# Non-textual Event Summarization by Applying Machine Learning to Template-based Language Generation

**Mohit Kumar** and **Dipanjan Das** and **Sachin Agarwal** and **Alexander I. Rudnicky**
Language Technologies Institute
Carnegie Mellon University, Pittsburgh, USA
mohitkum,dipanjan,sachina,air@cs.cmu.edu

## Abstract

We describe a learning-based system that creates draft reports based on observation of people preparing such reports in a target domain (conference replanning). The reports (or briefings) are based on a mix of text and event data. The latter consist of task creation and completion actions, collected from a wide variety of sources within the target environment. The report drafting system is part of a larger learning-based cognitive assistant system that improves the quality of its assistance based on an opportunity to learn from observation. The system can learn to accurately predict the briefing assembly behavior and shows significant performance improvements relative to a non-learning system, demonstrating that it's possible to create meaningful verbal descriptions of activity from event streams.

## 1 Introduction

We describe a system for recommending items for a briefing created after a session with a crisis management system in a conference replanning domain. The briefing system is learning-based, in that it initially observes how one set of users creates such briefings then generates draft reports for another set of users. This system, the Briefing Assistant(BA), is part of a set of learning-based cognitive assistants each of which observes users and learns to assist users in performing their tasks faster and more accurately.

The difference between this work from most previous efforts, primarily based on text-extraction approaches is the emphasis on learning to summarize event patterns. This work also differs in its emphasis on learning from user behavior in the context of a task.

Report generation from non-textual sources has been previously explored in the Natural Language Generation (NLG) community in a variety of domains, based on, for example, a database of events. However, a purely generative approach is not suitable in our circumstances, as we want to summarize a variety of tasks that the user is performing and present a summary tailored to a target audience, a desirable characteristic of good briefings (Radev and McKeown, 1998). Thus we approach the problem by applying learning techniques combined with a template-based generation system to instantiate the briefing-worthy report items. The task of instantiating the briefing-worthy items is similar to the task of Content Selection (Duboue, 2004) in the Generation pipeline however our approach minimizes linguistic involvement. Our choice of a template-based generative system was motivated by recent discussions in the NLG community (van Deemter et al., 2005) about the practicality and effectiveness of this approach.

The plan of the paper is as follows. We describe relevant work from existing literature in the next section. Then, we provide brief system description followed by experiments and results. We conclude with a summary of the work.

## 2 Related Work

Event based summarization has been studied in the summarization community. (Daniel et al., 2003) described identification of sub-events in multiple documents. (Filatova and Hatzivassiloglou, 2004) mentioned the use of event-based features in extractive summarization and (Wu, 2006; Li et al., 2006) describe similar work based on events occurring in text. However, unlike the case at hand, all the work on event-based summarization used text as source material.

Non-textual summarization has also been explored in the Natural Language Generation (NLG) community within the broad task of generating reports based on database of events in specific domains such as medical (Portet et al., 2009), weather (Belz, 2007), sports (Oh and Shrobe, 2008) etc. However, in our case we want to summarize a variety of tasks that the user is performing and present a summary to an intended audience (as defined by a report request).

Recent advances in NLG research use statistical approaches at various stages of processing in the generation pipeline like content selection (Duboue and McKeown, 2003; Barzilay and Lee, 2004), probabilistic generation rules (Belz, 2007). Our proposed approach differs from these in that we apply machine learning after generation of all the templates, as a post-processing step, to rank them for inclusion in the final briefing. We could have used a general purpose template-based generation framework like TG/2 (Busemann, 2005), but since the number of templates and their corresponding aggregators is limited, we chose an approach based on string manipulation.

We found in our work that an approach based on modeling individual users and then combining the outputs of such models using a voting scheme gives the best results, although our approach is distinguishable from collaborative filtering techniques used for driving recommendation systems (Hofmann, 2004). We believe this is due to the fact that the individual sessions from which ranking models are learned, although they range over the same collection of component tasks, can lead to very different (human-generated) reports. That is, the particular history of a session will affect what is considered to be briefing-worthy.

## 3 System Overview

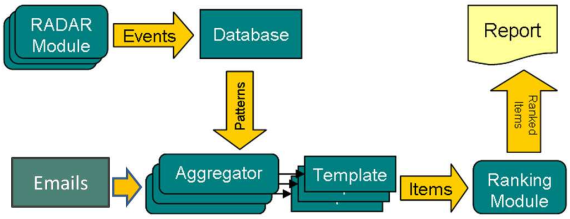


Figure 1: Briefing Assistant Data Flow.

***The Briefing Assistant Model:*** We treat the task of briefing generation in the current domain[1] as non-textual event-based summarization. The events are the task creation and task completion actions logged by various cognitive assistants in the system (so-called specialists). As part of the design phase for the template-based generation component, we identified a set of templates, based on the actual briefings written by users in a separate experiment. Ideally, we would like to adopt a corpus-based approach to automatically extract the templates in the domain, like (Kumar et al., 2008), but since the sample briefings available to us were very few, the application of such corpus-based techniques was not necessary. 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. A ranking model was also designed for ordering instantiations of this set of templates and to recommend the top 4 most relevant ones for a given session.

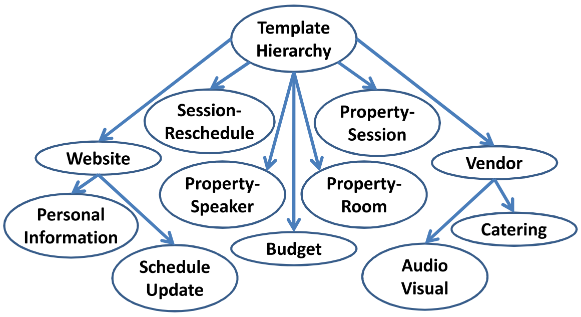


Figure 2: The category tree showing the information types that we expect in a briefing.

The overall data flow for BA during a session (runtime) is shown in Figure 1. The various specialist modules generate task related events that are logged in a database. The aggregators operate over this database and emails to extract relevant patterns. These patterns in turn are used to populate templates which constitute candidate briefing items. The candidate briefing items are then ordered by the ranking module and presented to the user.

***Template Design and Aggregators:*** The set of templates used in the current instantiation of the BA was derived from a corpus of human-generated briefings collected in a previous experiment using the same crisis management system. The set of templates was designed to cover the range of items that users in that experiment chose to include in their reports corresponding to nine categories shown in Figure 2. We found that information can be conveyed at different levels of granularity (for example, qualitatively or quantitatively). The appropriate choice of granularity for

[1] More details about the domain and the interaction of BA with the larger system are mentioned in a longer version of the paper (Kumar et al., 2009)

a particular session is a factor that the system can learn[2].

***Ranking Model, Classifiers and Features:*** The ranking module orders candidate templates so that the four most relevant ones appear in the briefing draft. The ranking system consists of a consensus-based classifier, based on individual classifier models for each user in the training set. The prediction from each classifier are combined (averaged) to produce a final rank of each template.

We used the Minorthird package (Cohen, 2004) for modeling. Specifically we allowed the system to experiment with eleven different learning schemes and select the best one based on cross-validation within the training corpus. The schemes were Naive Bayes, Voted Perceptron, Support Vector Machines, Ranking Perceptron, K Nearest Neighbor, Decision Tree, AdaBoost, Passive Aggressive learner, Maximum Entropy learner, Balanced Winnow and Boosted Ranking learner.

The features[3] used in the system are static or dynamic. Static features reflect the properties of the templates irrespective of the user's activity whereas the dynamic features are based on the actual events that took place. We used the Information Gain (IG) metric for feature selection, experimenting with seven different cut-off values $All, 20, 15, 10, 7, 5, 4$ for the total number of selected features.

## 4 Experiments and Results

***Experimental Setup:*** Two experimental conditions were used to differentiate performance based on knowledge engineering, designated MinusL and performance based on learning, designated PlusL.[4]

***Email Trigger:*** In the simulated conference replanning crisis, the briefing was triggered by an email containing explicit information requests, not known beforehand. To customize the briefing according to the request, a natural language processing module identified the categories of information requested. The details of the module are beyond the scope of the current paper as it is external to our system; it took into account the template categories we earlier identified. Figure 4 shows a sample briefing email stimulus. The mapping from the sample email in the figure to the categories is as follows: "expected attendance" - Property-Session; "how many sessions have been rescheduled", "how many still need to be rescheduled", "any problems you see as you try to reschedule" - Session-Reschedule; "status of food service (I am worried about the keynote lunch)" - Catering Vendors.

***Training:*** Eleven expert users[5] were asked to provide training by using the system then generating the end of session briefing using the BA GUI. For this training phase, no item ranking was performed by the system, i.e. all the templates were populated by the aggregators and recommendations were random. The expert user was asked to select the best possible four items and was further asked to judge the usefulness of the remaining items. The resulting training data consists of the activity log, extracted features and the user-labeled items. The trigger message for the training users did not contain any specific information request.

***Test:*** Subjects were recruited to use the crisis management system in MinusL and PlusL condition, although they were not aware of the condition of the system and they were not involved with the project. There were 54 test runs in the MinusL condition and 47 in the PlusL condition. Out of these runs, 29 subjects in MinusL and 43 subjects in PlusL wrote a briefing using the BA. We report the evaluation scores for this latter set.

***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 users. We calculate two variants of Recall: *Category-based*—calculated by matching the categories of the BA recommended templates and user selected ones ignoring the granularity and *Template-based*—calculated by matching the exact templates. The first metric indicates whether the right category of information was selected and the latter indicates whether the information was presented at the appropriate level of detail.

We also performed subjective human evaluation

[2] The details of template design process including sample templates, categories of templates and details of aggregators are presented in (Kumar et al., 2009)

[3] Detailed description of the features are mentioned in (Kumar et al., 2009)

[4] The details of the experimental setup as part of the larger cognitive assistant system are presented in (Kumar et al., 2009).

[5] Members of the project from other groups who were aware of the scenario and various system functionalities but not the ML methods

using 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. They were advised about certain briefing-specific characteristics (e.g. negative bullet items are useful and hence should be rated favorably). They were also asked to provide a global assessment of report quality, and evaluate the coverage of the requests in the briefing stimulus email message. This procedure was very similar to the one used as the basis for template selection.

***Experiment:*** The automatic evaluation metric used for the trained system configuration is the *Template-based* recall measure. To obtain the final system configuration, we automatically evaluate the system under the various combinations of parameter settings with eleven different learning schemes and seven different feature selection threshold (as mentioned in previous sections). Thus a total of 77 different configurations are tested. For each configuration, we do a eleven-fold cross-validation between the 11 training users i.e. we leave one user as the test user and consider the remaining ten users as training users. We average the performance across the 11 test cases and obtain the final score for the configuration. We choose the configuration with the highest score as the final trained system configuration. The learned system configuration in the current test includes Balanced Winnow (Littlestone, 1988) and top 7 features.

***Results:*** We noticed that four users in PlusL condition took more than 8 minutes to complete the briefing when the median time taken by the users in PlusL condition was 55 seconds, so we did not include these users in our analysis in order to maintain the homogeneity of the dataset. These four data points were identified as extreme outliers using a procedure suggested by (NIST, 2008)[6]. There were no extreme outliers in MinusL condition.

Figure $3a$ shows the Recall values for the MinusL and PlusL conditions. The learning delta i.e. the difference between the recall values of PlusL and MinusL is 33% for *Template-based* recall and 21% for *Category-based* recall. These differences are significant at the $p < 0.001$ level. The statistical significance for the *Template-based* metric, which was the metric used for selecting system parameters during the training phase, shows that learning is effective in this case. Since the email stimulus processing module extracts the briefing categories from the email the *Category-based* and *Template-based* recall is expected to be high for the baseline MinusL case. In our test, the email stimuli had 3 category requests and so the *Category-based* recall of 0.77 and *Template-based* recall of 0.67 in MinusL is not unexpected.

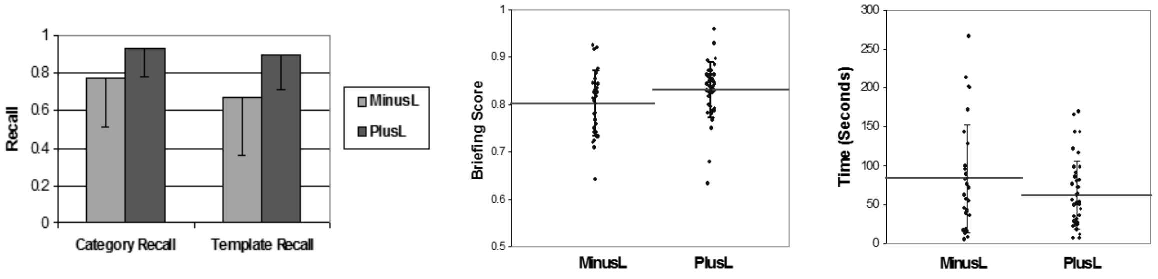
Figure $3b$ shows the Judges' panel scores for the briefings in MinusL and PlusL condition. The learning delta in this case is 3.6% which is also statistically significant, at $p < 0.05$. The statistical significance of the learning delta validates that the briefings generated during PlusL conditions are better than MinusL condition. The absolute difference in the qualitative briefing scores between the two conditions is small because MinusL users can select from all candidates, while the recommendations they receive are random. Consequently they need to spend more time in finding the right items. The average time taken for a briefing in MinusL condition is about 83 seconds and 62 seconds in PlusL (see Figure $3c$). While the time difference is high (34%) it is not statistically significant due to high variance.

Four of the top 10 most frequently selected features across users for this system are dynamic features. This indicates that the learning model is capturing the user's world state and the recommendations are related to the underlying events. We believe this validates the process we used to generate briefing reports from non-textual events.

## 5 Summary

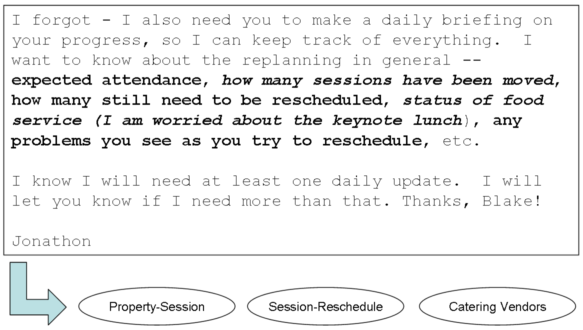
The Briefing Assistant is not designed to learn the generic attributes of good reports; rather it's meant to rapidly learn the attributes of good reports within a particular domain and to accommodate specific information needs on a report-by-report basis. We found that learned customization produces reports that are judged to be of better quality. We also found that a consensus-based modeling approach, which incorporates information from multiple users, yields the best performance. We believe that our approach can be used to create flexible summarization systems for a variety of applications.

[6] Extreme outliers are defined as data points that are outside the range $[Q1-3*IQ, Q3+3*IQ]$ in a box plot. $Q1$ is lower quartile, $Q3$ is upper quartile and $IQ$ is the difference $(Q3 - Q1)$ is the interquartile range.



(a) (b) (c)

Figure 3: (a) Recall values for MinusL and PlusL conditions (b) Briefing scores from the judges' panel for MinusL and PlusL conditions (c) Briefing time taken for MinusL and PlusL conditions.


I forgot - I also need you to make a daily briefing on your progress, so I can keep track of everything. I want to know about the replanning in general -- **expected attendance, *how many sessions have been moved*, how many still need to be rescheduled, *status of food service (I am worried about the keynote lunch)*, any problems you see as you try to reschedule,** etc.

I know I will need at least one daily update. I will let you know if I need more than that. Thanks, Blake!

Jonathon

Figure 4: Template categories corresponding to the Briefing request email.

## References

Regina Barzilay and Lillian Lee. 2004. Catching the drift: probabilistic content models, with applications to generation and summarization. In *Proceedings of NAACL*.

Anja Belz. 2007. Probabilistic generation of weather forecast texts. In *Proceedings of HLT-NAACL*.

Stephan Busemann. 2005. Ten years after: An update on TG/2 (and friends). In *Proceedings of European Natural Language Generation Workshop*.

William W. Cohen. 2004. Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data. http://minorthird.sourceforge.net, 10th Jun 2009.

Naomi Daniel, Dragomir Radev, and Timothy Allison. 2003. Sub-event based multi-document summarization. In *Proceedings of HLT-NAACL*.

Pablo A. Duboue and Kathleen R. McKeown. 2003. Statistical acquisition of content selection rules for natural language generation. In *Proceedings of EMNLP*.

Pablo A. Duboue. 2004. Indirect supervised learning of content selection logic. In *Proceedings of INLG*.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based extractive summarization. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

Thomas Hofmann. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115.

Mohit Kumar, Dipanjan Das, and Alexander I. Rudnicky. 2008. Automatic extraction of briefing templates. In *Proceedings of IJCNLP*.

Mohit Kumar, Dipanjan Das, Sachin Agarwal, and Alexander I. Rudnicky. 2009. Non-textual event summarization by applying machine learning to template-based language generation. Technical Report CMU-LTI-09-012, Language Technologies Institute, Carnegie Mellon University.

Wenjie Li, Mingli Wu, Qin Lu, Wei Xu, and Chunfa Yuan. 2006. Extractive summarization using inter- and intra- event relevance. In *Proceedings of ACL*.

Nick Littlestone. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318.

NIST. 2008. NIST/SEMATECH e-handbook of statistical methods. http://www.itl.nist.gov/div898/handbook/, 10th Jun 2009.

Alice Oh and Howard Shrobe. 2008. Generating baseball summaries from multiple perspectives by reordering content. In *Proceedings of INLG*.

François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816.

Dragomir R. Radev and Kathleen R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):470–500.

Kees van Deemter, Emiel Krahmer, and Mariet Theune. 2005. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24.

Mingli Wu. 2006. Investigations on event-based summarization. In *Proceedings of ACL*.