Summarizing Non-textual Events with a 'Briefing' Focus

Mohit Kumar   **Dipanjan Das**   **Alexander I. Rudnicky**

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA-15213, USA
{mohitkum, dipanjan, air}@cs.cmu.edu

Abstract

We describe a learning-based system for generating reports based on a mix of text and event data. The system incorporates several stages of processing, including aggregation, template-filling and importance ranking. Aggregators and templates were based on a corpus of reports evaluated by human judges. Importance and granularity were learned from this corpus as well. We find that high-scoring reports (with a recall of 0.89) can be reliably produced using this procedure given a set of *oracle* features. The report drafting system is part of a learning cognitive assistant RADAR, and is used to describe its performance.

Introduction

In this paper we describe a briefing system that forms a part of a learning cognitive assistant. The assistant aids a human being in handling sudden crisis situations that may arise during a conflict resolution scenario. The purpose of the briefing assistant, as a part of the larger system, is to summarize a variety of events that occur during the span of conflict resolution. The primary difference of this work from previous research on summarization lies in the fact that we attempt to create briefings from real world 'events' rather than solely from natural language text. Most often, summarization techniques from single as well as multiple documents use an extractive paradigm, where no natural language generation methods are used. We arrive at a compromise between purely extractive and abstractive techniques by designing a set of templates that have wide coverage on the types of activities during system operation then customizing these to reflect actual event patterns. This set of templates is determined by performing corpus study of human generated briefings after test sessions, noting the range of briefing items that human beings use in summaries.

We create candidates for a briefing by filling up dynamic fields in templates that take up different values over different sessions. The filling up process relies on a set of aggregators that accumulate user activity on different tasks that the assistant supports. User activity is logged by components of the larger system for consumption of component modules such as the briefing assistant. During template design, we noticed that in our corpus, certain classes of templates which convey the same information have different granularities in terms of quality. The granularity difference creates templates summarizing the same information but in different ways. A ranking model is designed for ranking the tentative set of templates that helps extract the most relevant ones given a session. It is modeled as a consensus-based classifier, where individual classifier models are built for each user in the training set. The prediction scores of each classifier

are combined to produce a final rank of each template. The briefing system's recommendations are the four top-ranking templates. A best performance of 89% recall with *oracle* features is achieved using the attempted methods of learning the best set of briefing items.

The plan of the paper is as follows. We describe relevant work from existing literature in the next section. Then, we provide an overview of the full cognitive assistant featuring the underlying model of the briefing system. A detailed description of its components, including the ranking model is outlined in the following section. All the experiments performed and the results achieved are described next, following which we conclude the current work with a mention of directions of future work.

Related Work

Automatic text summarization, a classical problem in Artificial Intelligence, has been explored for almost half a century (Luhn, 1958). Recent work has primarily been in the areas of newswire summary generation (DUC, 2002 2003 2004 2005 2006). However, there is a marked difference between summarization and briefing systems. (Radev and McKeown, 1998) describes the SUMMONS system and distinguishes between the two to point out the objective of the latter type, that of *briefing* the user about information he or she is interested in. (Mani et al, 2000) describes another briefing system that considers user preference. Precisely, the system asks the user to create the briefing outline and using that as input it fills up the content for the briefing. This is quite similar to our objectives, but our system does not interact with the user in creating the briefing outline. Moreover, the application uses multimedia information in the final report, which is not in our system's scope. Contrasting furthermore, we focus on identifying relevant information from data and creating a representation that suits the user and the tasks that have been completed in the larger world. (Kumar et al, 2007) describes yet another briefing system that consumes user's feedback for model tuning and feature discovery and learns a personalized model of a user writing a report. However, the system differs from the present work in the two ways. First, we don't attempt to model a person's behavior and rather use a consensus-based model to arrive at the suitable briefing items. Second, the cited system is a text-based extractive summarization system unlike ours.

Focusing on event-based summarization, (Daniel et al, 2003) talks about identification of sub-events in multiple documents to create a summary. The work focuses on dividing an article into events and accumulating perspectives of different documents on the same topic to create a useful summary. (Filatova and Hatzivassiloglou, 2004) mentions the use of event-based features in extractive summarization and investigates the results of their system compared with a baseline feature set corresponding to other state of the art summarization methods. (Li et al, 2006; Wu, 2006; Xu, 2006) describe similar work based on events occurring in text and techniques of summarizing from single and multiple documents. However, unlike the case at hand, all the work on event based summarization has text as the content on which the techniques are applied.

In spite of there being a colossal amount of work on text summarization, non-textual summarization techniques have not been investigated in abundance. We can cite instances of non-textual summarization techniques like (Buyukkokten et al, 2001), where the authors present methods of summarizing the contents of a webpage to make it presentable to users of a hand-held device, considering aspects of computer-human interaction issues. Summary creation of other

media like video has been investigated quite frequently (Maybury and Merlino, 1999) but the techniques belong too exclusively to the multimedia domain to provide enough cues for the kind of summarization we explore.

However, (Maybury, 1995) explores summarization from events (e.g. weather, financial and medical knowledge bases) rather than from reduction of free text. The events are selected by analyzing domain dependent semantic patterns, link analysis of different events and statistical analysis. (McKeown et al, 1995) describes another summarization approach that takes non-textual data as input. They describe two systems that generate summaries of basketball games and that of telephone network planning activity, and focus on linguistic summarization to convey as much information as possible through a short amount of text. Their work relies on natural language generation techniques, as opposed to our approach of minimal linguistic involvement in summarizing events. (McKeown et al, 1997) another such system that describes the use of temporally coordinated speech, text and graphics generated from healthcare databases and not from pure text. However, primary focus of the work lay in generation issues rather than mining more relevant topics for the user.

Our system describes a model that derives final briefing items from a voting scheme between learned models of users, but the approach is distinguishable from collaborative filtering techniques of creating recommendations (Melville et al, 2002; Hofmann, 2004) where usually there is a step involving a profiling process of different users who have a set of items to rate or choose from. Patterns common to several users are searched for, to calculate a prediction for a new user. In contrast, our application creates an environment where the status of different tasks may vary widely, and hence for different users, the available options may differ to large extents. Below, we describe our approach towards a summarization method that aggregates real world events to form candidate briefing templates, learns to rank candidate items and produces a final briefing for the user.

System Overview

Domain Description

The target domain of the briefing application is a cognitive assistant that uses machine learning to assist a human being to solve a set of tasks whose number might turn out to be very large and the environment might be affected by a sudden crisis. The long-term objective of this work is to develop technology that will be able to function "in the wild", without significant intervention by experts and without the need for expert knowledge on behalf of the end user. The current domain is a conference scheduling scenario, where a number of entities such as sessions, speakers, conference venue, the venue's rooms, different vendors supplying materials to the various sessions etc. co-occur and need to be scheduled by a human being.

The scenario is simulated such that a set of task requests arrive in the form of emails in the user's inbox. A finite amount of time is assigned to the user to complete a part of this set, towards a goal of scheduling the conference while taking into account various constraints and difficulties that arise over the course of a session. As crisis situations arise during the session, the system helps the user solve the problem. A supervisor of the system user (the system user's role can be visualized as a secretary to the former) asks for a report through an email, to brief him about the

activities that went on during the session. It is at this juncture that the Briefing Assistant (BA) comes into picture. The function of the BA is to aggregate information about all the completed tasks and the tasks that were ought to be completed, rank the information in an order that the module thinks to be appropriate and suggest to the user briefing bullets that he might further modify or add to and create the final report. Figure 1 gives an overview of the domain and the role of the briefing assistant with respect to the larger world.

Detailed Overview

To gain a better understanding of the BA's operations, the reader requires a detailed overview of the module's interactions with the other parts of the larger system. As indicated by Figure 1,
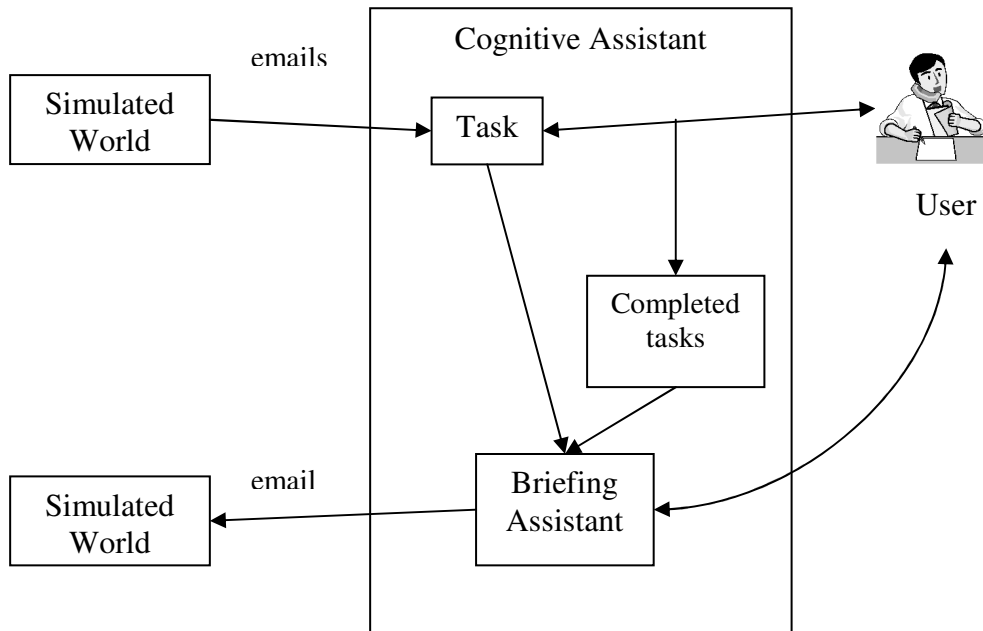

Figure 1: Role of the Briefing Assistant

incoming emails are converted to tasks automatically with minimal user participation by an email classifier (Bennett and Carbonell, 2005). Eight types of tasks can occur in the conference scheduling scenario, and each email is classified into one or more categories, thus creating a "to-do" list for the user handling the system. The eight tasks are defined as follows:

1. CHANGE-ROOM: This task corresponds to changing the attributes of a room that belongs to the conference world. Attributes can be the conference capacity of the room, the number of audio-visual equipments present in the room and so on.

2. CHANGE-SPEAKER: Similar to the previous one, this task changes the attributes of a speaker of the conference. An attribute can be the availability of the speaker, which may affect the way his sessions fit in to the final conference schedule.

3. CHANGE-SESSION: This task changes the attributes of a particular session in the conference. An example attribute of a session is the attendance of the session.

4. WEB-VIO: VIO stands for virtual information officer, and these tasks are requests asking for website updates. The conference has a website that contains information about speakers, their papers, their contact information, etc. Often, some information is missing or incorrect and WEB-VIO emails ask for the necessary changes on the website.

5. WEB-WBE: These tasks refer to batch updates on the website and provide a link to a file (say .csv) which should be used to make the update.

6. INFO-REQUEST: This task type is for generic requests for information: directions, website pointers, answers to simple questions, etc. For example, someone asks for the conference website link.

7. MISC-ACTION: This type of task is used for miscellaneous tasks that do not fit into the other task types and are largely satisfied with vendor orders. For example, if someone asks for a vegetarian meal or a slide projector for a session, it will be classified as a MISC-ACTION task.

8. BRIEFING: This is the task that requests a briefing of the ongoing activities in the conference scheduling scenario.

The BA interacts with three modules of the larger system. They are described as follows:

1. *The Space Time Planner (STP):* The space time planner is the module that manages and schedules different sessions, rooms and times in an intelligent manner (Fink et al, 2006), satisfying many constraints. The constraints can be the availability of speakers, unavailability of parts of buildings and so on. The Briefing Assistant interacts with the STP using its API to retrieve attributes of various sessions. The schedule update is also noted by the STP. It also enumerates the sessions that are moved by its scheduler. It lists the sessions that have not been scheduled in any room.

2. *The Task Manager (TM):* The TM (Garlan and Schmerl, 2006) forms the backbone of the system and provides the user with the console that contains all the tasks that need to be performed. The incoming emails provide a user interface (Faulring and Myers, 2005; Faulring and Myers, 2006) that helps accomplish the associated task. The CHANGE-ROOM, CHANGE-SESSION and CHANGE-SPEAKER tasks are such that the form provided in the console helps the user complete it. All changes are systematically logged by the Task Manager. The WEB-VIO task provides a link that activates the interface of the VIO module. The web updates are written back to the TM and it is logged similarly.

3. *Natural Language Processing (NLP):* The NLP module analyzes the responses to vendor orders placed by the user. The vendor order requests come in the form of MISC-ACTION emails. The orders can be either for food, flowers, audio-visual equipments or security. They are placed by the user using an external web-portal. The corresponding vendors send responses or confirmations that are parsed by the NLP-module. These parses form an important input for the briefing assistant to report.

A *crisis* in the conference scheduling world can have many definitions. The present world defines it in the form of a sudden unavailability of a part of a building where many sessions were initially scheduled. The unavailability spans over a period of a few days or the entire duration of the conference. The user is notified of the crisis, and with the help of the system, he reschedules the sessions with optimality.

Data Collected

Thirty-three subjects were asked to use the system under various configurations and to write Briefings. They were given broad guidelines, for example that the Briefing should have 4 bullet points corresponding to the most important activities that should be reported. The briefings were then evaluated by a panel of three judges. The judges assigned scores (0-4) to each of the bullets based on the coverage of the crisis, clarity and conciseness, accuracy and the correct level of granularity. So the data corresponding to each user consisted of the task-based event logs and the corresponding briefing. We used the judges' scores for selecting the data for template design and creating the training/test corpus for the ranking module (these are further explained in the template design and the experiment sections respectively).

The Briefing Assistant Model

We modeled the problem of Briefing generation in the current domain as non-textual event-based summarization. The events are the task creation and task completion actions logged by the various specialists. To circumvent the problem of natural language generation we designed a set of templates, based on the actual briefings written by users. More details about the template design process are given in the 'Template Design' subsection. Based on this set of templates we identified the patterns that needed to be extracted from the event logs in order to populate the templates.

The overall data flow of the BA is shown in figure 2. The various specialist modules generate task related events that are logged in a database. From this database the aggregators extract the important patterns. These patterns are used to populate the set of templates that form the candidate briefing items. The candidate briefing items are then ranked by the Ranking module and evaluated against the user's selection.
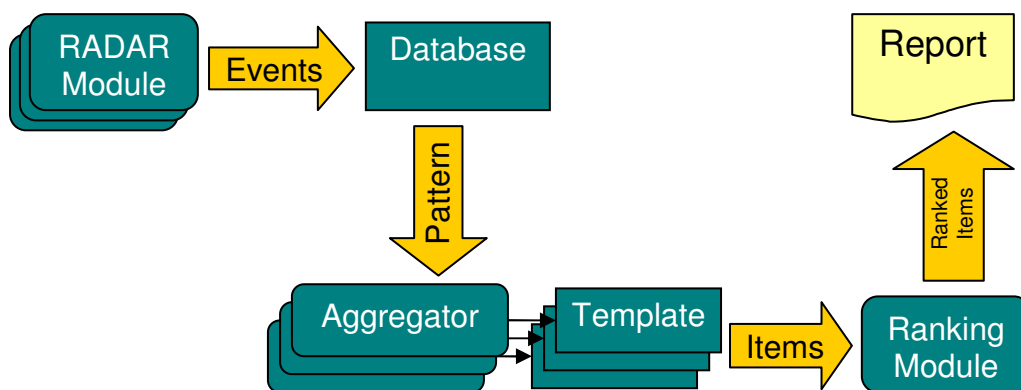

Figure 2: Briefing Assistant Data Flow

Template Design

For designing the templates, we selected 81 briefing items from 26 users that had a judge's score greater than an empirically chosen threshold (1.67). We were lenient in setting the threshold as we wanted to capture templates with wide coverage. Additionally, we added a few templates that we believed should be present in the set corresponding to data that is difficult to obtain (for example the number of sessions moved) and negative templates (the user may not realize that he hasn't completed certain tasks). So we have a set of 26 templates that correspond to eight broad categories. These eight categories do not correspond to the task types rather they are the information types that we expect to see in a briefing. Figure 3 shows the tree corresponding the eight categories.
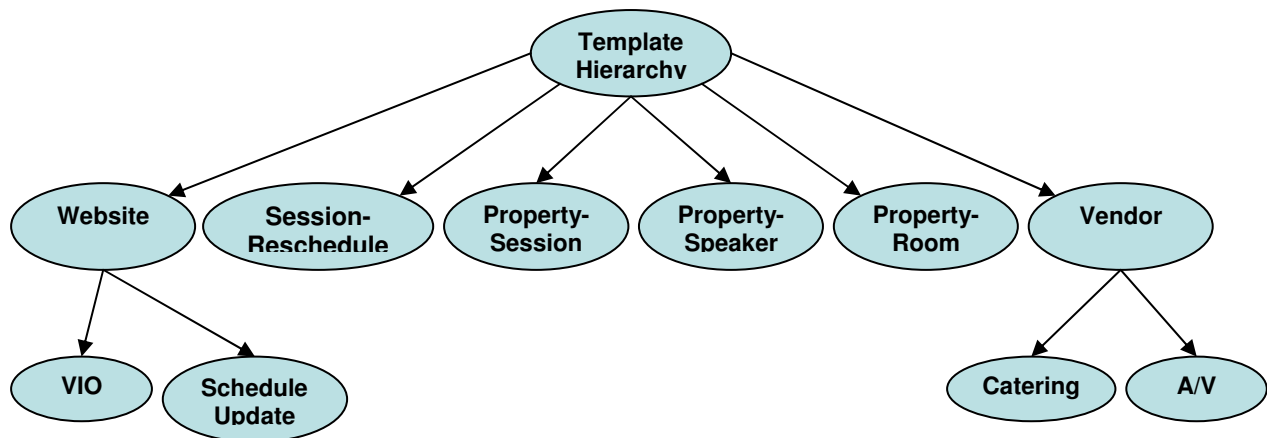

Figure 3: The category tree showing the information types that we expect in a briefing

The templates categories are defined as follows:

a. Website updates:

   VIO: As mentioned earlier, VIO stands for Virtual Information Officer and it maintains a website that contains details about speakers, their papers, organizations that they belong to and so forth. The VIO-related tasks that the user completes include (for example) changing an incorrect paper title. This template corresponds to such updates and summarizes the number of such updates done during a session.

   Schedule Update: During the scheduling process of the conference, the user moves sessions around with the help of the system. They might decide to publish the schedule on the website after significant changes. This template attempts to capture this website publish.

b. Session-Reschedule: This template captures the number of sessions that have been rescheduled because of the crisis scenario. The crisis corresponds to the unavailability of rooms over a span of time, and the user is forced to reschedule the sessions scheduled in these rooms.

c. Property-Session: It often happens during the system operation that an email arrives to inform the user of an increase or a decrease of attendance of important sessions. The user updates the session properties and this template covers such updates.

d. Property-Speaker: Speakers send in emails asking to change their availability during the span of the conference, as and when their personal schedules change. The user has an option of changing the availability of different speakers. This availability might affect the schedule of different sessions too. The current template tries to aggregate the changes in the speaker availability.

e. Property-Room: When a crisis hits the conference scheduling process, new rooms might become available to compensate for the loss of previously scheduled rooms. The availability of new rooms might indicate the increase of different capacities of the room, like the conference capacity, the banquet capacity etc. The user has the option to change these figures through the system console. Such updates are captured by this template.

f. Vendor Orders

Catering Vendors: As part of the miscellaneous tasks, there are requests for special meals for particular sessions from the conference attendees. The user uses different vendor quotes to confirm orders to catering vendors. This template summarizes such orders for different sessions and different types of requests.

A/V Vendors: Similar to the previous template, this template creates a summary of confirmed vendor orders that deal with audio visual equipments. During the scheduling process, many users might require special A/V equipments like laptops, slide projectors etc for their presentations. The system user makes corresponding vendor orders for these equipments, and the current template mines this information.

Component Description

Aggregators

The aggregators are the foundation for the set of templates that form the candidate items that make up a briefing. These components are a set of methods that mine the logs and databases and summarize the different user interactions that occur during a session. The aggregators are in four main categories:

1. Console-Based: These aggregators accumulate information about events that occur as a result of the user's interaction with the system console. The *Property-Session*, *Property-Speaker* and *Property-Room* templates are the ones that use these aggregators. The system user interface carefully logs all information and writes the contents of a completed task to the TM. The aggregators interact with the TM's API to mine all related tasks and summarizes required information to provide the dynamic fields to these three types of templates.

2. VIO-Based: VIO forms a component of the system that is independent of the console, and as remarked earlier, the console invokes the VIO module where the user enters updates to information on the conference website. The outcome of a VIO task is in turn written back to the TM, and again an aggregator corresponding to the *VIO* template reads its API and summarizes on all the VIO tasks.

3. STP-Based: The STP-Based aggregators depend on information that the Space-Time Planner provides. It reads the information to infer details of the sessions that have been rescheduled and whether these sessions are indeed the ones in crisis. These aggregators serve the *Session-Reschedule* category of template. The STP also indicates whether the session has been published on the website, and this indication serves as input to the *Schedule-Update* template.

4. NLP-Based: The NLP pipeline in the system provides an API for accessing the annotations that it performs on emails. The vendor-order confirmation emails are parsed by the pipeline and items like number of special meals, the cost of the meals etc are annotated. The interface is read by these aggregators to extract information about the *Vendor*-related templates.

Thus, the aggregators serve as template-populators that fill in the dynamic elements in different templates. The candidate templates are subsequently ordered by a ranking model to identify the most important items for the final briefing.

Ranking Model and classifiers

The ranking module ranks the set of candidate templates to help select the most relevant ones for inclusion in the final briefing. The ranking system is modeled as a consensus-based classifier, where individual classifier models are built for each user in the training set. The prediction scores of each classifier are combined to produce a final rank of each template. BA's recommendations are the four top-ranking templates.

Four different learning schemes were used for the individual models. The schemes were Naïve Bayes, Voted Perceptron, Support Vector Machines and Ranking Perceptron (Collins, 2002). We used the Minorthird package (Minorthird, 2006) for developing the system.

Features

The features used in the system can be classified as static or dynamic. The static features are the properties of the templates irrespective of the user's activity whereas the dynamic features are based on the actual events that took place. The static features are:-

1: *template id* – Feature corresponding to the unique template number. Instead of treating this feature as an 'enumeration' we converted it to a set of 26 boolean features corresponding to each of the templates.

2: *template class* –Feature corresponding to the information category type. Similar to template id, we converted this feature from an enumeration of 6 broad categories corresponding to the Figure 3 to a set of 6 boolean features.

3: *negativity* – Boolean feature indicating whether the template indicates that tasks were incomplete.

4: *abstraction* – Boolean feature indicating if the template contains abstraction over some underlying entities for example usage of 'personal information' instead of 'name, address, phone number etc'.

5: *granularity* – Feature corresponding to the granularity of the template. For example if it's a detailed template or a succinct one. We designed templates with 5 levels of granularity and hence we converted this feature to a set of 5 boolean features.

The dynamic features are:-

1: qualitative feature – Feature corresponding to the qualitative value of the template; it is valid only for certain templates. The feature depends on the user's activity for example if the user completed 4 out of the proposed 10 WEB-VIO tasks then the qualitative feature would be 'some'. We considered four levels of qualitative values (*few*, *some*, *most* and *all*) and thus have a set of 4 boolean features.

2: global task coverage – Numeric feature corresponding to the percentage of tasks that are 'covered' by the template. For example, if there were 15 WEB-VIO tasks that were done and the total number of tasks done was 75, then the value of this feature for the related templates would be $15/75 = 0.2$. This feature captures the intuition about the relative amount of work that the user did associated with this template, the higher the amount of work done, the more likely for that template to be of greater importance.

Feature Selection: We used Information Gain (IG) metric for feature selection. We experimented with different cut-off values for the total number of selected features. The experiments are detailed in the next section.

Experiments & Results

Data Preparation

We manually labeled each briefing bullet from the thirty-three subjects with the corresponding template numbers. The labeling was done by three annotators to a) validate the labeling process where we expected almost perfect inter-annotator agreement; b) to resolve the contentious cases by redesigning the templates. In the event, some of the briefing bullets were also labeled as 'other' as we did not have a template corresponding to them. This data will also serve as training data for planned work on learning the templates from the free-form text.

For the training/test corpus, we selected the user based on the following criteria: a) at least three labeled briefing bullets with each bullet having a score greater than 2; b) overall briefing score greater than 0.6 (the overall briefing score ranges from 0-1). The motivation for these criteria was that we wanted to use a set of 'good' users to learn the summarization behavior and the thresholds were chosen empirically after observing the score distribution. The above selection process yielded eleven users in our pool of training/test corpus.

Evaluation

The base performance metric is Recall, defined in terms of the briefing templates recommended by the system compared to the templates ultimately selected by the user. We justify this by noting that Recall can be directly linked to the expected time savings for the eventual users of a prospective summarization system based on the ideas developed in this study. In each of the test iterations, we leave one user out as the test user and consider the remaining ten users as training users. We obtain the *oracle* features corresponding to this test user using IG scores. We then train

ten individual classifiers corresponding to each of the training users considering only the *oracle* features from the test user and combine their prediction scores for obtaining the final ranking of the templates. The top four templates are considered as the system's recommendations.

We compare our experimental results with two baseline systems. The first baseline is a random baseline that is calculated by randomly selecting templates over a large number (10000) of runs of the system and determining the mean performance value. The second baseline is a 'frequentist' baseline where we propose the most frequently selected templates across training users as system's recommendation.

Experiment 1

We created four sets of training/test corpora based on different numbers of *oracle* selected features. The four corpora have *All*, *Top 10*, *Top 5* and *Top 4* features respectively. This is the notation used in the figures below to represent the feature set. We experimented with smaller cut-off values (less than 4) but this led to performance degradation. We test these corpora using the four chosen learning schemes. Figure 4 shows the result of the experiment. Each of the data points in the figure represents the recall value averaged across the eleven users. Naïve Bayes and Support Vector based model perform best with a recall of 0.89 for the system with *Top 4* features. The trend shows a significant result, in that consensus model built on top of classifiers based on the most informative features have the best generalization capability.
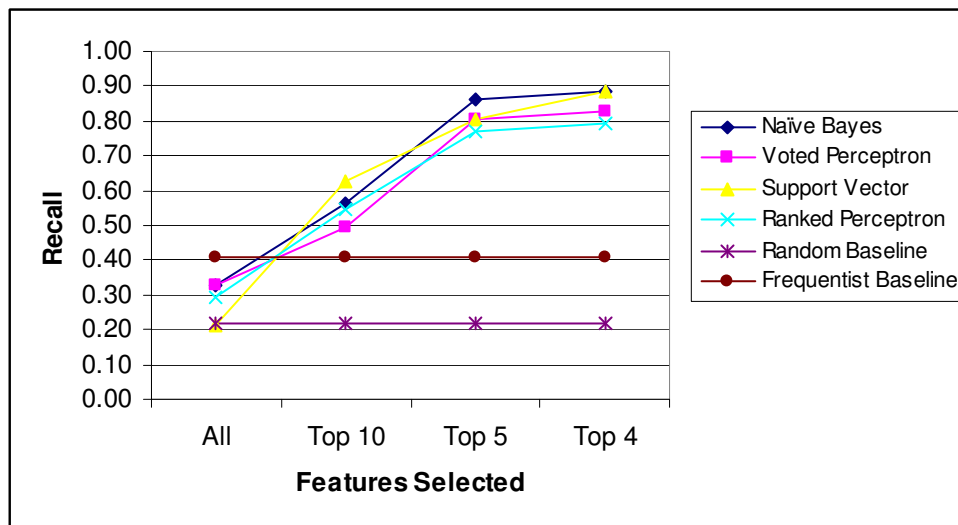

Figure 4: Recall values averaged across eleven users for Experiment 1.

Experiment 2

A more practical system is obtained when we apply Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998). In our context, MMR implies that we do not have more than one template of the same category in system's final recommendation. Figure 5 shows the repetition of experiment 1 after applying MMR. Although the overall accuracy of the best performing models (Naïve Bayes and Support Vector) for *Top 4* features drops relatively by 6.7%, we argue that the MMR-based system is more practical as the user is unlikely to write two briefing bullets on the same topic. Hence, the Recall value of 0.83, although lower than the result achieved without MMR is appropriate.

Experiment 3

To evaluate the overall goodness of the ranking model (as compared to the recall based on only the top four ranked templates), we use the Mean Average Precision (MAP) metric (Yates and Neto, 1999). MAP is defined as: Over a set of queries, find the mean of the average precisions, where Average Precision is the average of the precision after each relevant document is retrieved. Higher MAP value indicates that the user selected templates are ranked high. Figure 6 shows the MAP evaluation for the experimental settings from Experiment 1. The highest MAP score of 0.91 is obtained for Naïve Bayes model for *Top 4* features.
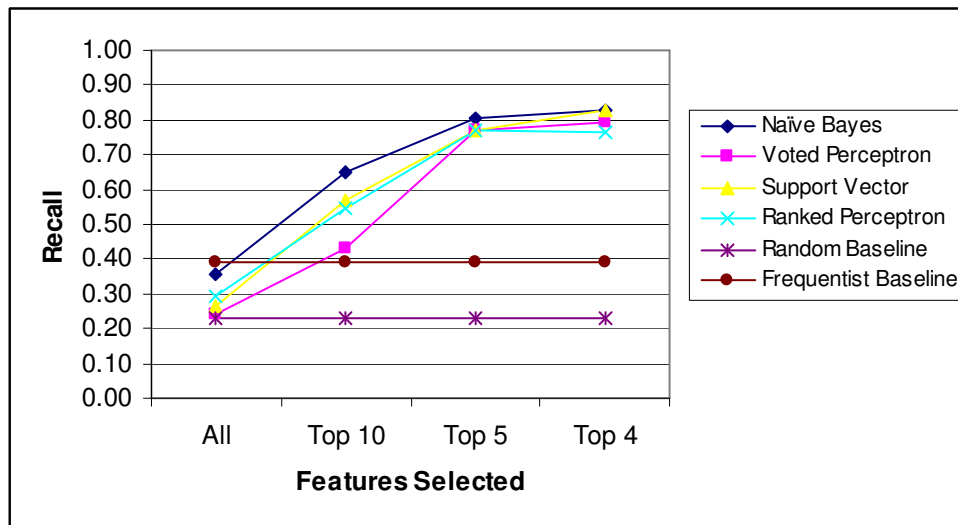
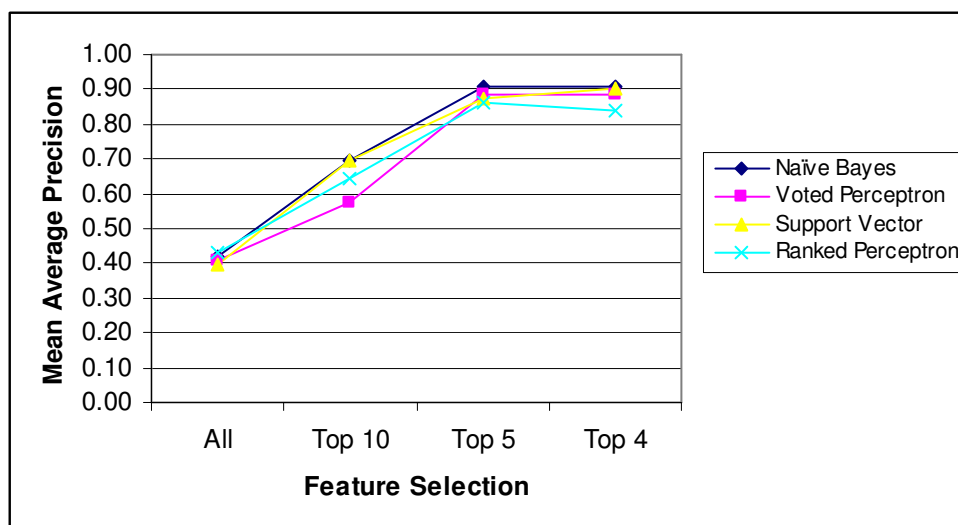Figure 5: Recall values averaged across eleven users after applying MMR.

Figure 6: MAP values averaged across eleven users.

Experiment 4

For a real system, we would not have access to the *oracle* features and we need to dynamically select the best features for each training user model. With these dynamically selected *Top 4* features we got a recall value of 0.41 corresponding to the Experiment 1, which is half as good as the original accuracy. We attribute the performance degradation to incomplete data and appending artificial data because of the absence of real information. We are awaiting user trials in a framework that will provide complete background information required by our summarization module.

Feature analysis

Overall the system based on the *Top 4* features performed the best across users. Table 1 shows the five most frequently selected features across users for this system. We observe that the top features are static. This observation along with high recall values for the experiments indicates that most of the users in our dataset behaved similarly and the consensus model was able to generalize significantly. Also it indicates that collaborative filtering techniques may perform well in this domain since the user-specific activities do not affect the feature based representation and the input space of features is uniform across users.

Feature	Frequency (across eleven users)
granularity_intermediate	5
granularity_detailed	4
template_4a	3
template_1a	3
template_3	3

Table 1: Most frequently selected features across eleven users in Top 4 system

Conclusion

The experiments show that consensus based model generalizes well across users and gives significant recall. Naïve Bayes and Support Vector based system performed best with Top 4 *oracle* selected features. We should note that our approach is not meant to discover general attributes of good reports; rather it provides a framework within which to rapidly learn the characteristics of good reports within a particular domain. As such, these custom attributes are more likely to lead to quality reports. In conjunction with this, we also note that a consensus-based approach yields the best performance, implying that it should be possible to apply data-driven techniques to learn consistent models in this domain.

Future Work

We have described a system that creates a briefing from events occurring in a learning cognitive assistant. However, the summarization process focuses on ranking different candidate items and selecting the most important ones from the set. The challenge of natural language generation in the summary creation process is obviated by adopting a template based approach rather than

generating sentences from event data. An interesting direction of future work is the elimination of fixed templates that get populated, and the induction of templates from conspicuous patterns in event data. The process of summary creation that we describe in this paper can be reverted, if instead of aggregating data to feed the fixed set of templates, we search for data patterns and create new templates summarizing them.

Learning templates from natural language text that constitute human generated briefings without any human intervention is another attractive avenue of future research. Instead of having a fixed set of templates, a variety of templates can be learnt from example briefings automatically. A generic set of aggregators can be designed to capture all kinds of information that can arise in the subject scenario, in turn catering to the learnt templates. Thus, summary creation can be made more comprehensive by extending the paradigm that we have described in this paper.

References

Bennett, P. N., Carbonell, J. G. (2005) Detecting Action-Items in E-mail. In *Proceedings of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Salvador, Brazil, Sep, 2005).

Buyukkokten O., Garcia-Molina H. and Paepcke A. (2001) Accordion summarization for end-game browsing on PDAs and cellular phones. In *Proceedings of the SIGCHI conference on Human factors in computing systems,* Seattle, Washington, USA, 2001

Carbonell, J. and Goldstein, J. (1998) The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, Melbourne, Australia, 1998.

Collins M. (2002) Ranking algorithms for named-entity extraction: Boosting and the voted Perceptron. *In Proceedings of ACL*, 2002.

Daniel N., Radev D. and Allison T. (2003) Sub-event based multi-document summarization. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop - Volume 5,* Alberta Canada, 2003.

DUC (2002, 2003, 2004, 2005, 2006) *Proceedings of the second, third, fourth, fifth and sixth document understanding conference.* 2002, 2003, 2004, 2005, 2006.

Faulring A. and Myers B.A. (2005). Enabling Rich Human-Agent Interaction for a Calendar Scheduling Agent. In *Proceedings of the Conference on Human Factors in Computing Systems Extended Abstracts (CHI 2005)*. Portland, Oregon, USA, April 2–7, 2005.

Faulring A. and Myers B.A. (2006). Availability Bars for Calendar Scheduling. In *Proceedings of the Conference on Human Factors in Computing Systems Extended Abstracts (CHI 2006)*. Montréal, Québec, Canada. April 22–27, 2006.

Filatova E. and Hatzivassiloglou V. (2004) Event-Based Extractive Summarization. In *Proceedings of ACL Workshop on Summarization*, Barcelona, Spain., 2004

Fink E., Jennings P. M., Bardak U., Oh J., Smith S. F., and Carbonell J. G. Scheduling with uncertain resources: Search for a near-optimal solution. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 137-144, 2006

Garlan D. and Schmerl B. (2006) An Architecture for Personal Cognitive Assistance. In *18th International Conference on Software Engineering and Knowledge Engineering*, San Francisco Bay, USA, 5-7 July 2006.

Herrera-Viedma E., Herrera F.and Chiclana F. (2002) A consensus model for multiperson decision making with different preference structures. *In IEEE Transactions on Systems, Man, and Cybernetics, Part A 32(3).*

Hofmann T. (2004) Latent semantic models for collaborative filtering. In *ACM Transactions on Information Systems (TOIS).* Volume 22, Issue 1

Kumar M., Garera N., Rudnicky A.I. (2007), "Learning from Report-writing Behavior of Individuals", To appear in the *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, Jan 6-12, 2007, Hyderabad, India.

Li W.J., Xu W., Wu M.L., Yuan C.F. and Lu Q. (2006) Extractive Summarization using Inter- and Intra- Event Relevance. In *Proceedings of the 21st International Conference Computational Linguistics and 44th Annual Meeting of ACL (ACL/COLING'06)*, Sydney, Australia, July 17-21 2006, pp369-376.

Luhn H. P (1958) The Automatic Creation of Literature Abstracts. In *IBM Journal of Research Development*, 2(2):159–165, 1958.

Mani I., Concepcion K. and Guilder L.V. (2000) Using summarization for automatic briefing generation. In *Workshop on Automatic Summarization, NAACL-ANLP*, Seattle, USA, 2000

Maybury M. and Merlino A. (1999) An empirical study of the optimal presentation of multimedia summaries of broadcast news. In *Advances in Automatic Text Summarization*, pages 392-401. 1999.

Maybury M.T (1995) Generating summaries from event data. In *International Journal of Information Processing and Management: Special Issue on Text Summarization*, 31(5) 1995.

McKeown K., Robin J., and Kukich K. (1995) Generating concise natural language summaries. In *Information Processing and Management*, 31(5), 1995

McKeown K. R, Pan S., Shaw J., Jordan D., and Allen B.A. (1997) Language generation for multimedia healthcare briefings. In *Proceedings of the ACL Conference on Applied Natural Language Processing*, pages 277-282, 1997.

Melville P., Mooney, R.J. and Nagarajan R. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence* (AAAI-2002), pp. 187-192, Edmonton, Canada, July 2002.

Minorthird (2006). http://minorthird.sourceforge.net

Radev D.R and McKeown K.R. (1998) Generating natural language summaries from multiple on-line sources. In *Computational Linguistics*, 1998.

Wu M.L (2006). Investigations on Event-Based Summarization. In *Proceedings of ACL Student Workshop*, Sydney, Australia, July 17-21 2006, pp37-42.

Xu W., Li, W.J., Wu M.L., Li W. and Yuan C.F. (2006) Deriving Event Relevance from the Ontology Constructed with Formal Concept Analysis. In *Proceedings of the Seventh International Conference on Intelligent Text Processing and Computational Linguistics (CiCling'06)*, Mexico City, Mexico, February 19-25, 2006.

Yates, B.R. and Neto, R.B. (1999) Modern Information Retrieval. *ACM Press*, 1999.