Learning from the Report-writing Behavior of Individuals

Mohit Kumar
Carnegie Mellon University
Pittsburgh, USA
mohitkum@cs.cmu.edu

Nikesh Garera*
Johns Hopkins University
Baltimore, USA
ngarera@cs.jhu.edu

Alexander I. Rudnicky
Carnegie Mellon University
Pittsburgh, USA
air@cs.cmu.edu

Abstract

We describe a briefing system that learns to predict the contents of reports generated by users who create periodic (weekly) reports as part of their normal activity. The system observes content-selection choices that users make and builds a predictive model that could, for example, be used to generate an initial draft report. Using a feature of the interface the system also collects information about potential user-specific features. The system was evaluated under realistic conditions, by collecting data in a project-based university course where student group leaders were tasked with preparing weekly reports for the benefit of the instructors, using the material from individual student reports.

This paper addresses the question of whether data derived from the implicit supervision provided by end-users is robust enough to support not only model parameter tuning but also a form of feature discovery. Results indicate that this is the case: system performance improves based on the feedback from user activity. We find that individual learned models (and features) are user-specific, although not completely idiosyncratic. This may suggest that approaches which seek to optimize models globally (say over a large corpus of data) may not in fact produce results acceptable to all individuals.

1 Introduction

In this paper we describe a personalized learning-based approach to summarization that minimizes the need for learning-expert time and eliminates the need for expert-generated evaluation materials such as a "gold standard" summary, since each user provides their own standard. While the work we describe is specific to summarization, we believe that the basic techniques are extensible to other learning situations that share the characteristic of repeated execution and the availability of unambiguous user feedback.

Of course this comes at a cost, which is the end-user time needed to teach the system how to produce satisfactory summaries. We would however argue that end-user involvement is more likely to generate quality products that reflect both functional needs and user preferences and is indeed worth the effort. The current paper describes the application of this approach in the context of an engineering project class in which students were expected to produce weekly summaries of their work. For the course we worked with, a reporting requirement was already in place. We minimally modified the process (by augmenting and instrumenting the web-based reporting software already in use) to collect relevant data from users; apart from this their normal activities were not perturbed. The data collected was used to perform off-line learning experiments.

While each student produces their own logs, their team leader was additionally tasked with reading the logs and selecting log entries (referred as 'items' hereon) suitable for a summary of group activity. Learning is based on information about which items are selected by the team leader. We further collect data identifying the "features" deemed important by the leader creating the summary. This is done by asking the team leader to highlight key words/phrases in the selected summary items (the phrases that they believe led them to select that item). This might be thought of as having the users directly select Summary Content Units [Nenkova and Passonneau, 2004] or identify Rouge-like units (n-grams, word sequences) [Lin, 2004]. Although in this paper we focus on extractive summarization, we believe that this approach can be extended to prime information for abstractive summarization, one of our long-term goals for this work.

The plan of this paper is as follows. We describe relevant ideas from the literature. We then describe the domain and the report generation tasks performed by the class and give details of the data collected. The Learning System section outlines key details (feature representation, weighting, ranking, model selection etc.). After describing the basic Learning framework, we describe the Experiments and Results from the different settings of our system followed by the Conclusions.

2 Related Work

Although automatic text summarization has been explored for almost a half-century [Luhn, 1958], most current work focuses on generic newswire summaries [DUC, 2002 2003 2004 2005 2006]. [Radev and McKeown, 1998] provide

*This work was done in part when the author was a student at CMU.

a useful distinction between briefings and the general concept of summaries where they focus on generating multi-document newswire briefings. [Mani *et al.*, 2000] also focus on generating briefings, however their approach contrasts with ours and is in a different domain. They assume that users are given an outline of the briefing and then try to populate the outline, whereas our system does not provide an initial structure (except for a generic template). On the other hand [Mani *et al.*, 2000] extend their work to multimedia inputs. From our current perspective we believe that the system should focus on identifying important information, allowing the user to concentrate on its organization into a coherent presentation.

A few personalized interactive learning systems have also been proposed for summarization. [Amini, 2000] describes a query-relevant text summary system based on interactive learning. Learning is in the form of query expansion and sentence scoring by classification. [Leuski *et al.*, 2003] have explored interactive multi-document summarization, where the interaction with the user was in terms of giving the user control over summary parameters, support rapid browsing of document set and alternative forms of organizing and displaying summaries. Their approach of 'content selection' to identify key concepts in unigrams, bigrams and trigrams based on the likelihood ratio [Dunning, 1993] is different from our statistical analysis and is of some interest. [Zhang *et al.*, 2003] have proposed a personalized summarization system based on the user's annotation. They have presented a good case of the usefulness of user's annotations in getting personalized summaries. However their system differs from the current one in several respects. Their scenario is a single document newswire summary and is very different from a briefing. Also, their system is purely statistical and does not include the concept of a human-in-the-loop that improves performance.

[Elhadad *et al.*, 2005] have applied personalized summarization in the medical domain where, given a patient profile and a set of documents from a search query, the system generates a personalized summary relevant to the patient. The active learning community [Raghavan *et al.*, 2006] has also been moving towards using user feedback for identifying important features. Some other interesting domains in which summarization systems have been developed are technical chat [Zhou and Hovy, 2005], newsgroup conversations [Newman and Blitzer, 2002], email threads [Rambow *et al.*, 2004], spoken dialogues [Zechner, 2001] as well as others.

[Garera and Rudnicky, 2005] describe a summarization system for a recurring weekly report-writing taking place in a research project. They found that knowledge-engineered features lead to the best performance, although this performance is close to that based on n-gram features. Given this, it would be desirable to have a procedure that leverages human knowledge to identify high-performance features but does not require the participation of experts in the process. We describe an approach to this problem below.

3 Target Domain

We identified a domain that while similar in its reporting structure to the one studied by [Garera and Rudnicky, 2005] differed in some significant respects. Specifically, the report-writers were not experienced researchers who were asked to generate weekly reports but were students taking a course that already had a requirement for weekly report generation. The course was project-based and taught in a university engineering school. The students, who were divided into groups working on different projects were required to produce a weekly report of their activities, to be submitted to the instructors. Each group had well-defined roles, including that of a leader. Students in the class logged the time spent on the different activities related to the course. Each time-log entry included the following fields: date, category of activity, time spent and details of the activity. The category was selected from a predefined set that included Coding, Group Meeting, Research and others (all previously set by the instructors).

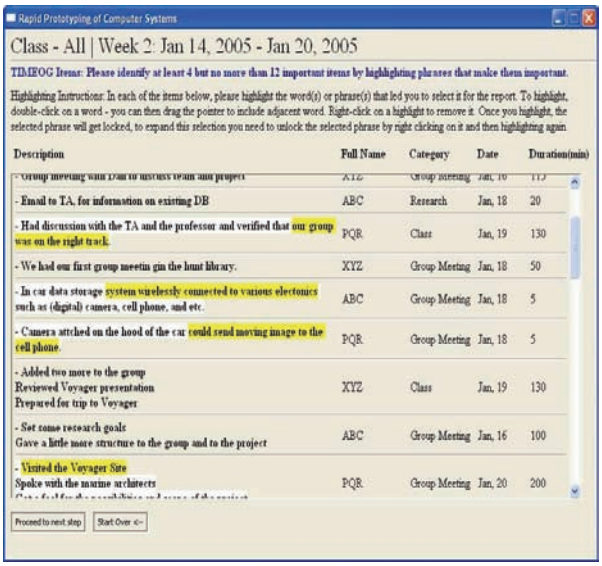


Figure 1: System Demonstration:Acquiring user based features via phrase highlighting

An example time-log entry is:"Visited the Voyager site. *Spoke with the marine architects. Got a feel for the possibilities and scope of the project.*[1]" The data in this domain is not as structured and grammatical as newswire documents. It is also different from chat-like/conversational corpora in its usage of words although it does contain a few spelling mistakes. One important characteristic of the data is that the content of the time-logs changes over time as the project progresses. So the word-based features that may indicate importance in the earlier weeks may not be useful in the later weeks.

The task of the team leader is to prepare a weekly report for the benefit of the instructor, using the time-log entries of the individual team members as raw material. As the students were already using an on-line system to create their logs, it was relatively straightforward to augment this application to allow the creation of leader summaries. The augmented application provided an interface that allowed the leader to more easily prepare a report and was also instrumented to collect data about their behavior. Instrumentation included mouse

[1]Italicized text shows the report-writer's highlighting for this time-log.

and keyboard level events (we do not report any analysis of these data in this paper).

3.1 Data Collection Process

Following [Garera and Rudnicky, 2005], the leader selected items from a display of all items from the student reports. Figure 1 shows the interface used for the data collection. The leader was instructed to go through the items and select a subset for inclusion in the report. Selection was done by highlighting the "important" words/phrases in the items (described to the participant as being those words or phrases that led them to select that particular item for the report). The items with highlighted text automatically become the candidate briefing items.[2] The highlighted words and phrases subsequently were designated as custom user features and were used to train a model of the user's selection behavior. Examination of the data indicated that users followed these instructions and did not, for example, simply select entire items or just the first word.

3.2 Data Collected

We were able to collect a total of complete 61 group-weeks of data. One group-week includes the time logs written by the members of a particular group and the associated extractive summaries. The class consisted of two stages, design and implementation, lasting about 6 and 9 weeks respectively. The groups and roles were reconstituted after the first stage. This meant that after the first stage we had new "authors" and could not combine the first stage data with the second stage into a continuous sequence. There were 5 groups in the first stage with average 4.8 students per group and 5 groups in the second stage with average 3.6 students per group (several students dropped the course after the first few weeks). There were on average 4.5 weekly summaries per group for Stage 1 and 7.5 weekly summaries per group in Stage 2. As is evident from the averages, not all the groups submitted a summary every week. To provide consistent data for development, testing and analysis of our system, we selected those 3 groups from the later stage of the class that produced reports most consistently (these are described further in the Evaluation section below).

4 Learning System

We modeled our Learning process on the one described by [Garera and Rudnicky, 2005]; that is, models were rebuilt on a weekly basis, using all training data available to that point (i.e., from the previous weeks). This model was then used to predict the user's selections in the current week. For example, a model built on weeks 1 and 2 was tested on week 3. Then a model built on weeks, 1, 2 and 3 was tested on week 4, and so on.

Because the vocabulary varied significantly from week to week, we trained models using only those words (features) that were to be found in the raw data for the target week, since it would not be meaningful to train models on non-observed features.

The resulting model is used to classify each candidate item as belonging in the summary item or not. The confidence assigned to this classification was used to rank order the raw items and the top 5 items were designated as the (predicted) summary for that week. The following sections explain the learning system in more detail.

4.1 Classifier Settings

Features

The features used in the classifier are words (or unigrams), stemmed and with stop words removed. We experimented with three different classes of features: a) **NEfalse**: only unigrams b) **NEclass**: unigrams + abstracted named entities using only the NEclass label (i.e., person, organization etc.) eg. 'White House' is treated as a unique token ('Location') representing its class.[3] c) **NEuniq**: raw unigrams + each named entity substituted by a unique token eg. 'White House' is treated as a unique token ('Location-1'). We used BBN-Identifinder [BBN-Technologies, 2000] for extracting the Named Entities.

Feature Extraction

We extract the features in two different settings. Firstly, we use the entire sentences for extracting the unigram features: F_{Raw}. Secondly, we combine the entire sentence features with the user-specific features:F_{User} ($F_{User} \subseteq F_{Raw}$) which is similar to the idea of combining the 'context' and 'annotated keywords' described in [Zhang *et al.*, 2003].[4] The details of how the final scores were computed are given below.

System Description

Standard Information Retrieval (IR) metrics are used to score the features. We did not fix the parameters in the schemes as the characteristics of the data did not particularly recommend any particular setting. The tunable parameters in the schemes and their possible values are: a) Term weighing method - TF (term frequency), TF.IDF, Salton-Buckley(SB) [Yates, 1999]. b) Corpus for measuring IDF: For any word, the inverse document frequency can be obtained by considering either the documents in the training set or the test set or both. Therefore we have three different ways of calculating IDF. c) Normalization scheme for the various scoring functions: no normalization, L1 and L2.

Feature scoring in the first setting of extracting unigram features F_{Raw} is straightforward using the above mentioned IR parameters (TF, TF.IDF or SB). For combining the scores under the second setting with the 'user-specific' features we used the following equation:

$$S_f = (1 + \alpha) * S_{f_{base}} \quad (1)$$

[2]The interface enforces a minimum (10%) and maximum (40%) number (as percentages of the total number of items for the particular week) of item selections to make sure some summary is reported.

[3]The idea being to capture user's preference wrt particular classes of NEs i.e. the user prefers to select an item where a person and organization are mentioned together.

[4]We also experimented with using just the user-specific features in isolation but found these less useful than a combination of all features.

Week No	Group1				Group2				Group3			
	TNI	ANW	NIS	ANS	TNI	ANW	NIS	ANS	TNI	ANW	NIS	ANS
1	8	8	2	2	15	8.6	6	2	21	6.9	5	3
2	10	11	4	1	15	8.6	5	2.8	13	9.5	3	3
3	26	7	7	2.6	24	7.4	7	1.4	22	7.5	4	1.3
4	18	7.9	5	1.6	17	5.5	3	2.7	11	3.2	2	2
5	25	8.6	6	4.3	18	10.7	3	8.3	18	7.7	4	7.3
6	15	8.5	2	2.5	16	14.7	7	6.1	12	10.3	2	5
7	20	8.4	3	1.3	24	13.4	7	7.4	17	11.7	3	3.3
8	25	9.3	6	3.8	26	7.2	6	1.7	16	9.5	4	8
9	28	7.3	4	2.5	-	-	-	-	-	-	-	-

Table 1: Table showing the week-wise item details for the three selected groups. TNI - Total Number of Items in that week, ANW - Average number of words per Item in that week, NIS - Number of Items selected for that week, ANS - Average number of words highlighted per selected Item for that week

where α is the weight contribution for the user-specific features and $S_{f_{base}}$ is the base score (TF or TF.IDF or SB). We empirically fixed α to '1' for the current study.

We tested the above mentioned variations of feature description, feature extraction and feature scoring using four learning schemes: Naive Bayes, Voted Perceptron, Support Vector and Logistic Regression. In the event, preliminary testing indicated that Support Vector and Logistic Regression were not suited for the problem at hand and so these were eliminated from further consideration. We used the Weka [Witten and Frank, 2000] package for developing the system.

4.2 Evaluation

The base performance metric is Recall, defined in terms of the items recommended by the system compared to the items ultimately selected by the user.[5] 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. The objective functions that we used for selecting the system model (built on the basis Recall) are:

1. *Weighted mean recall* (WMR): of the system across all weeks. The weeks are given linearly increasing weights (normalized) which captures the intuition that the performance in the later weeks is increasingly more important as they have consecutively more training data.
2. *Slope of the phase-wise performance curve* (Slope): We first calculate the three phase-wise recall performance values (normal average of the recall values) and then find out the slope of the curve for these three points.

Note that these metrics are used as a selection criterion only. Results in Figure 2 are stated in terms of the original Recall values averaged over the phase and across the three users. We compare these with the results for the random baseline. The random baseline is calculated by randomly selecting items over a large number (10000) of runs of the system and determining the mean performance value. [6]

[5]We are not doing a ROUGE-like [Lin, 2004] evaluation as we have modeled our problem as creating a draft summary in terms of the log items to be selected and not a final polished summary.

[6]For our domain, other baseline methods which have been traditionally used such as the 'first' sentence of a document etc. are not suitable because of a characteristically different corpora.

5 Experiment and Results

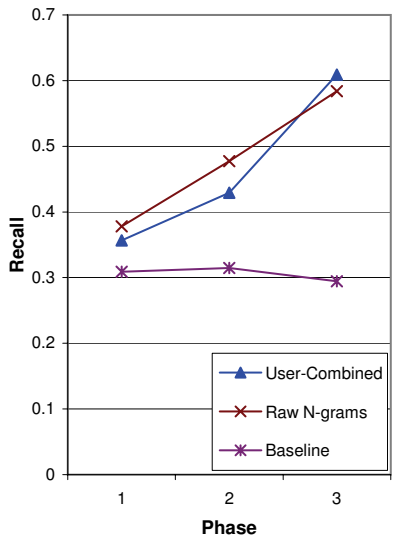
We selected for experimentation the three groups that most consistently generated complete weekly datasets . These groups had 9, 8 and 8 complete weeks of data. Detailed statistics of the data are shown in Table 1. Since the groups had different number of weeks, we grouped the observations into three phases by combining the results from successive weeks to create a phase. Merging the data into three phases is also consistent with our intuition about the task, that is, an activity has an initial starting period, middle activities and then the closing activity. We combined the observations with a sliding window to form the three phases. The window size and the overlap were different for the groups, though the same within a group.

Since we treated modeling technique as a free variable, we ran experiments for all the possible combinations of feature and system dimensions described in the previous section. The overall best performing system based on the jointly optimized metrics of WMR and Slope was selected as the final model from which to generate the system predictions. The final model has the following system parameters:- Learning Scheme: Voted Perceptron, Term Weighing Scheme: Salton-Buckley, Document Frequency Set: Training set, Normalization scheme: None. The feature set that gave best performance was NEuniq.

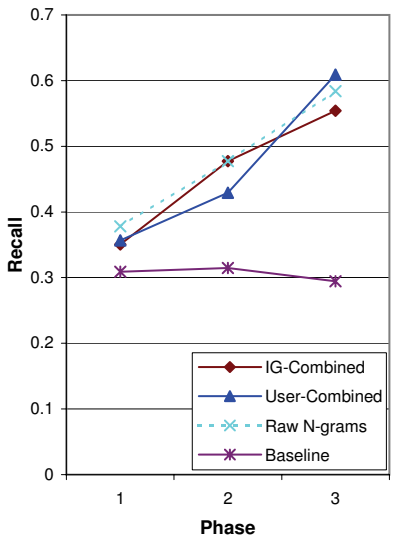
Figure 2 shows the key results of our experiments with the above-mentioned model. The performance is averaged across the three groups.

Figure 2(a) shows that the performance obtained with generic n-gram features is comparable to the performance incorporating user-specific features. While one would have hoped that user-specific features would do better than generic ones, it's clear that users select consistent personal features.

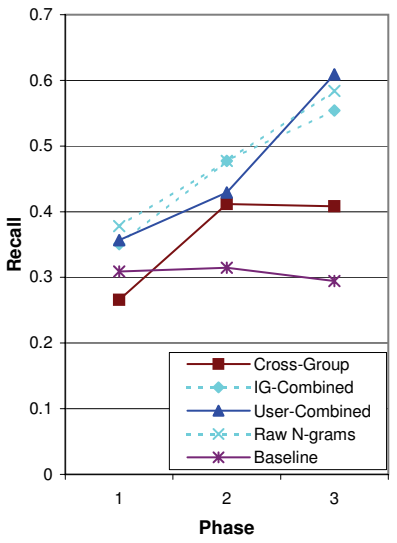
Figure 2(b) shows that user-selected features are at least as good as those selected based on information gain, arguably an "optimal" way to select useful features. Table 2 shows the mean number of active features in each phase (that is, how many previously identified features occur in the test set). Here we see something interesting, which is that information



(a) Recall Values for the Final model for Individual Users comparing the Raw N-gram features with User-Combined features



(b) Recall comparison between Information Gain selected features and User selected features



(c) Recall comparison between Individual user training and cross-group training

Figure 2: Figure showing the various experiments. 'Raw N-grams' represents the entire n-grams in the items. 'User-combined' represents the group-wise (single user) selected features combined with Raw n-gram features. 'IG-Combined' is the Information Gain selected features combined with Raw n-grams. 'Cross-group' is the training data pooled across groups and the user's selections combined with Raw n-grams.

gain (IG) selects very few features, while humans select a broader number, not only those that might directly predict item importance, but also features that could in the future be of use. It is also curious that there is little overlap between the IG selections and the human ones. One interpretation is that humans are selecting a broader and potentially more robust set of features. The latter is supported by the observation that the IG-based model fails to select any active feature 27% of the time, while the human model fails only 9% of the time for this reason.

Figure 2(c) shows what happens if we pool user-selected features across any two groups and use these to train a single model for testing performance on the third group. Here we see something different: cross-group features do not perform as well as group-specific features. This suggests that the specific features do matter, as they reflect a particular group's activities (as well as perhaps the leader's emphases in his or her reports). Table 3 is comparable to Table 2 and indicates that, while there is better overlap, pooling data introduces significant error into the model. Together these results suggest that important features are both task-specific and user-specific. The latter interpretation is consistent with the results reported by [Garera and Rudnicky, 2005].

Table 4 shows the most frequent active features selected under the conditions of Single User selection, Information Gain and the Cross-group pooled data.

6 Conclusions

We show that consistent summarization models can be built from relatively sparse user-generated data and that these models are sufficiently consistent to predict, with increasing accuracy, the summarization behavior of these users. We also show that naive end-users are able to consistently select features at least as well as an automatic process. There is moreover evidence that the feature sets selected by humans may turn out to be more robust in the long run than automatic features (since predictive power is spread over a larger number of features). Although this in itself is a useful result, it also opens the possibility of understanding and eventually au-

Phase No	IG-selected	SU-selected	Overlap
1	1.1	5.5	0
2	1.3	11.1	0.6
3	1.6	17.6	0.4

Table 2: Table showing the Phase-wise number of features under the Information Gain (IG) and Single User (SU) selected conditions. Last column shows the features that are common between the two techniques.

Phase No	CG-selected	SU-selected	Overlap
1	5.5	5.5	2.6
2	12.1	11.1	5.1
3	20.1	17.6	8.9

Table 3: Table showing the Phase-wise number of features under the Cross-group pooled (CG) and Single User (SU) training data conditions. Last column shows the features that are common between the two techniques.

SU-selected	IG-selected	CG-selected
work	group	work
test	PDA	class
class	sensor	test
boat	gp	boat
sensor	creat	sensor
script	GPS	PDA
code	schedul	script
PDA	io	system
data	timelin	phase
trip	input	code

Table 4: Table showing the most frequent active features selected under different conditions sorted by frequency: Single User (SU), Cross-group pooled (CG) and Information Gain (IG) selected conditions.

tomating the apparently sophisticated feature selection process used by humans.

These results, together with the implicit-feedback paradigm embodied in the report-generation interface suggest that it may be possible to design viable learning systems that can be effectively trained by an end-user. Such an approach to customization may be more powerful than ones based on adaptation, where models and features sets may be pre-determined, and that may produce system behavior that more faithfully reflects the needs of individual users.

Acknowledgement

We would like to thank Daniel P. Siewiorek and Asim Smailagic for allowing us to do data collection in their class and Susan Finger and team for helping us in instrumenting the data collection. We would also like to thank Eric Nyberg, Eric Riebling and team for the Named Entity annotations. This work was supported by DARPA grant NBCHD030010. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

[Amini, 2000] M.R. Amini. Interactive learning for text summarization. In *PKDD/MLTIA Workshop on Machine Learning and Textual Information Access*, 2000.

[BBN-Technologies, 2000] BBN-Technologies. Identifinder user manual. 2000.

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

[Dunning, 1993] T. Dunning. Accurate methods for the statistics of surprise and coincidence. In *Computational Linguistics*, 1993.

[Elhadad *et al.*, 2005] N. Elhadad, K. McKeown, D. Kaufman, and D. Jordan. Facilitating physicians' access to information via tailored text summarization. In *AMIA Annual Symposium*, Washington, DC, USA, 2005.

[Garera and Rudnicky, 2005] N. Garera and A.I. Rudnicky. Briefing assistant: Learning human summarization behavior over time. In *AAAI Spring Symposium on Persistent Assistants*, Stanford, California, USA, 2005. The AAAI Press.

[Leuski *et al.*, 2003] A. Leuski, C.Y. Lin, and E. Hovy. ineats: Interactive multi-document summarization. In *41st Annual Meeting of the Association for Computational Linguistics*, Sapporo , Japan, 2003.

[Lin, 2004] C.Y. Lin. Rouge: a package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out, ACL*, Barcelona, Spain, 2004.

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

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

[Nenkova and Passonneau, 2004] A. Nenkova and B. Passonneau. Evaluating content selection in summarization: The pyramid method. In *HLT-NAACL*, Boston, USA, 2004.

[Newman and Blitzer, 2002] P. Newman and J. Blitzer. Summarizing archived discussions: a beginning. In *Intelligent User Interfaces*, 2002.

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

[Raghavan *et al.*, 2006] H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on both features and instances. *Journal of Machine Learning Research*, 7:1655–1686, August 2006.

[Rambow *et al.*, 2004] O. Rambow, L. Shrestha, J. Chen, and C. Laurdisen. Summarizing email threads. In *HLT-NAACL*, 2004.

[Witten and Frank, 2000] I.H. Witten and E. Frank. *Data mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 2000.

[Yates, 1999] B.R. Yates, R.B.and Neto. *Modern Information Retrieval*. ACM Press, 1999.

[Zechner, 2001] K. Zechner. Automatic generation of concise summaries of spoken dialogues in unrestricted domains. In *SIGIR*, 2001.

[Zhang *et al.*, 2003] H. Zhang, Z. Chen, W.Y. Ma, and Q. Cai. A study for document summarization based on personal annotation. In *HLT-NAACL 2003 Workshop: Text Summarization (DUC03)*, Edmonton, Alberta, Canada, 2003.

[Zhou and Hovy, 2005] L. Zhou and E. Hovy. Digesting virtual geek culture: The summarization of technical internet relay chats. In *Association for Computational Linguistics*, Ann Arbor, MI, USA, 2005.