Master of Science Thesis Proposal

Abstractive summarization with a 'Briefing' focus

Mohit Kumar

Master Student
Language Technologies Institute
School of Computer Science
Carnegie Mellon University

Committee:

Alexander I. Rudnicky (thesis advisor)
Jaime Carbonell
Eric Nyberg
Roni Rosenfeld

Abstract

I propose to investigate the problem of abstractive summarization in the novel setting of a 'briefing' task. The novelty of the work is two-fold. Firstly, the task that we explore for summarization is a novel scenario where a user is expected to 'brief' his manager/superior with the summary of activities of a given time period based on a related set of text documents. Secondly, we pose the problem of summarization as modeling user's behavior instead of the traditional summarization approaches that do not distinguish between users. This formulation is motivated by the low inter-user agreement observed in abstractive summaries which strengthens the argument that each user has a different perception and hence different abstractive model. Thus summarization should be treated as a per-user learning task.

The field of abstractive summarization is an enduring research topic; however, the current state-of-the-art in abstractive summarization is not at all mature because of shortcomings in semantic representation, inference and natural language generation. The summarization task that is being focused on currently in the wider NLP community is extractive in nature (e.g. DUC evaluations) which is considered simpler than the abstractive task. Moreover debates continue about a proper evaluation methodology for even the extractive summaries. I propose to tackle a simplified problem in the domain to circumvent the following open issues: deep semantic representation and understanding of the documents, coherent natural language generation of summaries and evaluation of the summaries. I would focus on: encoding the summarization processes in the representation, modeling the user's summarization behavior and evaluating the summaries in the reduced semantic representation. Thus the reduced problem:

Given a simplistic graph-based semantic representation of the documents and the summary, learn the user's behavior for abstractive summarization in a briefing scenario.

1. Introduction and Motivation

Text summarization is the process of automatically creating a compressed version of a given text that provides useful information for the audience. The information content of a summary depends on audience's needs. Topic-oriented summaries focus on the audience's topic of interest, and present the information in the text that is related to the specified topic. On the other hand, generic summaries try to cover as much of the information content as possible.

Summarization approaches fall in two broad categories: extractive and abstractive. Extractive summarization produces summaries by choosing a subset of the sentences in the original document(s). This contrasts with abstractive summarization, where the information in the text is rephrased. Although summaries produced by humans are typically not extractive [2] and [18], most of the summarization research today is on extractive summarization (e.g. DUC evaluations). The problems in abstractive summarization, such as semantic representation, inference and natural language

generation, are relatively harder compared to a data-driven approach such as sentence extraction.

The novel summarization task that we are working on is a briefing scenario which is different from the newswire summary generation and other related tasks that have been explored in the NLP community. To illustrate the briefing scenario, consider a busy military personnel who is expected to send weekly reports to his superiors detailing the major relevant activities of the week. The personnel may utilize a number of sources of information for generating his report but a significant source is the set of emails that he sent and received during the week which reflect the activities of the week. So we may assist the personnel by automatically generating an abstractive summary using this set of text documents keeping in mind the audience for the summary.

Moreover every individual has a different summarizing behavior as is evident from poor inter-user agreement in the summaries which has been pointed out by the studies [2], [10] and [16] amongst others. So we propose to formulate our problem as learning the user's abstraction behavior following the interactive learning paradigm rather than the traditional abstractive problem. Thus our aim is to model the user's summarization behavior and produce successively better summaries. We conjecture that it is possible to learn useful abstractive summaries by extending the extractive techniques using additional representational mechanisms and semantic understanding of the content.

The approach that we are proposing is based on assuming a graphical model representation of the shallow semantic parse of the documents and the corresponding summary, similar to [14] and [6]. The motivation for using graph based approach is that they lend themselves easily to model/event based inference [14] which is desired since the data that we handle for our briefing task are typically activity and events. Also graph based models intuitively capture the semantics of the documents and are easier to visualize. Thus the approach is that we reduce our problem to a graph reduction problem where given a big graph representing the entire document collection it is to be reduced to a small graph corresponding to the summary of the documents. This graph reduction requires definition of the processes corresponding to the task of abstraction by which we 'reduce' the graph. These processes are exactly the same as those that are being researched by the community like semantic entailment [11], [19], event-based reasoning [14] and we would be using the current knowledge in these areas.

1.1. Related Work

The 'briefing' task scenario that we are working on is novel and has not been explored as yet. The closest tasks, which are based on similar corpus of documents as our task, are email thread summarization [3], IRC chat summarization [4] and discussion summarization for online classroom [5]. However, these tasks are dialog summarization tasks and make use of the dialog structure of the underlying corpus. Thus they are different from our task which is not a dialog summarization task by its description.

A great deal of work has been done in using the graph based representation of the documents for summarization tasks. [7] uses graphical models for word clustering while

[8] uses it for prepositional phrase attachment which are relevant sub-problems in summarization. [9] and [12] use a sentence level graph representation for extractive summarization where each sentence is represented as a node in the graph. [6] uses the predicates and arguments as the nodes in the graph and the relationship as edges and uses this graph for coming up with an event–centric summary which is close to what we want to achieve albeit in a different domain and task setting. [13] extends the approach in [6] with a different choice of nodes for the graph applied to an extractive summary task.

Some work in reasoning [17] which has been applied to the field of Question-Answering [14] in utilizing semantic parsing and structures is also relevant to our problem. In [14] the authors describe a scheme for question answering by identifying predicate argument structures and semantic frames from the input and then performing structured probabilistic inference using the extracted relations in the context of domain and scenario model. The piece of work that is most relevant to our problem is their representation of actions and events and inference related to them [17]. However, the domain models are still extracted manually and work is under progress to do so automatically from resources like semantic web etc.

Interactive learning (human-in-the-loop) paradigm has also been investigated previously in the context of text summarization. [15] describes a query-relevant text summary system based on interactive learning. Learning is in the form of query expansion and sentence scoring by classification. Another summarization system based on user's annotations was proposed by [16]. Annotations and their contexts are extracted to represent features of sentences, which are given different weights for representation of the document. These systems however are extractive summarization systems and the underlying text corpus that they use is also different.

2. Work to date

2.1. Data collection

The 'briefing' scenario that we are considering is a course project class at CMU. In the class, the students are divided into groups and are expected to report weekly as a group to the instructor. The underlying text corpus is the activities log that each student is required to maintain about the time spent on different activities related to the class. The detailed Experimental setup is described below.

2.1.1. *Experimental Scenario*- The scenario of our experiment was a project based class at CMU. The entire class was divided into groups working on different projects. Each group had well defined roles such as leader, editor etc. Each student in the class was required to log the time spent on the different activities related to the course. Each time-log entry consisted of the following fields: date, category of activity, time spent and details of the activity. Category of activity is selected from a predefined set of activities which represent the type of activities in the class like Coding, Group meeting, Research etc. A sample log entry is as shown in the figure 1. Each student was also required to answer three weekly questions related to the project. The questions are shown in Figure 2.

These questions are meant to capture the information about the activities which are not covered in the time-log and are not necessarily a repetition of the time-log data. Based on the above mentioned inputs from each student in the group, the group leader was required to produce a weekly extractive summary for the instructor. The basic unit for the extractive summary was a single time-log entry or the answer to the weekly questions. In addition, the leader was also asked to generate a free form abstractive summary.

2.1.2. *Experimental Method*- We used the online collaboration system developed at CMU called KIVA for logging the activities. The tool provides an easy interface for doing collaborative work with facility for discussions, postings, file storage etc. The interface had the provision for entering the time-logs as well as the weekly questions. Based on this interface we had another interface for the group leaders which presented all the time-logs and answers of the group members in a single interface in which they could mark the items to be selected for the extractive summary. In order to select an item as a summary item we required the leader to highlight a phrase(s) in the item that made the item summary-worthy. The idea for the phrase highlighting was that this would help us define user-defined features for the learning task and also tell us about the important concepts that make an item summary worthy. The summarization interface had a free form text field for the user to write down the abstractive free flowing summary.

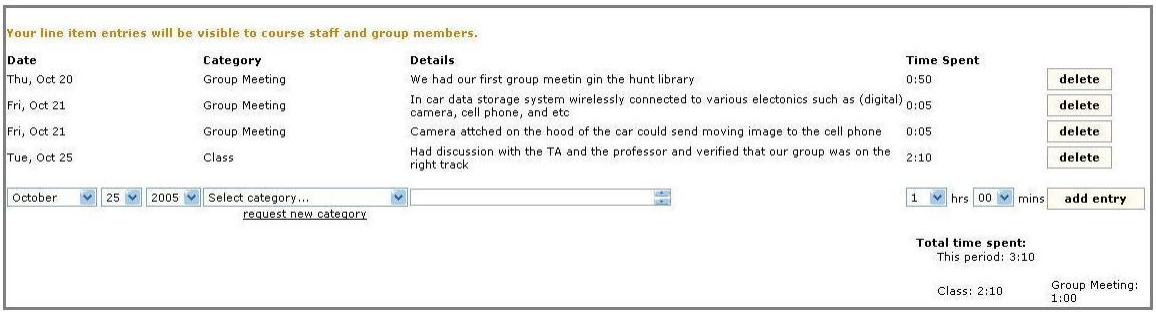

Figure 1: A sample worklog entry

2.1.3. *Data Collected*- We collected a total of 62 person week worth of data. One person week data includes the time logs and answers written by the members of a particular group and the associated extractive and abstractive summaries. Extractive summary have the highlighted phrases, which indicate the important concept of the extracted unit, associated with them while the abstractive summary is a free-form summary. There were two phases in the class and the groups were changed after the first phase. We had 5 groups in the first phase and 5 groups in second phase. There are on average 4.5 weekly summaries per group for Phase 1 and 7.5 weekly summaries per group in Phase 2.

We have also collected the data representing the click-activity of the user, a potential information stream for learning the user's summarization behavior.

2.1.4. *Preliminary Data Analysis*- The data collected is a mixture of good and bad data. By good data, I refer to weekly summaries where the student has written nice abstractive summaries and there is a good observable linkage from the worklogs and answer entries

to the summary. These linkages observed in the previous year's class data collection had motivated us to orchestrate this particular class' data collection. By bad data, I refer to cases where either the abstractive summary is not well-formed which may be because either the user has just copied the text or the number of entries in the worklogs is few and the summaries are not an abstraction in any significant way.

One general observation is that since the summary is for a week's activities there is not much event abstraction and it tends to be more extractive than abstractive. However the data are useful for studying semantic entailment and deletion techniques for abstractive summarization.

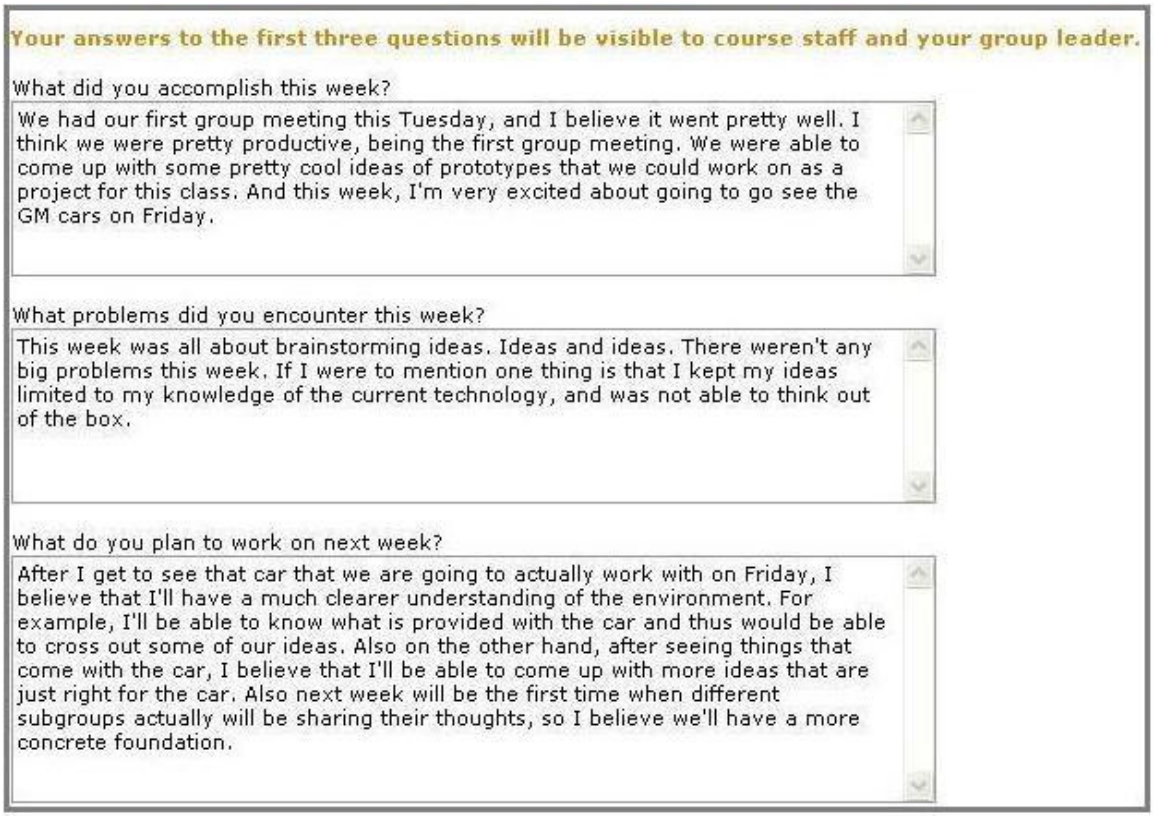

Figure 2: Sample answer entries showing the three weekly questions

2.2. Extractive summarization

- We developed an n-gram based extractive summarization system. I am currently running experiments to evaluate the performance of the system on the course data for the extractive task.
- We tried to capture user dependent features by requesting the user to mark the important concepts in the extracted units. Preliminary results that we have observed are mixed. We are still investigating the best way to make meaningful use of the marking data that we have obtained. However, the poor results do not prove a flaw in the concept instead indicate the HCI problem that the user's tended to highlight the entire item as important concept instead of selecting a phrase(s).

- It was a stepping stone for understanding the task of abstractive summarization.
- System: We have developed our extractive system using Javelin's AnnotationsDB API for getting the parsed and annotated data. Figure 3 shows the high level data flow diagram of the system.

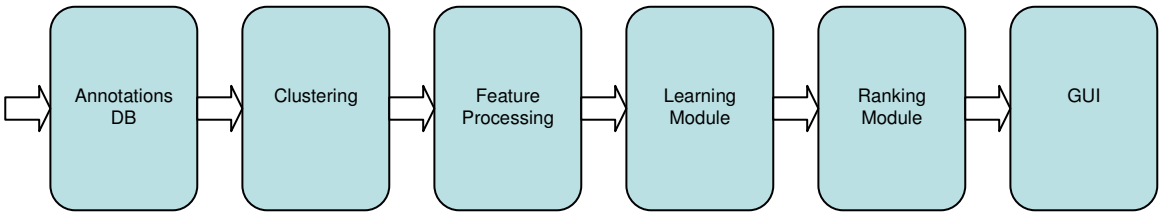

Figure 3: The data-flow diagram of the extractive summarization system.

3. Proposed work

I am proposing to survey and integrate the currently existing technologies/approaches in the different related areas like semantic entailing and event detection in a framework for doing abstractive summarization formulated as learning the user's preferences and behaviors in a briefing task. I realize that this is a difficult problem and that it currently lacks any mature solution in the area as yet and that I would have to follow an iterative design approach. I hope that this work will provide me with insights into the abstractive summarization task and will allow me to identify the short-comings as well as strengths of the relevant modules used in the task and will provide the basis for developing ideas of doing the task in a better way. I also realize that given the time constraints I would need to prioritize the modules and I have explained in the below sections my choice of the modules that I would be implementing.

So the proposed work is:

3.1. Integrated architecture for abstractive summarization (subsumes extractive summarization)

I will develop an architecture for performing abstractive summarization that would be modular and motivated by the general Question Answering (QA) approach in the QA community [14]. Defining this architecture would allow us to understand and evaluate the dependence of each of the modules on the overall task. In my current understanding, the various higher level modules that would be present in the architecture are: Semantic representation, Abstractive inference module, Natural Language Generation. A preliminary architecture is shown in Figure 4 sans the details for the Natural language generation which is not a concern/focus of this work. To illustrate the overall working of the architecture, I show a sample semantic document graph, borrowed from [13], in Figure 5 and the corresponding summary in Figure 6. Under the given representation firstly we define the processes (as detailed in Task 3.3) for the graph reduction. We then use the output of each of the individual processes as a feature for learning the user's behavior. The output of the learning module is a representation of user model which may then be used in test cases for the graph reduction.

3.2. Graph based representation of documents and summaries

I would explore the various shallow semantic parsers that can be used to get the semantic parse for our document set. Some of the candidates are ASSERT, NLPWin and other Link parser based systems. Based on the output of the semantic parse I would come up with a graphical representation best suited for the parse motivated by [6], [13] and [14]. However the choice of the representation would be heavily guided by the existing technologies in the subsequent modules especially semantic entailing and event modeling which use this representation for inference.

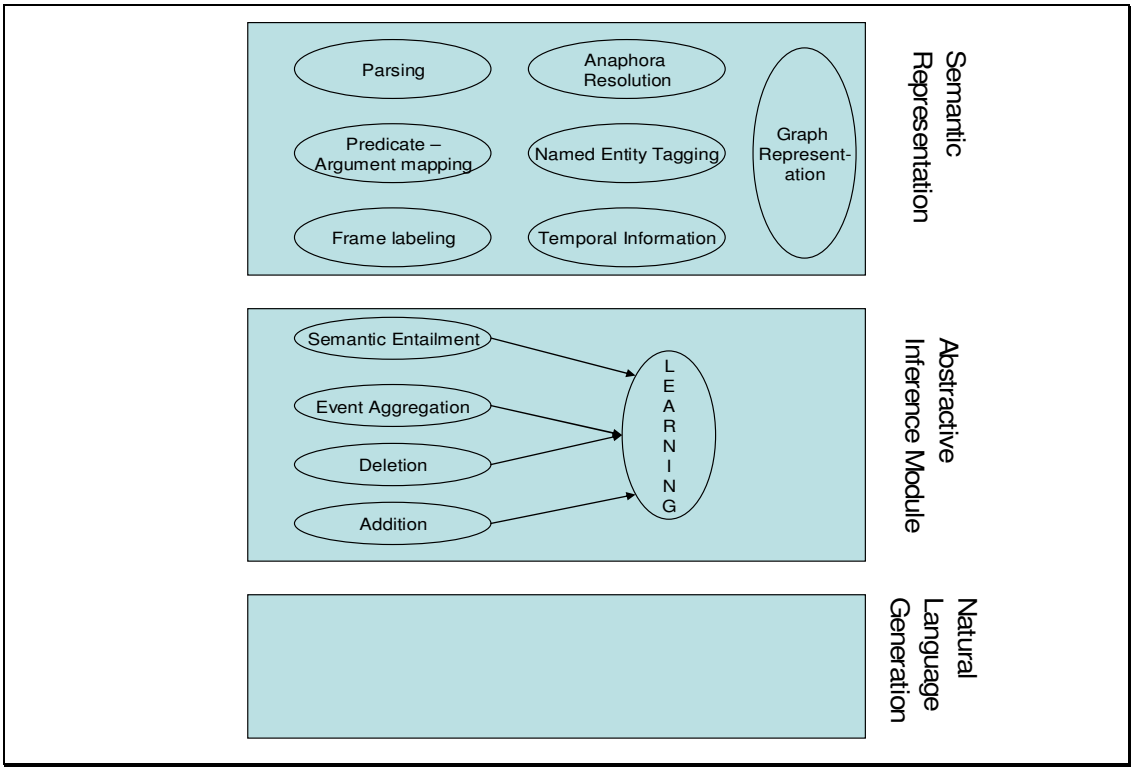

Figure 4: A draft of the proposed framework for abstractive summarization

3.3. Defining processes for abstraction

3.3.1. Based on my observation of the data, I hypothesize four processes/components for abstractive summarization. Figure 4 shows the following components under the Abstractive inference module.

- Semantic entailing – similar sentences *collapse* [11], [19].
- Sub-events *aggregate* to form events (model/event based inference [14])
- Unimportant facts are *deleted* like 'attended class' in the course data
- Domain dependent contextual facts are *added* like 'We would be working hard to get the module working by the end of next week'.

3.3.2. I would come up with a way to encode these processes as features to be used for characterizing the user model. For our current dataset, the most important component is semantic entailment followed by deletion. The event aggregation model is also important but since the data is at a week's level there's not much abstraction in terms of events so this would be low priority module for us currently. For example, in a week a student is

typically engaged in one major activity like ‘occupied with GPS sensor’ which he may mention in the worklogs as ‘played with GPS sensor’ , ‘GPS sensor testing’ and this should be captured by entailment. We would not be handing the ‘addition’ component as it requires discourse and domain knowledge which is hard to obtain.

I would be coming up with an implementation of a semantic entailment module motivated by [19] and other approaches proposed in the PASCAL-RTE [11] challenge workshop. The advantage of the approach in [19] is that it is based on lexical syntactic units. The model needs a knowledge base of basic patterns along with compositional probabilistic inference rules. So I need to come up with the basic patterns in the domain and the inference rules for the entailment task. Another approach that I would be exploring for implementing entailment is using a SCONE [21] based reasoning system. The idea behind using a knowledge base like SCONE, which is graph-based, is that it may subsume resources like FrameNet and WordNet and once we populate it with the domain dependent concepts we may reason about them which would be relevant for implementing semantic entailment.

An example to illustrate and motivate the difference in the entailment task and the event based inference:
Consider the following two sentences:
‘I was occupied with I/O module’ and
‘I developed the I/O module’.
Corresponding worklog entries:
‘I coded rest of the I/O module’,
‘I wrote the code for I/O module’, and
‘I tested all the sensors with the I/O module’.

Now the first sentence with predicate ‘occupied’ may be obtained by semantic entailment however the second sentence with predicate ‘developed’ requires event abstraction where the sub-events ‘coding’ and ‘testing’ should be combined to get the concept ‘develop’. For the first-cut of the abstractive system and given time-constraints, I would not be working on the event-based model.

For the sentence deletion module, I would train a supervised classifier by heuristically obtaining training data from the sample worklogs and the corresponding summary. A simple heuristic to get the data, to start with, would be a simple word matching where I would tag any sentence to be *deleted* if it doesn’t contain any word present in the summary. It can be extended by using resources like WordNet. I would be designing the features to be used in the classifier starting with the features ‘category_type’, unigrams etc. The confidence score of the classifier can then be used as a feature in the subsequent modules.

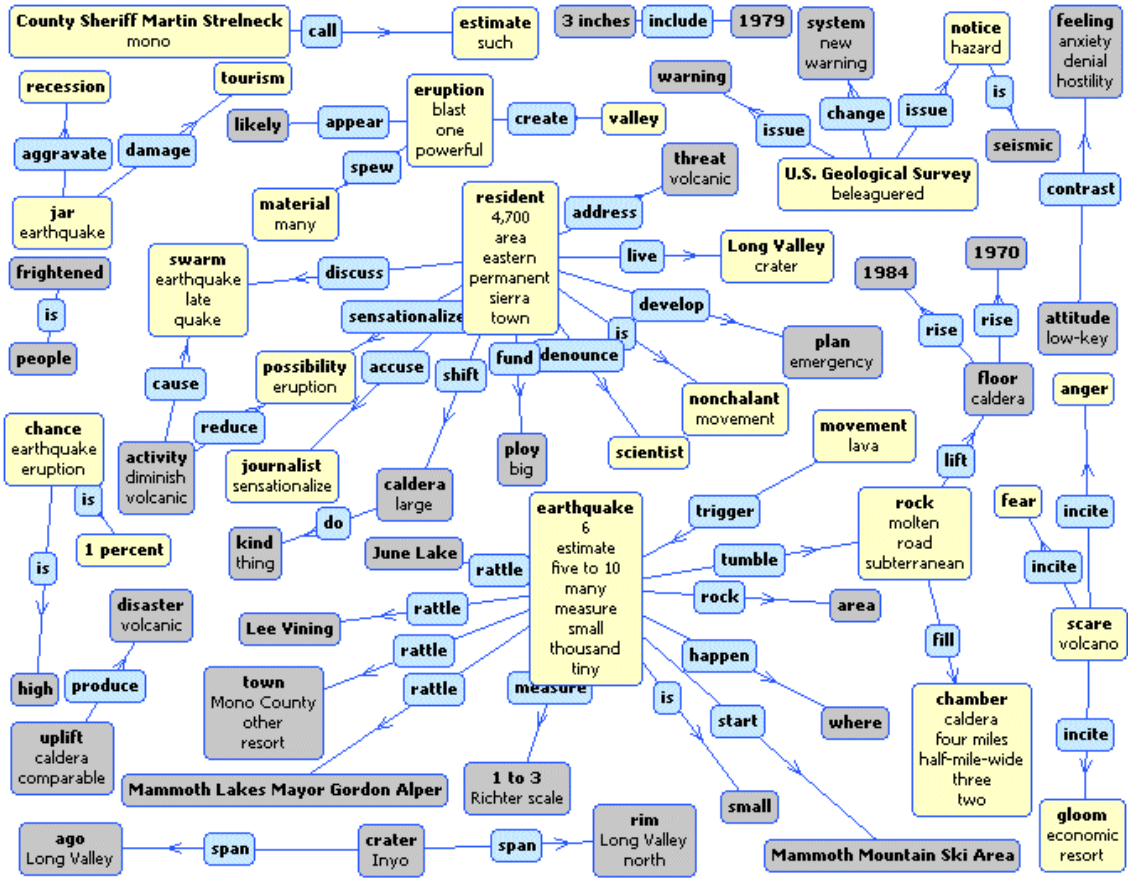

Figure 5: A sample semantic graph representation of a document borrowed from [13]

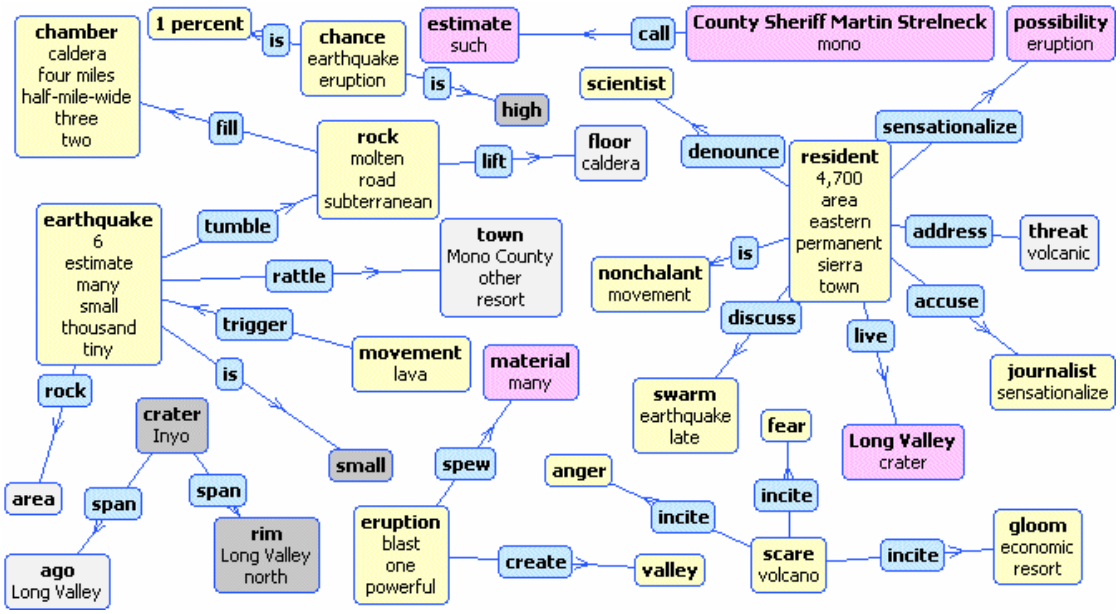

Figure 6: A sample semantic graph representation of the summary of the document shown in figure 5, borrowed from [13]

3.4. Modeling user's summarization behavior as graph reduction

Using the encodings of the abstraction processes, which may just be probabilities/confidence scores for the different processes, I would define a graph reduction framework to capture the user's behavior. The reason that we believe that it is a user-dependent model is the intuition that different user's may have different probability scores for different processes. For example, a user may believe that all the worklog entries of a certain category like 'attend class' are useless while some other user may not believe the same. Thus we would come up with a probabilistic graph reduction algorithm.

3.5. Evaluation

In order to demonstrate the utility of my approach I would come up with a reasonable way to evaluate the summaries generated by the system. Evaluation metrics [20] used in graph matching would be useful. Also human judgment in evaluating the quality of the reduced semantic graph corresponding to the summary would also be useful. I would try to orchestrate user studies with the actual 'consumers' (TA and the instructors) of the summary of the class data. I would be experimenting with the individual components like the semantic entailment and sentence deletion and would try to evaluate their individual as well as combined contributions.

4. Timeline

Nov end – Decide the Semantic entailment module design and requirements in terms of parsers
Dec end – Sentence deletion classifier – feature design and implementation + semantic entailment + design evaluation setup
Jan end – Semantic entailment module
Feb end – Graph reduction framework
Mar end – Experiments
Apr end – Thesis writing
May mid – Thesis defense

References

[1] Luhn, H. The Automatic Creation of Literature Abstracts. IBM Journal of Research Development, 1958.

[2] Lin, C.Y. and Hovy, E. The Potential and Limitations of Automatic Sentence Extraction for Summarization. In Proceedings of the HLT-NAACL Workshop on Automatic Summarization, Edmonton, Canada, 2003.

[3] Rambow, O., Shrestha, L., Chen, J. and Lauridsen, C. Summarizing Email Threads. In Proceedings of HLT-NAACL, Boston, USA, 2004.

[4] Zhou, L. and Hovy, E. Digesting Virtual "Geek" Culture: The Summarization of Technical Internet Relay Chats. In Proceedings of ACL, Ann Arbor, USA, 2005.

[5] Zhou, L., Shaw, E., Lin, C. Y. and Hovy, E. Classummary: Introducing Discussion Summarization to Online Classrooms. In Proceedings of HLT-EMNLP (demo), Vancouver, B.C., Canada, 2005.

[6] Vanderwende, L., Banko, M., and Menezes, A. Event-Centric Summary Generation. DUC 2004.

[7] Brew, C., and im Walde, S. S. Spectral clustering for German Verbs. In Proceedings of ACL, Philadelphia, USA, 2002.
[8] Toutanova, K., Manning, C., and Ng, A. Learning Random Walk Models for Inducing Word Dependency Distributions. In Proceedings of ICML, Banff, Canada, 2004.
[9] Erkan, G. and Radev, D. R. LexRank: Graph-based Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research (JAIR) 2004.
[10] Jing, H. Using Hidden Markov Modeling to Decompose Human-Written Summaries. In Proceedings of ACL, Philadelphia, USA, 2002.
[11] Dagan, I., Glickman, O. and Magnini, B. The PASCAL Recognising Textual Entailment Challenge. In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005.
[12] Mihalcea, R. Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. In Proceedings of ACL, Barcelona, Spain, 2004.
[13] Leskovec, J. Milic-Frayling, N. and Grobelnik, M. Extracting Summary Sentences Based on the Document Semantic Graph. Microsoft Research Technical Report MSR-TR-2005-07. January 2005.
[14] Narayanan, S. and Harabagiu, S. Question Answering based on Semantic Structures, International Conference on Computational Linguistics (COLING), Geneva, Switzerland, 2004.
[15] Massih, A. Interactive Learning for Text Summarization. In Proceedings of the PKDD/MLTIA Workshop on Machine Learning and Textual Information Access, Lyon, France, 2000.
[16] Zhang, H., Zheng, C., Wei-ying, M. and Qingsheng, C. A Study for Document Summarization Based on Personal Annotation. In proceedings of HLT-NAACL Workshop: Text Summarization (DUC), 2003.
[17] Narayanan, S. Reasoning about Actions in Narrative Understanding. In proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, 1999.
[18] Banko, M. and Vanderwende, L. Using Ngrams to Understand the Nature of Summaries. In Proceedings of NAACL, Boston, USA 2004.
[19] Dagan, I. and Glickman, O. Probabilistic textual entailment: Generic applied modeling of language variability. In Learning Methods for Text Understanding and Mining, Grenoble, France, 2004.
[20] Sanfeliu, A. and Fu, K.S. A Distance Measure between Attributed Relational Graphs for Pattern Recognition. IEEE Transactions on Systems, man and Cybernetics, 1983.
[21] Fahlman S.E. SCONE user manual, 2005.