Untangling Topic Threads in Chat-Based Communication: A Case Study

Dr. Sowmya Ramachandran[1], Randy Jensen[1], Oscar Bascara[1], Tamitha Carpenter[1], Todd Denning[2], Lt Shaun Sucillon[3]

[1]Stottler Henke Associates Inc.
[2]AFRL/RHA
[3]AFRL
{sowmya, jensen, bascara, tamitha}@stottlerhenke.com
todd.denning.ctr@nellis.af.mil
shaun.sucillon@wpafb.af.mil

Abstract

Analyzing chat traffic has important applications for both the military and the civilian world. This paper presents a case study of a real-world application of chat analysis in support of team training exercise in the military. It compares the results of an unsupervised learning approach with those of a supervised classification approach. The paper also discusses some of the specific challenges presented by this domain.

Analyzing microtexts for topic identification has many applications such as gauging the current topics that are capturing our collective attention, or discovering what people are saying about products, for example. While most of the focus of microtext analysis has been in the service of business intelligence, it has pretty powerful applications in the educational domain. Topic identification can be used to analyze chat messages exchanged in context of an educational application to track the focus and depth of student interactions. Chat analysis is also a powerful tool for assessing team communications when used in a training or operational context.

Unsupervised learning techniques such as clustering are very popular for analyzing text for topic identification (Anjewierden,, Kollöffel and Hulshof 2007; Adams and Martel 2008). These techniques have several attractive features, the most significant being that they do not require labeled training examples. This however is also a disadvantage under some circumstances. Without the guidance of labels and supervised learning algorithms, clustering approaches can discover concepts that, while distinct, are not relevant to the analysis objectives. One way around this is to carefully engineer the features used to represent the data in order to guide the discovery. However, whatever you might have gained by not having to label examples you might have lost in feature engineering. Supervised learning algorithms for chat analysis (Banerjee and Rudnicky 2006; Herring 2006; Shi 2006) on the other hand can be guided towards concepts of interest via labeled training examples, but typically labeling the examples requires considerable human input. If the concepts being learned are expected to be stable over time and generally applicable to a wide range of analysis domains or areas of analysis, the upfront cost of hand labeling examples may be acceptable. When this is not the case, these costs can be prohibitive.

Such was the case in the analysis problem we are considering. We are currently developing a chat analysis tool called IDA (*Intelligent Diagnosis Assistant*) for use with simulation exercises for training operational planners in the Air Operations Center. The purpose of the tool is to help trainers analyze chat communications between among team members to assess how closely they followed the TTPs (Tactics, Techniques and Procedures). The trainers specifically required that the tool be able to separate the messages in the chat logs by topics (what topic means in this context will be explained later in this section).

This analysis problem falls in the space that presents challenges to both the supervised and unsupervised paradigms. On the one hand there is a pre-determined set of topics that is of interest and therefore the analysis has to be directed. On the other hand, these topics are not stable and likely to change from one exercise to another. The costs of hand labeling training data for every exercise would more or less nullify the benefits of automated analysis. There are few options here; one can either try to find ways to constrain unsupervised algorithms towards the concepts of interest or you can find other sources of data labels. We got lucky; typically the planners mention the mission name explicitly in a small subset of their chat messages. This gave us our set of labeled training examples. Additionally, there is an exercise database where the planners record important decisions with respect

to a mission. Often they will cut and paste messages as annotations. While this constitutes a small fraction of the chat database, they were still useful as training examples and contain high information content.

While this particular analysis context problem was able to provide us with low-cost labeled training examples, this presents a space of problems with unique challenges. These conditions are likely to be manifested in similar contexts in other domains.

In this paper, we will report on techniques for automatically identifying topic threads in chat-based conversations. This work is in support of the research at the Air Force Research Lab at Mesa, AZ and is aimed at improving team training outcomes by developing exercise visualization and debriefing tools that will help trainees and trainers.

The Training Research Exercise (T-REX) environment at the Air Force Research Laboratory allows mission-ready warfighters to practice their assigned duties using real-world systems in a scenario designed to test the full spectrum of decisions and coordination required in operational planning. The suite of systems includes collaborative planning tools such as chat rooms and shared databases. T-REX addresses the task of operation planning and execution within a Dynamic Effects Cell. The initiator for planning is normally a problem statement in the form of intelligence data or operational data reported to the team. The team then examines the problem in sequence with other planning tasks or a sub-team may be tasked to examine the issue in parallel with other team activities. Most of the team communication occurs via chat. Since the team handles multiple planning problems simultaneously, conversations regarding these missions are concurrent and highly interleaved.

At the end of each training session, instructors lead a group after-action review (AAR) session to reflect on performance. The main factors to consider are how the team addressed each planning task or mission separately, and how the team handled multi-tasking. The ability to evaluate chatlogs during the review is important. Additionally, it becomes necessary to isolate and view the chat messages according to the topic/mission. Since each training session can run for four hours and generate on the order of a thousand chat messages spread across twenty five chat rooms, tools to filter and navigate the data to quickly zoom in on relevant bits of communication is of critical importance.

Intelligent Diagnostic Assistant (IDA) is a chat visualization and analysis tool to support team AAR during team training exercises such as T-REX. Based on a requirement analysis, we have determined that classification of chat data according to missions is an important analysis capability for IDA. While visualization is also a key aspect of objectives of IDA, we will focus on the analysis problem in this paper.

The Baseline

The problem we are addressing is:

> **Given**: A database of chatlogs from a T-REX training session and other data logged/generated during training,
> **Produce**: For each chat message, identify the mission to which it refers.

To get an insight into developing an automated solution, we interviewed an SME to understand how a human expert would perform this classification. These interviews led to the formulation of a 4-step process, each step implementing a rule. This forms the baseline approach.

Rule 1: In this domain, each mission is assigned a unique identification number (ID). Trainees sometimes, but not always, will refer to this ID while talking about a mission. When they do, it becomes easy to associate those chat messages with a mission. IDA makes one pass through the data set to identify those messages that have explicit references. These messages form the core set upon which subsequent rules build.

Rule 2: The next pass uses mission-specific keywords to classify chat messages. SMEs typically use their knowledge of the exercise events and examine associated exercise data to come up with a set of unique keywords for each mission. They would leverage these to tag chat messages. We automated this by having SMEs input mission-specific keywords via a configuration file. IDA uses these keywords to tag messages.

Rule 3: There are some types of temporal patterns that can be detected with reliable accuracy without the need to understand the content of utterances. An example is recognizing the pattern of a turn-by-turn interaction between two people in the same room (e.g. A says something to B and 3 minutes later B says something to A) and inferring that they belong to the same topic thread. Making an assumption of dialog coherence, one can say with some degree of confidence (represented in IDA by a weight that varies inversely with the distance between the pair) that such conversation dyads refer to the same topic thread. The message classifications identified using the keyword-based approach are used as the basis to further identify and tag such messages pairs.

Rule 4: Finally, locality influence is used to attempt to classify remaining unclassified chat lines. For each such line, IDA examines its neighboring messages and finds the most common mission association, weighted by distance of the neighbor from the line. If the combined influence of all the messages within that window that are associated with this mission is over a threshold, the chat line is also assigned to that mission. This rule was not found to have any significant influence on classification accuracy and we will drop it from future versions of IDA.

While the baseline approach mimics the analysis of a human expert, it has the disadvantage of requiring manually provided keywords. These might vary by training sessions; requiring SMEs to provide keywords before each and every after-action review will place an undue burden on them. Our next objective was to replace Rule 2 with a step that automatically finds content-based similarity to classify chat messages.

Clustering For Topic Identification

We tried two different similarity-based clustering approaches to group chat messages together according to topic. One used the term frequency-inverse document frequency (TF-IDF) similarity measure presented in (Adams and Martell, 2008). Here similarity is determined by the number of overlapping words between two messages, weighted by the uniqueness of the words.

We used a hierarchical clustering algorithm where each chat message is matched with its nearest neighbor. If one message from this pair is already a part of a cluster, the other is added to it. If each message in the pair belongs to different clusters, these are merged. If neither belongs to a cluster, then a new cluster is created. We modified this basic algorithm to include stemming, filtering of stop words, and a moving chat window. The stop word list consisted of the hundred most common words in English. It also included all the call signs that are used by the team to identify each member. A chat window was introduced in an effort to localize the clusters based on the observation that topics typically consist of subtopics that shift over time. With this modification, each message was paired with a nearest neighbor occurring within a surrounding window (currently set to include 10 messages occurring before and 10 messages after the one under consideration). Finally, the algorithm ignores messages with less than three words (as most of these are related to message acknowledgments).

Once clusters are identified, the algorithm then assigns to each cluster a topic label based on Rule 1. One of the following is true about each cluster identified: 1. None of the messages in the cluster were assigned a label by Rule 1, 2. Some messages in the cluster were labeled by Rule 1 and all them are identified with the same topic, 3. Some messages in the cluster were assigned labels by Rule 1 and they are identified with different topics. In the first case, the cluster itself is not assigned any label. In the second case, all the messages assigned to this cluster are assigned to the topic identified. In the third case, the cluster is disregarded because it represents multiple topics and therefore not considered relevant.

In addition, we also tried the Latent Dirichlet Allocation (LDA) approach method (Blei et. al. 2003) for clustering. We used an off-the-shelf package that implements LDA.

Preliminary results indicated that neither of these approaches improved classification accuracy. A closer examination found that these algorithms resulted in clusters that were not unique to missions. For example, one cluster that they found had messages that were about expected time on target (ETOT). While this is an interesting classification, such discussions are equally applicable to all missions and therefore are not relevant to the particular analysis objectives. We tried some feature engineering, but abandoned the effort as the engineering was found to be getting increasingly data specific.

Using Naïve Bayes Classification to Find Mission-Specific Keywords

The domain provides another related data source that can be usefully exploited. All trainees use a database system called Joint Automated Deep Operations Coordination System (JADOCS) to record critical information about the various missions, such as target intelligence, operational orders etc. A very common practice is to copy over messages from chat streams to the JADOCS database (DB) as annotations. This results in a set of chat messages stored in the JADOCS DB with definite mission associations that can be mined to learn mission-specific identifiers.

In addition to the JADOCs data, we can use the chat messages tagged using Rule 1 as training examples for the learning algorithm as these associations have a very high probability of being correct.

We modified the analysis algorithm to use Naïve Bayes classifiers that are trained on the message-mission associations found in the JADOCS and on the messages tagged using Rule 1 (Langley 1995). This is done in place of Rule 2 of the original approach. The remaining rules are applied as before.

A Naïve Bayes classifier is a simple classifier that uses the Bayes Theorem to assign conditional probabilities to classes given feature values. Its simplicity derives from the feature independence assumptions underlying the classifier. This is a strong assumption that may not hold in a lot of real-world examples. Despite this, Naïve Bayes classifiers have been found to be remarkably effective and often beat out their competition in terms of accuracy.

A Naïve Bayes classifier is typically binary. To handle multiple classes, we used a one-against-all approach, where each class has a dedicated classifier trained on a data set where the positive examples are labeled messages belonging to that class, and negative examples are labeled messages belonging to the rest of the classes. In this domain, we do allow for chat messages to be assigned multiple class labels.

As with the clustering approach, the messages are stemmed and filtered of stop words prior to training. The data is then used to train one classifier for each topic.

Unlabelled chat messages are classified by passing each message to each of the classifiers. A message is labeled with a topic/mission if the corresponding classifier assigns it a high probability (i.e. higher than a

parameterized threshold which is set to be a heuristic multiple of the prior class probability).

Results

We analyzed the topic classification accuracy of IDA on 8 data sets are from actual T-REX sessions. Each data set has an average of 800 chat messages. The numbers of topics in each data set range from 10 to 26. The distribution of the message across the classes tends to be skewed with some missions dominating the conversation. An analysis of the data to study the patterns of distribution and its impact on the classification accuracies of the various approaches is underway.

All accuracies were measured in terms of precision, recall, and F2-score for each mission. The purpose of the analysis is to filter the conversations by missions in order to reduce the amount of data that must be considered by the instructors during AAR. However, it is not essential to eliminate all the clutter; a low to moderate amount of false positives is acceptable. What is crucial, however, is to not eliminate those chat messages that do belong to the mission. Thus, false negatives are significantly less desirable that false positives. For this reason, we measure accuracy using the F2 score.

The data sets were hand labeled with message-mission associations by an SME. This formed the gold standard against which the output of IDA was evaluated.

We found that clustering combined with Rule 1 classification led to significant ($p<.001$) improvements in recall, it also significantly degraded precision and there was only a marginal (and not significant) effect on the F2-Score. Table 1 shows the paired t-test statistic for these two conditions. However, combining Rule 1 with Bayesian classification led to a significant improvement in recall. While it did also significantly degrade precision but there was an overall improvement in the F2. Table 2 shows the paired t-test statistic for this comparison. Table 3 shows the accuracy of IDA using all rules except Rule 2, and that of the system that uses all rules and the Bayesian Classifiers for Rule 2. The differences in means for precision and F2-Scores between the two conditions are significant at $p<0.001$.

It is interesting to compare the accuracy of the Naïve Bayes approach with that resulting from using SME-provided keywords. We were given keywords by an SME for only one of the data sets. Table 4 compares the accuracy of the Naïve Bayes version against hand-coded keywords on this dataset. We have seen that the Naïve Bayes approach is significantly better than using no keywords (row 3) at all. However, SME provided keywords lead to significantly superior results on this data set. Though we have yet to analyze the factors contributing to the success of the keywords, one factor may be the bag of word approach taken by IDA is not sufficient. Including word bigrams (or other n-grams) as features may help improve its classification accuracy.

Domain-Specific Challenges

Working with real world problems gives rise to a number of practical challenges. In this case, lack of access to SME time was an issue. The research reported here was performed in the context of an operational training program and understandably the training objective took precedence for the users. This left them with little time to attend to our research objectives. We could only make limited demands on their time with the result that we were limited in our ability to analyze the results of the different algorithms in greater depth. The second challenge was the classified nature of the data. With the stringent restrictions on installing third-party software on classified computers, experimenting with different approaches meant that we had to implement them ourselves. This got in the way of rapid experimentation and impacted the research scope. We were able to import an open-source package for LDA but only after spending a lot of time and effort on getting approvals.

Within the bounds of these limitations, we were able to examine the data closely to study the reasons for classification failures. The primary reason seems to be that topics, while separate, are sometimes highly correlated. For example, exercises may have one mission dedicated to a High-Value-Individual (HVI), another to the location of the HVI, and another to a planned operation targeting the HVI. Sometimes these distinctions are genuinely necessary; at other times they are just artifacts of an incomplete understanding of the process on the part of trainees. Whatever the reason, this makes topic identification challenging in the absence of a deeper semantic interpretation. Statistical techniques for natural language analysis, such as the ones discussed in this paper, are limited in this respect.

Our examination also suggests that a lack of accurate labels for the data sets for validation could also be affecting our results. We have observed that even humans familiar with the details of the exercise find it difficult to associate chat messages to the relevant missions. This makes evaluating the performance of IDA a challenge as there are no accurate ground truth classifications to serve as standards for comparison. Coming up with accurate labels seems to demand a level of time and labor commitment that were beyond the resources available to us.

Finally, we also noticed that it was somewhat common for teams to confuse missions. Leveraging automated analysis techniques to detect such confusions, either during the exercise or for after-action review, will be an interesting direction for future investigations

It must be mentioned that the above observations are based on our examination of a subset of the data and are not backed up by quantitative measures. For example, though in our estimation a significant number of ground truth labels are incorrect, we were unable to verify the

exact number of such errors with the subject-matter experts.

Related Work

Previous related research involving multi-party dialog analysis has included much work to characterize spoken interactions in multi-party meetings, social structures, and collaborative learning environments. The most relevant work is being done by the Cognitive Agent that Learns and Organizes (CALO) project, a joint effort between SRI and the Center for the Study of Language and Information at Stanford University. (Zimmermann 2006), and (Tur 2008) describe efforts within the CALO project to support multi-party meetings with transcription, action item extraction, and, in some cases, software control such as document retrieval and display updating. (Niekrasz 2004) describe an architecture in which the spoken conversation between meeting participants is processed using automatic speech recognition techniques, and grounded against the artifact being produced (e.g., a schedule, a budget) and the drawings made on an electronic whiteboard. All of these inputs are used to create an electronic version of the artifact. Although experiments with dialog models from spoken interactions are transferable to research with chat communications, there are also unique challenges with the chat medium.

Much chat-related research has focused on the inherent communication artifacts of the medium, such as the emergence of conventional abbreviations, emoticons, and other common stylistic practices. To a lesser degree, some research has yielded methods and tools to analyze or visualize chat communication patterns. Most require a coding step carried out by a human reader to tag messages or explicitly identify dependencies before analysis takes place in any automated form.

(Cakir 2005) studied methods for assessing team problem solving with a chat environment and shared workspace. Essentially this employed a structure for organizing messages and identifying instances of interactions between two, three, or more participants as well as indices for factors like initiative. This is useful for learning research observations about how level and type of participation contribute to team dynamics and collaboration effectiveness.

(Shi 2006) introduce a conceptual framework for *thread theory*, which suggests an approach for sorting out different chat threads based on topic or theme, and for characterizing defining features such as life, intensity, magnitude, and level of participation. (Herring 2006) describes VisualDTA, a tool designed to generate a visualization of a chat conversation that has been manually coded. In this visualization, messages are plotted in a descending tree, with temporal spacing represented on one axis, and semantic divergence represented on the other. The tool also accommodates the possibility of completely new topic threads appearing within the chat stream, resulting in new trees. This is useful for social interaction research, where plots of communication patterns reveal behavioral features.

(Adam and Martell, 2008) used the TF-IDF measure discussed earlier to identify topic threads in chat conversations. Their approach used only clustering whereas we have suite of other techniques to help the process. Whereas they were concerned with detecting topics in general public chat session that are not focused on any particular domain, our objectives are a little narrower. We are concerned primarily with chat conversations that are occur within military team training exercises. This gives us the benefit of leverage chat protocols, domain-specific vocabulary other data sources to help refine our technique.

Our research also shows that it is useful to look beyond the chat database for other related context information that can help learning. Fortunately, the military environment presents some significant opportunities not found in most other application areas. In particular, the continuous logging of information access and manipulation actions of users offers a rich resource that can be exploited in attributing meaning to the space of workspace objects, without the interruptions or bias associated with producer-centric manual metadata specification.

References

Adams, P.H. and Martell, C. H. 2008. Topic Detection and Extraction in chat. In the Proceedings of the IEEE Conference on Semantic Computing. Santa Clara, CA.

Anjewierden, A., Kollöffel, B., and Hulshof, C. (2007). Towards educational data mining: Using data mining methods for automated chat analysis to understand and support inquiry learning processes. In *Proceedings of International Workshop on Applying Data Mining in e-Learning (ADML 2007) as part of the 2nd European Conference on Technology Enhanced Learning (EC-TEL 2007)*, Crete, Greece.

D. Blei, A. Ng, and M. Jordan (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993-1022.

Banerjee, S, and Rudnicky, A.I. (2006). A TextTiling Based Approach to Topic Boundary Detection in Meetings. Proceedings of the Interspeech - ICSLP 2006 Conference. Pittsburgh, PA.

Cakir, M., Xhafa, F., Zhou, N., & Stahl, G. (2005). Thread-based analysis of patterns of collaborative interaction in chat. Paper presented at the Conference of Artificial Intelligence for Education (AIEd 2005), Amsterdam, Netherlands.

Herring, S. C., & Kurtz, A. J. (2006). Visualizing dynamic topic analysis. In Proceedings of CHI 2006. New York: ACM Press.

Langley, P. (1995). Elements of Machine Learning.Morgan Kaufman Series in Machine Learning. 1995

Manning, C. & Schüteze, H. (1999). Foundations of Statistical Natural Language Processing. The MIT Press. 1999.

Niekrasz, J., Gruenstein, A., & Cavedon, L. (2004). Multi-human dialog understanding for assisting artifact-producing meetings. In Proceedings of the 20th International Conference on Computational Linguistics (COLING).

Shi, S., Mishra, P., Bonk, C. J., Tan, S., & Zhao, Y. (2006). Thread theory: A framework applied to content analysis of synchronous computer mediated communication data. International Journal of Instructional Technology and Distance Learning, 3(3), 19-38.

Tur, G., Stolcke, A., Voss, L., Dowding, J., Favre, B., Fernandez, R., Frampton, M., Frandsen, M., Frederickson, C., Graciarena, M., Hakkani-Tür, D., Kintzing, D., Leveque, K., Mason, S., Niekrasz, J., Peters, S., Purver, M., Riedhammer, K., Shriberg, E., Tien, J., Vergyri, D., & Yang, F. (2008). The CALO meeting speech recognition and understanding system. In Proceedings of the 2008 IEEE Workshop on Spoken Language Technology.

Zimmermann, M.; Liu, Y.; Shriberg, E. & Stolcke, A. (2006). Joint Segmentation and Classification of Dialog Acts in Multiparty Meetings. In Proceedings of IEEE ICASSP, Toulouse, France (2006).

Only Rule 1 Vs. Rule 1 + Clustering	**Rule 1 Only**	**Rule 1 + Clustering**	**Mean difference**	**95% confidence interval**
Average Precision	0.932	0.795	0.13	[0.078, 0.195]
Average Recall	0.211	0.238	-0.027	[-0.060, 0.006]
Average F2-Score	0.249	0.277	-0.025	[-0.059 0.009]

Table 1: Differences in Classification Accuracy: Using Rule 1 vs. using Rule 1 and Clustering

Only Rule 1 Vs. Rule 1 + Bayesian Classifiers	**Rule 1 Only**	**Rule 1 + Bayesian Classification**	**Mean difference**	**95% confidence interval**
Average Precision	0.932	0.637	0.295	[0.194, 0.395]
Average Recall	0.211	0.458	-0.247	[-0.351, -0.144]
Average F2-Score	0.249	0.461	-0.228	[-0.305 -0.150]

Table 2: Differences in Classification Accuracy: Using Rule 1 vs. using Rule 1 and Bayesian Classifiers

	No keywords, No automated classification			**No keywords, With automated classification**		
Data Set	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F2-Score**
Mean Scores	0.63	0.34	0.38	0.52	0.57	0.56

Table 3: Comparison of the results from using all rules except Rule 2 and using all rules with automated classification in place of Rule 2

Classification Method			**IDA Accuracy**		
Rule-Based	**Hand Coded Keywords**	**Automated Keyword Detection**	**Precision**	**Recall**	**F2-Score**
Yes	Yes	No	0.45	0.89	0.74
Yes	No	Yes	0.66	0.57	0.56
Yes	No	No	0.45	0.49	0.38

Table 4: Comparisons between using keywords to identify topics vs. using the classifiers