxonomy of affect provides a multitude of categories into which chat messages can be classified. There are

several ways that we could formulate this as a multiclass classification problem, where we would produce a single trained classifier that selected one from among the 13 affect codes for each message submitted. However, because our categories are not mutually exclusive, we decided to create a separate binary classifier for each of the affect codes tested. We plan to experiment with other configurations in future work.

Classifier	F-measure	Precision	Recall	Accuracy
Naïve Bayes	0.650	0.637	0.691	0.637
Logistic Reg.	0.730	0.731	0.731	0.730
SVM (SMO)	**0.759**	**0.766**	**0.751**	**0.761**
C4.5 (J48)	0.700	0.724	0.680	0.710

Table 2: Performance comparison of classification algorithms from preliminary experiments, averaged over 4 runs of cross validation for each of the 13 codes tested.

Our early experiments used Weka [12] to test a variety of classification algorithms including Naïve Bayes, C4.5 decision trees, support vector machines (SVM), logistic regression, voted perceptron, boosting and bagging. We tested different parameter configurations for each of these algorithms, devoting more attention in subsequent experiments to those with more promising initial results. In these initial experiments, we found that linear-kernel SVM and logistic regression were quite effective (Table 2), which is consistent with prior results [9]. Our later experimental setups focused on configurations of SVM and logistic regression classifiers, and included Naïve Bayes and decision tree approaches for comparison.

Code	F-measure	Precision	Recall	Accuracy
interest	0.925	0.925	0.926	93%
amusement	0.734	0.78	0.694	75%
agreement	0.779	0.813	0.748	79%
considering	0.761	0.774	0.749	76%
confusion	0.738	0.743	0.733	74%
acceptance	0.773	0.805	0.743	78%
annoyance	0.642	0.668	0.618	66%
apprehension	0.638	0.657	0.619	65%
supportive	0.626	0.66	0.596	64%
surprise	0.71	0.789	0.645	74%
anticipation	0.748	0.743	0.753	75%
serenity	0.663	0.74	0.601	69%
frustration	0.673	0.734	0.621	70%

Table 3: Classifier performance for each of the top 13 codes.

RESULTS AND DISCUSSION

In this section we describe the performance of our classification pipeline on each of the 13 most commonly occurring affect codes in our taxonomy (Table 3). We also discuss which features were most useful for classifying each of the affect codes we tested (Figure 6).

Classification performance

The results of evaluating the SVM classifier with 10-fold cross validation for the top 13 affect codes are provided in Table 3. Accuracy for most affect codes fell in the 70-80% range. The SVM algorithm is designed to be robust to large feature spaces, which are typical in text classification applications [32]. Others working on affect classification in text have also found SVMs to be effective [26,33].

Predictive features

Of particular interest to researchers studying affect in collaborative text chat are the specific signals that chat participants use to communicate affect. We examined the weight vectors produced by the linear SVM training algorithm to better understand which features were most influential. Note that this analysis is limited in that it does not allow quantitative comparison from one trained classifier to another. However, it does reveal the most significant features for each affect code, as modeled by the classifier (Figure 6).

The top features span the variety of types of features; however, different codes are associated not only with specific features but specific feature types. For example, *amusement* is most clearly indicated by various emoticons. However, most of the other affect codes do not strongly rely on emoticons. Meanwhile, the *anticipation* code chiefly uses bag-of-words features, words that are often used when discussing the future. In contrast, more immediate, active affect codes, such as *frustration* and *surprise* are based mostly on punctuation, message rate, and low-level features.

Pennebaker et al. have had success using functional linguistic cues to detect emotional content [30]. In everyday speech, there are words we use that carry informational content (the semantics of what we mean to say) and those that make the utterance sensible. The latter, functional elements of text are comparatively more meaning- and context-agnostic. In contrast, EmpathyBuddy relied on real-world knowledge, extracting emotional content from semantics [15]. The evaluation of the tool did indicate that the method was able to identify some emotional content.

For some of the affect categories that we analyzed, entirely non-semantic cues like capitalization and punctuation appear to be the most important. Semantic cues, including smiley faces and content words that carry meaning, are more useful for other codes.

In prior work, it has been common for some categories of affect, e.g. negative sentiment [31], to be more easily classified than others. Our results suggest that these types of differences may stem from these researchers' different choices for features. In our experiments without emoticons, for example, the classification accuracy for *amusement* decreased significantly. This suggests that future

Considering	Annoyance	Frustration
“think”	# swearing	# swearing
# question marks	"pascal"	# 1st sg. pronouns
"maybe"	"--" (dash)	msg. length
ellipsis length	"all"	ellipsis length
"or"	"damn"	capital. length
hmm length	"again"	chars/second
# hmmm	"I"	# negation words
???? length	"only"	"it"
"probably"	"me"	# repeated letters
"x"	msg. length	# interrogative prns
Surprise	**Serenity**	**Interest**
# exclamation pts.	"good"	???? length
"wow"	emoticon ":)"	# question marks
msg. length	"nice"	"je" (fr.) (-)
???? length	"cool"	"sunrise"
!!!! length	!!!! length	"bert"
"oh"	msg. length	"est" (fr.) (-)
ellipsis length	"right"	"where"
# repeated letters	"too"	"wonder"
segment duration	# 1st pl. pronouns	"sunset"
"right"	_do (-)	"interesting"
Confusion	**Apprehension**	**Amusement**
???? length	"bad"	emoticon ";)"
# question marks	"something"	emoticon ":)"
"understand"	"problem"	laughter
"confus_"	"we"	emoticon ";-)"
"why"	"seem"	"fun"
"what"	"too"	laughter length
"nothing"	msg. length	"p"
"wrong"	"not"	# people names
msg. length	# 3rd sg. Pronouns	"sleep"
"thought"	# swearing	"of"
Agreement	**Acceptance**	**Anticipation**
"yes"	"ok	"hope"
"yeah"	"okay"	"if"
"yep"	"ah"	"next"
msg. length	msg. length	"should"
segment duration	# 1st sg. pronouns	"think"
"right"	"oh"	"will"
"yup"	"yep"	"try"
"agree"	# question marks	"at"
"sure"	"put"	"like"
"okay"	segment duration	"to"
Supportive		
"good"		
???? length (-)		
msg. length		
"if"		
"about"		
"the"		
"--" (dash)		
"derek"		
"he"		
"think"		

Figure 6. Top 10 features for each of 13 classes. (-) indicates that feature negatively relates to the affect code.

improvements may be obtained by continuing to develop new features, focusing specifically on the worst performing affect categories.

Methodology

Related work by Rosé et al. in the domain of computer-supported collaborative learning research has also developed analytic tools that allow people to code data more efficiently [25], as we aim to do for analysis of affect in chat messages from distributed collaborations. One key distinction of our approach is the coding scheme itself. Although we did incorporate an existing taxonomy of emotion, we also engaged in a crucial open coding process. This method is especially important for the analysis of communication of a distributed group using an evolving digital medium that influences interactions [27]. Analyzing the role of affect in distributed collaboration by examining traces created by ever-evolving technological media requires the freedom to refine and expand coding schemes. The automation step, therefore, would best serve this type of work if it did not methodologically require a complete and stable taxonomy of manual labels.

In discussing the methodological validity of automating coding, Rosé et al. raise the issue that automated methods base their classifications on the most predictive features, which may not be relevant to the cognitive process of human coding. For this reason, it is important to use algorithms that have interpretable learned mechanisms; optimizing for quantitative performance metrics is comparatively less important than maintaining a grasp on the methodological reasonableness of the classification itself. We reported the most predictive features for the SVM classifier, but not all algorithms afford even this level of transparency. We plan to consider how to surface this kind of information, as well as allow more transparent access to the results of classification, in future work.

Gill et al. suggest that successful social engagement relies on understanding the experience and emotional cues of others, noting the challenge of doing this in a relatively impoverished computer-mediated environment [10]. Our analysis of influential features for each of the affect categories offers clues as to how different affect states might be expressed and experienced uniquely in text chat. Grammar use, punctuation, the length of responses, and other features, all form a part of this experience much the way facial cues, tone of voice, and body language might augment the emotions of face-to-face communication in different ways. These results suggest that the experience of affect in text-based chat environments may indeed be much richer and less impoverished than long imagined.

Limitations

In future work we will consider a more qualitative consideration of *which* errors classifiers make. It is possible, for example, that one classifier makes qualitatively worse mistakes than another, misclassifying more obvious examples, whereas a better classifier only missteps on nuanced cases that are difficult even for humans. This evaluation would require an additional step of human re-evaluation of code appropriateness.

Furthermore, some apparent classification errors may not actually be errors. Preliminary inspection of a selection of chat messages where the labels produced by the SVM disagreed with our own manual coding found that in many cases a strong case could be made that the classifier’s label actually made sense. As classification methods become

increasingly robust and accurate, reaching human accuracy in difficult problems like this one, the errors that human coders make pose a challenge. Although reliability can help to verify the internal consistency of raters, to err is human, and so perhaps classifiers ought to balance measures of confidence against models of human error to produce more verbose descriptions of labeling results than precision and recall.

CONCLUSIONS AND FUTURE WORK

The study of affect expression in distributed work can benefit considerably from traces left by digital communication media, if scalable analytic methods are available.

We contribute an application of machine learning to scale fine-grained, subjective human analysis up to a large chat log. We interpreted affect as a dynamic phenomenon by segmenting the chat data set on a temporal, per-participant basis. We further augmented a standard bag-of-words feature set with analogues of non-verbal cues, such as grammatical markers including unusual spelling and emoticons, and meta-information about the chat messages such as duration, length, and rate. These decisions led to better classification results, though other avenues of exploration may offer additional improvements.

We were able to classify text for 13 affect codes with F-measures of 70-90%. We also produced a set of predictive features for each of the 13 types of affect considered, which may be applicable in other domains. We have made our machine learning software, *ALOE*[2], open source to facilitate validation, comparison to other techniques, and further research on affect in chat messages within the CSCW community.

ACKNOWLEDGMENTS

We thank the reviewers for their feedback which greatly strengthened this paper, and we thank the scientists of the SNfactory collaboration. This work was funded in part by an NSF Graduate Research Fellowship in Computer Science.

REFERENCES

1. Amabile, T.M., Barsade, S.G., Mueller, J.S., and Staw, B.M. Affect and Creativity at Work. *Administrative Science Quarterly 50*, 3 (2005), 367–403.
2. Aman, S. and Szpakowicz, S. Identifying expressions of emotion in text. In *Text, Speech and Dialogue*. 2007, 196–205.
3. Aragon, C., Poon, S., Monroy-Hernandez, A., and Aragon, D. A Tale of Two Online Communities: Fostering Collaboration and Creativity in Scientists and Children. *Proc. C&C 2009*, ACM (2009), 9–18.
4. Ashforth, B.E. and Humphrey, R.H. Emotion in the Workplace: A Reappraisal. *Human Relations 48*, 2 (1995), 97–125.
5. Barsade, S.G. The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly 47*, December (2002), 644–675.
6. Charmaz, K. *Constructing grounded theory: A practical guide through qualitative analysis*. SAGE, London, 2006.
7. Cohen, J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement 20*, (1960).
8. Fleiss, J.L. Measuring nominal scale agreement among many raters. *Psychological Bulletin 76*, 5 (1971), 378–382.
9. Gilbert, E. Phrases that signal workplace hierarchy. *Proc. CSCW 2012*, ACM Press (2012), 1037.
10. Gill, A.J., French, R.M., Gergle, D., and Oberlander, J. The language of emotion in short blog texts. *Proc. CSCW 2008*, (2008), 299–302.
11. Grandey, A. Emotions at Work: A Review and Research Agenda. In *Handbook of Organizational Behavior*. SAGE, London, 2008.
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter 11*, 1 (2009), 10.
13. Hayes, A.F. and Krippendorff, K. Answering the Call for a Standard Reliability Measure for Coding Data. *Communication Methods and Measures 1*, 1 (2007), 77–89.
14. Keshtkar, F. and Inkpen, D. Using sentiment orientation features for mood classification in blogs. *Proc. NLPKE 2009*, IEEE (2009), 1–6.
15. Liu, H., Lieberman, H., and Selker, T. A model of textual affect sensing using real-world knowledge. *Proc. IUI 2003*, (2003), 125–132.
16. Mentis, H.M., Reddy, M., and Rosson, M.B. Invisible emotion. *Proc. CSCW 2010*, ACM Press (2010), 311–320.
17. Milliken, F.J., Bartel, C.A., and Kurtzberg, T.R. Diversity and creativity in work groups: A dynamic perspective on the affective and cognitive processes that link diversity and performance. In *Group creativity: Innovation through collaboration*. Oxford University Press, New York, 2003, 32–62.
18. Mishne, G. Experiments with Mood Classification in Blog Posts. *Proc. Style 2005*, (2005).
19. Mohammad, S. Portable Features for Classifying Emotional Text. *Proc. NAACL-HLT 2012*, (2012), 587–591.

[2] http://depts.washington.edu/sccl/tools

20. Moore, B.S. and Isen, A.M. *Affect and Social Behavior*. Cambridge University Press, 1990.
21. Neviarouskaya, A., Prendinger, H., and Ishizuka, M. Affect Analysis Model: novel rule-based approach to affect sensing from text. *Natural Language Engineering 17*, 01 (2010), 95–135.
22. Plutchik, R. *The Emotions*. University Press of America, Lanham, MD, 1991.
23. Plutchik, R. The Nature of Emotions. *American Scientist 89*, 4 (2001), 344–350.
24. Poon, S., Thomas, R.C., Aragon, C., and Lee, B. Context-Linked Virtual Assistants for Distributed Teams: An Astrophysics Case Study. *Proc. CSCW 2008*, ACM Press (2008).
25. Rosé, C., Wang, Y.-C., Cui, Y., et al. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *Computer-Supported Collaborative Learning 3*, 3 (2008), 237–271.
26. Russ, S.W. *Affect and Creativity: The Role of Affect and Play in the Creative Process*. Routledge, 1993.
27. Scott, T.J., Kuksenok, K., Perry, D., Brooks, M., Anicello, O., and Aragon, C.R. Adapting Grounded Theory to Construct a Taxonomy of Affect in Collaborative Online Chat. *Proc. SIGDOC 2012*, (2012).
28. Strapparava, C. and Valitutti, A. WordNet-Affect: an affective extension of WordNet. *Proc. LREC 2004*, (2004).
29. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics 37*, 2 (2011), 267–307.
30. Tausczik, Y.R. and Pennebaker, J.W. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology 29*, 1 (2009), 24–54.
31. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology 61*, 12 (2010), 2544–2558.
32. Witten, I.H., Frank, E., and Hall, M.A. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011.