Text Classification and Machine Learning Support for Requirements Analysis Using Blogs

Douglas S. Lange

Space and Naval Warfare Systems Center
San Diego, CA 92152
doug.lange@navy.mil

Abstract. Text classification and machine learning technologies are being investigated for use in supporting knowledge management requirements in military command centers. Military communities of interest are beginning to use blogs and related tools for information sharing, providing a comparable environment to the use of blogs for system requirement discussions. This paper describes the work in the area being performed under the Personalized Assistant that Learns (PAL) program sponsored by the Defense Advanced Research Projects Agency. Comparisons are then made to how the technology could provide similar capabilities for a requirements analysis environment. An additional discussion of how the task learning capabilities from PAL could also benefit requirements analysis in a rapid prototyping process is provided.

1 Introduction

The United States Military has adopted several network based communications mechanisms. During the second Gulf War, *chat* was an important method of communications, reducing the need for voice circuits. *E-mail* protocols have been used heavily since the early '90s for longer more structured messages in the place of old teletype methods. Now, *blogs* and *wikis* have come into use for knowledge sharing purposes [1].

The U.S. Strategic Command has developed and uses heavily a capability that can best be described as a hybrid of wikis and blogs. The Strategic Knowledge Integration Web (SKIWeb) allows users to post information about key events, and allows other users to add comments and edit the information. Events can be linked to other events, and lists of events are used to provide key information to various communities of interest [2]. While SKIWeb is structured differently than most blogging capabilities, the information within it can, with small transformations, be represented as being structured exactly like a collection of blogs.

With sponsorship by the Defense Advanced Research Projects Agency (DARPA), the Space and Naval Warfare Systems Center (SSC) along with SRI International and Northrop-Grumman is working to transition machine learning technology into both SKIWeb and a blog/Really Simple Syndication (RSS) capability being developed for U.S. Navy command and control. It is envisioned that the learning technology can not only aid the bloggers, reducing the labor costs of publishing information, but can also

B. Paech and C. Martell (Eds.): Monterey Workshop 2007, LNCS 5320, pp. 182–195, 2008.

extract information for other purposes. It is this second feature that is most closely aligned with the goal of extracting software requirements from blogs.

2 PAL Blogs

Machine learning from the DARPA Personalized Assistant that Learns (PAL) program is being used in several ways in conjunction with blogs. These approaches will help both those who publish information on blogs and those who subscribe to receive the information over Really Simple Syndication (RSS) feeds. Figure 1 depicts an envisioned Navy Composeable FORCEnet (CFn) PAL Blog capability. The two sides of the picture depict the publishing activities (left) and the subscription activities (right). Sections 2.1 and 2.2 describe how machine learning contributes to both sides. The figure shows the situation envisioned for CFn use.

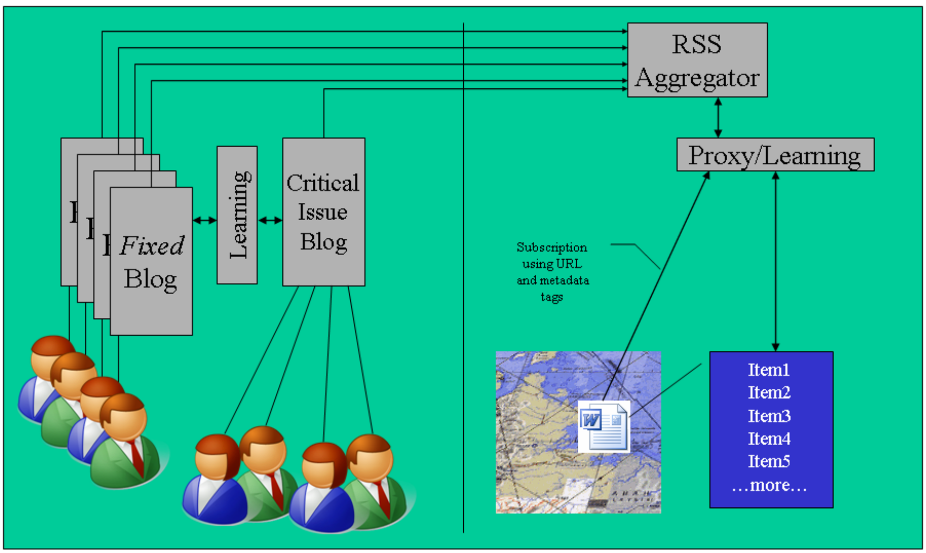

Fig. 1. PAL Blog

On the left side, each information publisher would have their own "fixed" blog. Someone managing parts orders for aircraft would publish information above and beyond what fixed databases allow. On the other hand, when a crisis occurs, multiple users may contribute to a blog about a particular critical issue. These blogs would be utilized during the life of the crisis. The goal of adding a learning capability to support publishing is to help authors find additional information held that relates to posts already made in their blogs. If an author posts information about wing parts for a particular jet, were there emails, received blog articles, documents, etc. that related to the posts. Perhaps the automated assistant can help the author publish more relevant information faster.

The right side of the diagram depicts an RSS Aggregator subscribing to a series of blogs that it has learned may contain information relevant to the user. Additional information is extracted from the display as configured by the user and documents that fall within the relevance criteria for the user. In Figure 1, items relating to a document posted on the map display are found by comparing the text within the document and the topic models learned concerning the user's interests and needs.

For SKIWeb, SKIPAL (the SKIWeb version of PAL) provides similar benefits. Using topic modeling [9], SKIPAL learns to recognize events of interest to each user and provides a recommended reading list. Text classifiers are taught to recognize particular topics of interest and SKIPAL will identify the people most likely connected to the events and other events that are most closely related. SKIPAL will also learn to gather additional information about an event and learn the tasks that must be accomplished when particular kinds of events occur.

A simple illustration of how SKIPAL communicates with SKIWeb is in Figure 2 below.

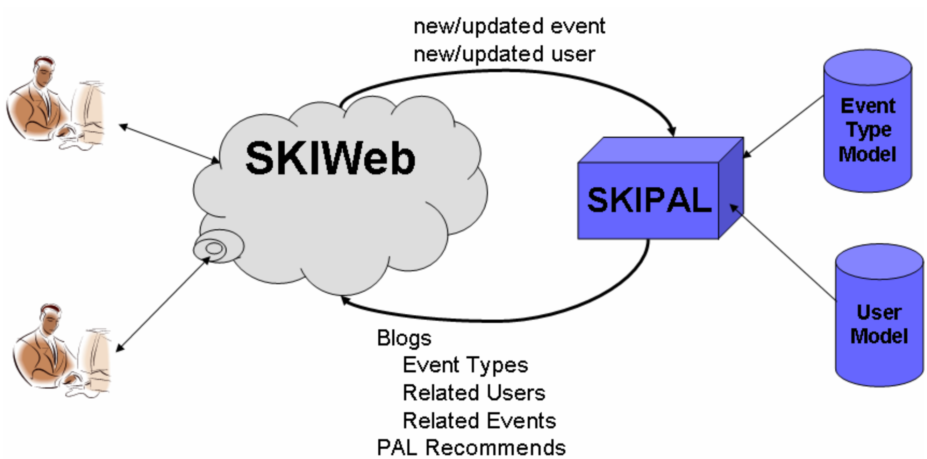

Fig. 2. SKIPAL - SKIWeb Architecture

Users enter through the *SKIPAL Recommends* page as shown in Figure 3 below.

Selecting an event brings up a SKIWeb page that has been supplemented by SKIPAL. SKIPAL shows a list of related people, related events, indicates if the event was recognized by a trained classifier, and provides amplifying links and tasks. A sample is shown in Figure 4.

2.1 Learning on the Publishing Side

Two technologies are applicable to the publishing side in the CFn capability. The first is statistical text classification. Statistical text classifiers group documents based on learning the probabilities of documents within a category containing particular words [10]. Various learning techniques can be applied to text to help map the topics found in a corpus. Various algorithms can be used [3] and the selection can depend on the characteristics of the text, and the mode by which the classifier is to be trained and used. Experience with naïve Bayesian classifiers is discussed in section 4.1.

Fig. 3. SKIPAL Recommends

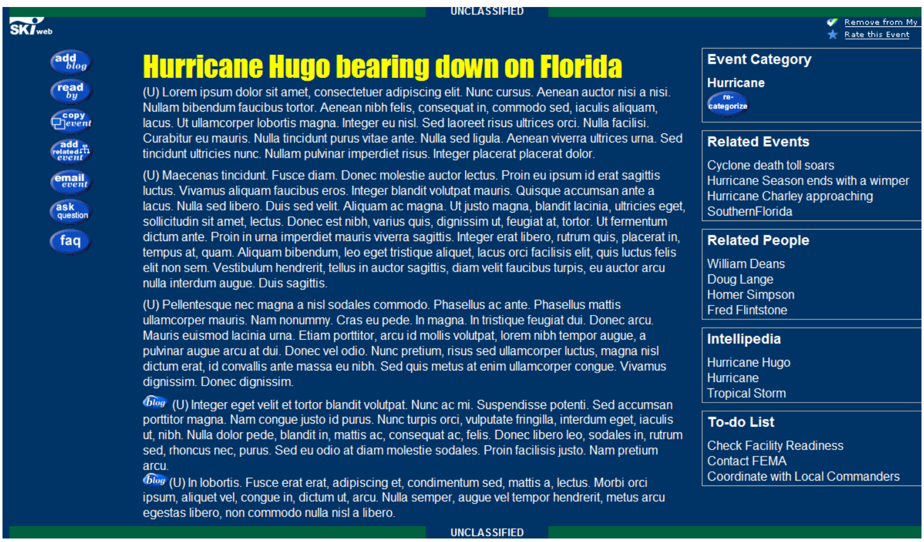

Fig. 4. SKIPAL Supplemented Event

Text classifiers occupy the *Learning* module in Fig. 1. By subscribing to the blog of a user, text classifiers can determine what topics the user writes about. If a user frequently reports the status of aircraft in his/her blog, a model of the writer's interests can reflect that. Further, through the use of intelligent search and indexing that employs the same classification capabilities, PAL may find new email, documents, chat, or other information sources that have new information on aircraft status and suggest to the user that the information be added to the blog. In Section 3, it will be argued that this capability can be useful in defining system or software requirements from blogs.

A second technology applicable for the PAL Blog provides *task learning* [4]. Another approach to helping the user publish is for PAL to learn and generalize common tasks for the user. If the user frequently gets email about the status of aircraft, he or she may typically choose to do additional research before publishing the results. Perhaps a database of parts needs to be queried and a decision aid to calculate delivery times must be run. The PAL task learning technology allows a user to teach PAL that when such email arrives in the future, PAL is to go through those steps and report the results in a particular manner on the blog, thereby relieving the user of the need to perform the task. PAL. This technology is not directly applicable to an effort to use natural language tools such as blogs as requirements sources, but task learning itself can be a means to requirements gathering by using the task models that are generated as representative of the capabilities required.

2.2 Learning on the Subscription Side

On the subscription side, the goal of learning is to model the topics being published and in turn used by the reader in order to predict what information the user would like to see and find related material. In the PAL Blog, learning is being used to observe the reading habits of users and suggest RSS feeds to subscribe to, and even which entries from a feed to treat with higher priority. This will be done with text classification methods, mapping of topics, and even social networking clues such as which bloggers provide more authoritative information [9]. The features here are nearly identical to those being fielded in SKIPAL.

In SKIPAL, events are recommended to users based on learning what types of events the user authors, comments on, and reads. Explicit feedback is utilized as well. The user can direct SKIPAL to provide more or fewer of a particular kind of event. We are experimenting with using text classifiers and a more sophisticated topic modeler for this purpose. The topic modeling software (iLink) is discussed in the next section. In section 4, results from recent experiments will be provided in order to discuss how well the tools are able to accomplish these goals.

2.3 Models of Expertise

The third technique being applied to blogging activities is social network analysis. The iLink [10] capability, developed by SRI International as part of the PAL program, learns to attribute different levels of expertise on a topic to each member of a network. This is based on the ability of participants to respond to questions satisfactorily, the likelihood that other participants in the network will route messages to them and the numbers of messages written and read on the topic. The way iLink is structured, when a user poses a question through iLink it is distributed to those who are known to have some expertise in the area. Recipients are able to answer the question or forward it to others who they feel might know the answer. When answers do come, those who have provided useful information and those who referred the question to them have their scores raised. Those who provide poor information or cannot answer have their scores lowered. In this way, expertise is determined by the quality of information rather than by simple claims in social network metadata. Social network analysis can show that the real organization of an entity may be very different than what the organization chart shows. Likewise, the expertise may not reside in those who are advertised as the experts on a topic. Similar results may show that the key stakeholders regarding particular requirements may be different than those believed.

3 Applications for Requirements Engineering

If we consider that the text of individual blogs contains information important to the requirements of a system being developed, then the use of machine learning capabilities in text classification, topic modeling, and social networking tools may provide a means to organize the statements, record arguments for and against a capability, identify critical stakeholders, and even provide some sense of priority. In this section, we will look at each of the activities being performed by machine learning technologies in the SKIPAL capability and compare them to the domain of software requirements and in particular to requirements being gathered based on blog posts.

3.1 Text Classifiers

In [7] there are only three people represented in the discussion on requirements. The power of using blogs or other network communications tools is that the number of people who could express an opinion or provide information can grow much larger. Text classification can group the posts into categories useful for requirements engineering. Those posts that relate to the "carrying of liquids" would get grouped together, much like SKIWeb events about particular subjects are grouped to help decision makers in a command find information.

Classifiers require training [10], so the method process for requirements engineering might be similar to the following:

1. Hold a limited conversation as was done in the case study, or extract a subset of the comments made.
2. Through human analysis determine the categories represented by the posts and label the text in preparation for using the classifier.
3. Train the classifier to recognize the categories using the labeled posts.
4. Run the remaining posts through the classifier and have the classifier do the labeling.
5. Use the output to see if stakeholders in general held similar or dissimilar views. In posts from outside the small initial group, if the posts were about the same topic, were the conclusions the same?
6. Investigate posts that failed to be recognized by the classifier. These may be sources of requirements not thought of by the initial group. Use these to start new discussion threads that can be analyzed by repeating the process.

Statistical methods are at their best when there are large quantities of data to work on. Classifiers would not be useful in the small case study of 3 subjects and 13 posts. But consider how blogs would allow large portions of the stakeholders to comment on many different issues relevant to the new development.

Text classification is also targeted as a technology to support the publishers of information in the PAL Blog. If each of the stakeholders involved in the discussion had access to publisher-side learning tools, they could quickly provide amplifying information for their opinions in their posts. An FAA official who had many emails indicating the weakness of a particular process or technology, or concerns about specific threats, may want to provide that information in the blog posts. The process for using the technology in this way might be the following:

1. The author posts comments on a particular issue.
2. The classifier (already trained earlier...see above), recognizes the categories that the post fits into and scans the users disk and accessible network locations for documents (letter, email, chat, etc.) that fit within the same category.
3. The user is shown the candidate items and can choose whether to publish them in the blog to support previous statements.

3.2 Topic Model Filtering

Which stakeholders have a particular interest in which topics? If we continue to assume that one reason to use blogs or other similar technology is that we want to reach a large community and we want in depth discussion of many topics, we probably want to help the stakeholders filter which posts they pay attention to.

The typical approach would be to define a priori what issues we are going to discuss with each stakeholder. The machine learning approach used in SKIPAL uses a different strategy. Everything is available to the user. SKIPAL learns based on what the user writes, reads, bookmarks and focuses the user's attention on the most relevant posts. Users are still free to look at others and by doing so improve the model of the user's interests. This allows the stakeholders to be involved in the areas they care most about.

The topic model further serves the requirements engineer by ensuring that those most concerned with a topic see questions and comments posted about it. Well before classifiers are trained and used, we can ensure that related posts are being put together and are noticed by the right people.

3.3 Social Network Analysis and Expertise Modeling

Through social network analysis, as blog entries are analyzed for good requirement content, judgments can be made about the quality of the input by engineers and used to learn the level of authority that should be attributed to individual authors both through the judgment of the engineers and by the level of agreement with those that are already judged to be authoritative.

That we expand our discussion to include tens, hundreds, or thousands of stakeholders, doesn't mean that we weight all opinions equally. The iLink capability used within SKIPAL develops a model of the expertise levels of every user on every topic. iLink is structured to treat posts as either questions or answers, but the questions don't have to be in proper question form. These could be statements and responses to statements just as easily. iLink is not recognizing the English structure of the sentences, but learning to build statistical models based on the words used.

In SKIPAL, a reader of an event may decide to post a question. After the user types the question, iLink determines the people with the highest expertise level for the topic and provides them as candidates for receiving the question. The user then chooses which people the question should be sent to. Recipients of the question, can

- Answer the question
- Fail to answer the question
- Forward the question on to somebody they believe can answer the question.

When the question is answered, the person posting the question can rate the response. The expertise model is updated to reflect the new information on who can answer questions on this topic or at least knows who to go to.

Utilizing the Q&A capability of iLink might be done in the following way:

1. A requirements engineer posts a question or states an issue relevant to the requirements being explored.
2. If we are early in the process and the model is ill-defined, the engineer can choose from among those who would normally be selected using a priori knowledge. In parallel, the same question could be posted to the blogs to allow all users to notice the question.
3. If the answers come back satisfactorily, those people will have their expertise ratings increased. The users may pull others in by forwarding the question and the model benefits from increased information.
4. Meanwhile the topic and expertise models are being built up by the posts and reading habits of stakeholders. When we next ask a question about the topic some new stakeholders may be suggested to the engineer.
5. When answers are returned, the engineers should judge them for the quality of the information rather than whether they agree or disagree with the position. This will lead to developing a model of the social network that points towards those stakeholders who can provide quality information to future questions.
6. When the requirements engineers use the classifier to pull out and group comments on issues, those posts by people with higher expertise values for the topic might be given more weight. In fact, tools could be developed that would sort the comments by the expertise of the poster.

When we expand the number of stakeholders we want to reach, the use of an expertise model allows us to understand the real social network among the stakeholders rather than just the advertised social network from organization charts. This should lead to better requirements by surfacing information from the real experts.

3.4 Task Learning

Finally, a major capability within the PAL program, but only indirectly related to text and blogging capabilities, is *task learning*. There are several mechanisms by which PAL derived systems can learn tasks that vary in the amount of interaction required by the user. The results of the learning produce a rich task model that describes the actions that must be performed, conditions and events that influence the tasks, and probabilistic information of the likelihood of success and duration of a particular step [5].

SKIPAL will use task learning in later spirals to manage tasks associated with categories of events. SKIPAL will be taught that when an event of a particular category is found that a certain set of tasks must be completed, that those with high levels of expertise from the models described above need to be sought out to perform some of them, and that SKIPAL must use what it has been taught to complete the rest. The PAL Blog proposes to use task learning in a similar fashion. When a message (in any form) is received by the user, a set of learned tasks are initiated that result in posts being either automatically or semi-automatically being made to the blog.

While some related set of processes may be possible for the requirements engineering blogs described in the case study, the utility isn't directly evident. However, tt may be possible to look at the task learning being done in PAL as a way to manage prototyping efforts.

If we view prototyping as shown in Figure 5 below, task learning can provide a valuable contribution to rapid prototyping. If the user working with a learning system can create a task that fulfills his/her needs, the resulting task model can become a requirements specification for the step in the figure labeled "Construct production system". This obviously will only work for requirements that can be described within an information system, but it may have merit nonetheless.

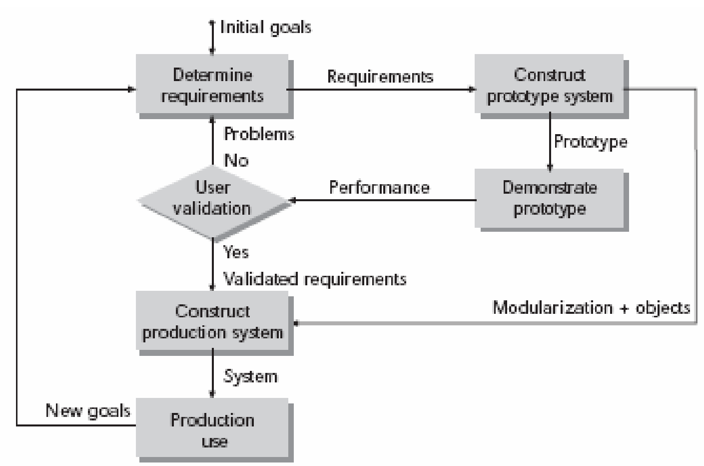

Fig. 5. Rapid Prototyping [From 6]

4 Results from Experiments with Blogs

During the summer of 2007, three experiments were done in analyzing text in the SKIPAL environment. Each of these experiments related to the problem presented in the case study [7] in ways described in Section 3.

4.1 Experiment 1 – Performance of the Classifier

In our first experiment, we trained a *Naïve Bayes* classifier to recognize events of different types using archival SKIWeb data. Table 1 below provides the results from the experiment.

Table 1. Results from Classifier Experiment

Category	# Events in Category	# Event Used for Training	# Events Used for Test	True Positive (TP)	False Positive (FP)	False Negative (FN)
Category #1	17	11	6	4	0	2
Category #2	84	56	28	28	0	0
Category #3	123	82	41	39	0	2
Category #4	39	25	13	10	1	3
Other	1387	909	471	470	7	1

In what is a fairly standard protocol, two-thirds of the events that belonged to the categories were labeled with category names. One third was left unlabeled and the classifier was asked to label them.

While the data in SKIWeb corresponded to discussions about events in the real world and in military exercises for which decisions needed to be made, they represent well the situation that could occur in requirements analysis. Using the case study [7], categories might have been labeled *breaks, screening tests, liquids,* or other representative topic labels, and the classifier could have sorted them out into different discussion threads. Likewise, the *other* category contains everything that wasn't judged to be in one of the labeled categories. This represents a source of new topic categories as well as place for less important messages to end up.

From Table 1, we see very small error rates even relative to the military knowledge management mission. These small numbers of errors would not be difficult for an engineering activity to work with. Prior to utilizing a naïve Bayes classifier for requirements engineering as discussed in Section 3, it would be worth investigating how small the training set could get before the error rate was unacceptable. It is possible that with only 10-20 blog posts labeled, acceptable classification could occur.

4.2 Experiment 2 – Relevance

In the second experiment, we trained the topic modeling engine from PAL and four other algorithms on one year of reading, writing, and book-marking habits of 6 users of the SKIWeb system. Five different recommendation engines, developed from the five algorithms, recommend thirty events from a new set of events the user should read. We then asked the users to judge whether the events recommended were relevant to them or not. The results, shown in Table 2 below, demonstrated that all of the techniques used for topic modeling were successful with some users, but other users were more difficult to characterize.

Review of these users showed that as expected that the topic modeling was able to recommend for active bloggers more easily than those who merely read. The hypothesis (which still needs to be checked) is that active participants focus their authorship in areas that are of more direct interest to them, but readers will read a wide variety of information. Most classifier based methods beat the topic modeling in this batch mode experiment, which put the topic modeling at a disadvantage due to its need for greater signal (expertise model from the questions and answers along with explicit feedback).

In future experiments we will hope to measure the difference in performance provided by the preferred environment for topic modeling which is the same environment for the requirements engineering blogs, and active discussion with dynamic social network. iLink is already being used in dynamic environments with user proclaimed success [9], but without metrics available.

The "Precision at n" measure is a standard used in information retrieval but can be problematic in recommender applications [8]. We provided a list of 30 recommendations and therefore measured "Precision at 30". SKIPAL actually produces a list of all events ordered in relevance order, but for the purposes of this experiment we cut the recommendations off at 30 so that users would only have to review 150 events each. It is possible that there were fewer than 30 relevant events to provide, in which case the algorithms were being penalized by the user interface. It is for this reason that it is only useful as a comparison among the algorithms and not as an absolute indication of value.

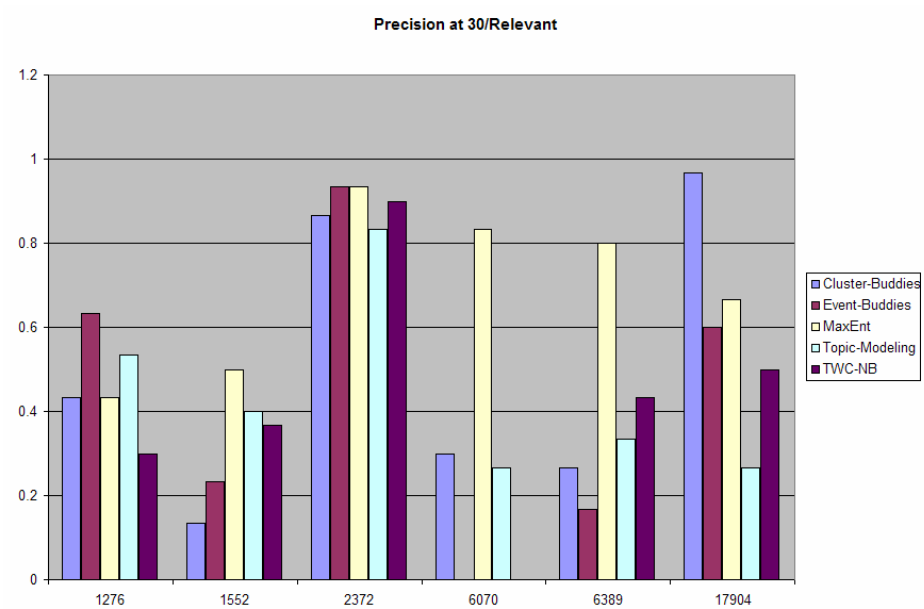

Fig. 6. Precision at 30 for Six Users of SKIWeb

Precision is defined as the number of relevant items in the list of 30, divided by the number of items recommended (i.e., 30). Recall (another information retrieval standard metric) was not possible to compute without asking the users to review all possible events from every day of the test, which was not feasible.

The precision measure indicates how well the recommender is doing in providing relevant recommendations within the list of 30 events relative to other methods providing the same number.

4.3 Experiment 3 – Relevance Based on 'Read-By' Data

In a third experiment, one year of SKIWeb data was again used to train five recommendation engine algorithms. These included:

- Topic modeling from iLink
- Naïve Bayes (labeld as TWCNB for transformed weight-normalized compliment naïve Bayes).
- Maximum Entropy Modeling (another classification approach)
- Event Buddies (using TWCNB to correlate events and users, then recommend events that most closely related users read)
- Cluster Buddies (events are clustered into word groups and users are associated with word groups based on events read, again using TWCNB).

This time, 50 users were selected whose reading habits were closest to the median number of events read, but without exceeding the median number. This time, two weeks of data was introduced one day at a time, and the recommendation engines were asked to provide a recommendation for the users' daily read. We then compared this to what the users had actually read assuming that it would be an indication of relevance. Again, signals for training included events read, authored, blogged comments, and bookmarks. The purpose was to help select an algorithm for use, so the metrics used again were mostly comparative. However, the following graphs show that all of the algorithms were successful at predicting what a user would read, and some significantly so. The metric used was a ratio of the area under a ROC curve [8] to the area under a ROC curve that represents a perfect ordering of the recommendations, where all relevant (read) events are above all irrelevant (unread) events. The results for each algorithm are represented by Gaussian distributions and graphed in Figure 7 below.

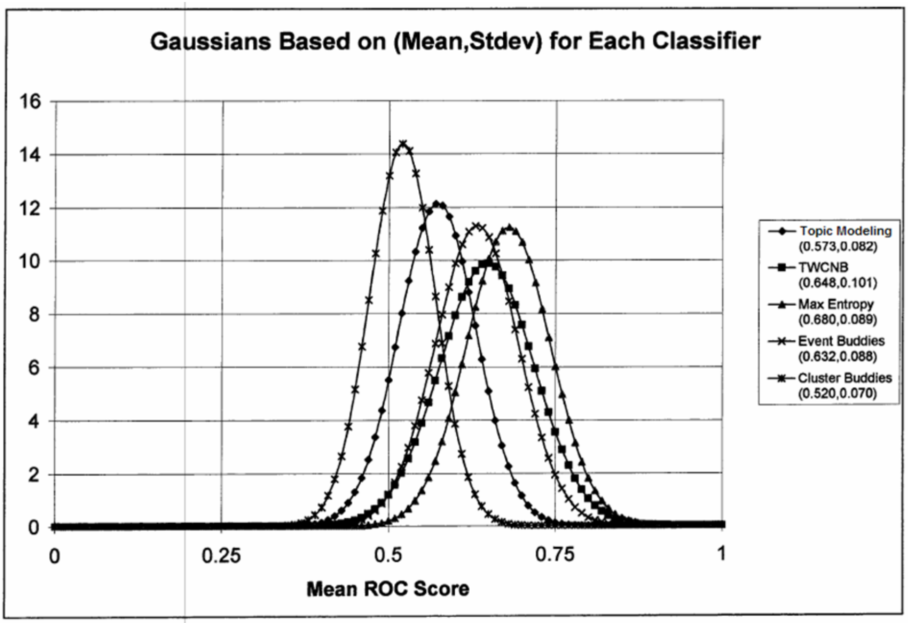

Fig. 7. Experiment Three Results

Again, the limitation of this experiment was its "batch" nature favoring classifiers which are more easily designed for such use. The topic modeling was therefore at a disadvantage as it requires interactive use to get all required signals.

4.4 Conclusions from Experimental Results as Related to Case Study

Classifiers and recommendation systems may well have a contribution to make if blogs are used to facilitate the discussion of system requirements. Classifiers can help find the entries that are relevant to particular requirements or issues. Likewise, recommendation systems that use topic modeling can identify the key stakeholders for which a requirement is important and allow the analysis to be focused on the needs of these stakeholders. The stakeholders are served through allowing them to focus on blog discussions relevant to their interests.

5 Conclusions

The problem posed by the case study is that we often would like to elicit requirements from a large and diverse population of stakeholders. Our current methods and tools require us to focus on a small number of stakeholder representatives. The case study by suggesting the use of blogs in defining requirements for a large enterprise like airport safety hints at something beyond 13 posts by 3 users.

SKIWeb and other military knowledge management capabilities serve exactly this same sort of population. Large commands themselves consist of hundreds or thousands of people, working in a variety of specialties. Distributed collaboration tools network many of these large commands together along with many smaller ones. SKIWeb was created out of a recognition that in a networked world, information and knowledge management was not a hierarchical process as was formerly the case in the military. Execution still has hierarchical components, but information doesn't need to be bound to such a structure.

Requirements elicitation and engineering in a large distributed enterprise doesn't differ very much from the primary activity of large military commands, deciding what actions need to be taken. Many staff members are acting as analysts. They collect information, develop a model of the environment, and suggest what actions are required.

Therefore, there are two aspects of these organizations that are related to the posed problem. First, we need capabilities that improve the ability of large numbers of people to communicate and share knowledge. Second, we need tools that allow analysts to ask questions or propose hypotheses, and get responses back in a way that doesn't overwhelm them.

The case study starts with an issue being raised and responses being proffered. SKIPAL will use iLink to perform this task. In using the statistical models of expertise and topic interests, users who raise issues have the ability to ensure that those with the greatest likelihood of responding authoritatively will see the questions. SKIPAL will evolve a model of who the experts are on any topic raised. Similarly, in a large enterprise, engineers need the ability to understand the real social network and find the true authorities. In a large diverse enterprise, limiting elicitation to a small number of proffered experts may yield poor results.

When we include the entire enterprise in on all topics, users need tools to allow them to filter what they read and focus on what matters to them. This is vital in the command and control environment of U.S. Strategic Command and the capability to focus on relevant information will help staff members avoid being overwhelmed with information. Similarly, if we want feedback from experts all over TSA, the FAA, and individual airports, then we must give them the tools to focus in on the issues that are important to them and that they feel they have something to contribute to. The topic models within SKIPAL perform this task, and the initial experiments were discussed in Section 4. User feedback on the performance of these tools now fielded are allowing the algorithms to be tuned and improved.

Text classifiers are useful in the SKIPAL environment to pick out events of particular interest. In the requirements engineering domain, classifiers can group posts by issue and allow requirements engineers to see all the arguments for an issue grouped together. By seeing what falls out of the classifiers into an "other" bin, new issues can be identified.

The machine learning tools applicable to large distributed command and control enterprises appear to offer many benefits to requirements engineering if we wish to elicit information from a large diverse population of stakeholders using network communication tools like blogs.

References

1. Seymour, G., Cowen, M.: A Review of Team Collaboration Tools Used in the Military and Government, `http://www.onr.navy.mil/sci_tech/34/341/docs/cki_review_team_collaboration.doc`
2. Boland, R.: Network Centricity Requires More than Circuits and Wires. Signal 61(1), 83–100 (2006)
3. Lange, D.: Boot Camp for Cognitive Systems: A Model for Preparing Systems with Machine Learning For Deployment. Ph.D. Dissertation, Naval Postgraduate School, Monterey, CA (2007)
4. Conley, K., Carpenter, J.: Towel: Towards an Intelligent To-Do List. Technical Report, SRI International, Menlo Park, CA (2006)
5. Myers, K.: Building an Intelligent Personal Assistant. AAAI, Menlo Park (2006) (Invited Talk)
6. Goguen, P., Luqi: Formal Methods: Promises and Problems. IEEE Software 14(1), 73–85 (1997)
7. Luqi, Kordon, F.: Advances in Requirements Engineering: Bridging the Gap between Stakeholders' Needs and Formal Designs. In: Paech, B., Martell, C. (eds.) Monterey Workshop 2007. LNCS, vol. 5320, pp. 15–24. Springer, Heidelberg (2008)
8. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating Collaborative Filtering Recommender Systems. Trans. Information Systems 22(1), 5–53 (2004)
9. Davitz, J., Yu, J., Basu, S., Gutelius, D., Harris, A.: iLink: Search and Routing in Social Networks. In: 13th International Conference on Knowledge Discovery and Data Mining, pp. 931–940. ACM, New York (2007)
10. Segaran, T.: Programming Collective Intelligence. O'Reilly, Sebastopol (2007)