An Ontology-based Visual Interface for Browsing and Summarizing Conversations

Shama Rashid
University of British Columbia
Vancouver, BC, Canada
shama@cs.ubc.ca

Giuseppe Carenini
University of British Columbia
Vancouver, BC, Canada
carenini@cs.ubc.ca

ABSTRACT
In this paper we present a visual interactive interface to create focused summaries of human conversations via mapping to the concepts within an ontology. The ontology includes nodes for the conversation participants, for Dialog Act (DA) properties such as decision, action-item or subjectivity, as well as for entities mentioned in the conversation. The classifiers used to annotate conversation data with DA property and entity tags can be applied to any conversational modality including face to face meetings, emails, blogs and chats. Our interface allows the user to explore these conversations and identify informative sentences by their association with nodes of interest on the tree-structured visual representation of the ontology. The sentences thus selected by the user as potentially important components of the summary can then be used to derive a brief and focused overview of the conversation. The display interface and data parsing components in the initial prototype were all developed based on Java frameworks and toolkits.

INTRODUCTION
Multimodal conversations have become an integrated part of our everyday communication with others. We email for business and personal purposes, attend meetings in person and remotely, chat online, and participate in blog or forum discussions. It is clear that automatic summarization can be of benefit in dealing with this overwhelming amount of interactional information. Automatic meeting abstracts would allow us to prepare for an upcoming meeting or review the decisions of a previous group. Email summaries would aid corporate memory and provide efficient indices into large mail folders. Summaries of technical blogs could become an important support platform for developers, administrators, and technology enthusiasts in general.

Summarization of human conversations have been addressed in the past for different modes of conversation, including meetings [7], emails [23, 3], telephone conversations [32] and internet relay chats [31]. In all these previous works the dominant approach to summarization has been extractive, which means that the summary is generated by selecting and concatenating the most informative sentences from the source document(s). Extractive summarization has been popular at least in part because it can be framed as a binary classification task that lends itself well to machine learning techniques, and does not require a natural language generation component. Extrinsic evaluations have also shown that, while extractive summaries may be less coherent than human abstracts, users still find them to be valuable tools for browsing documents [9, 15, 19]. However, these same evaluations also indicate that concise abstracts are generally preferred by users and lead to higher objective task scores. The limitation of a cut-and-paste summary is that the end-users do not know *why* the selected sentences are important; this can often only be discerned by exploring the context in which each sentence originally appeared. One possible improvement is to create *structured summaries* that represent an increased level of abstraction, where selected sentences are grouped according to the entities they mention as well as to phenomena such as *decisions*, *action items* and *subjectivity*, thereby giving the users more information on why the sentences are being selected. For example, the sentence *Let's go with a simple layout* is about a *simple layout* and represents both a decision and the expression of a positive subjective statement.

Our first attempt to build an interface to create visual structured summaries of conversations was presented in [2]. This interface relied on mapping the utterances of the conversation into an ontology, similar to the faceted browsers in [30, 6], that then could be used to search the conversation according to the annotation. Our ontology initially contained only the participants of the conversation and properties of the utterance such as whether it was expressing a decision, a subjective statement, etc. Our first prototype comprised two panels (see Fig 1). The right panel displayed the ontology, while the left panel displayed the whole conversation, where sentences are temporally ordered. Given the information shown in the two panels, the users could generate visual, structured summaries by selecting nodes in the ontology. As a result, the sentences that were mapped in the selected nodes would be highlighted.

In this paper we present a novel interface (see Fig 2) that addresses several limitations of our initial prototype. First, we have extended the ontology to also include entities mentioned in the conversation. Searching the conversation us-

Workshop on Visual Interfaces to the Social and Semantic Web (VISSW2011), Co-located with ACM IUI 2011, Feb 13, 2011, Palo Alto, US.

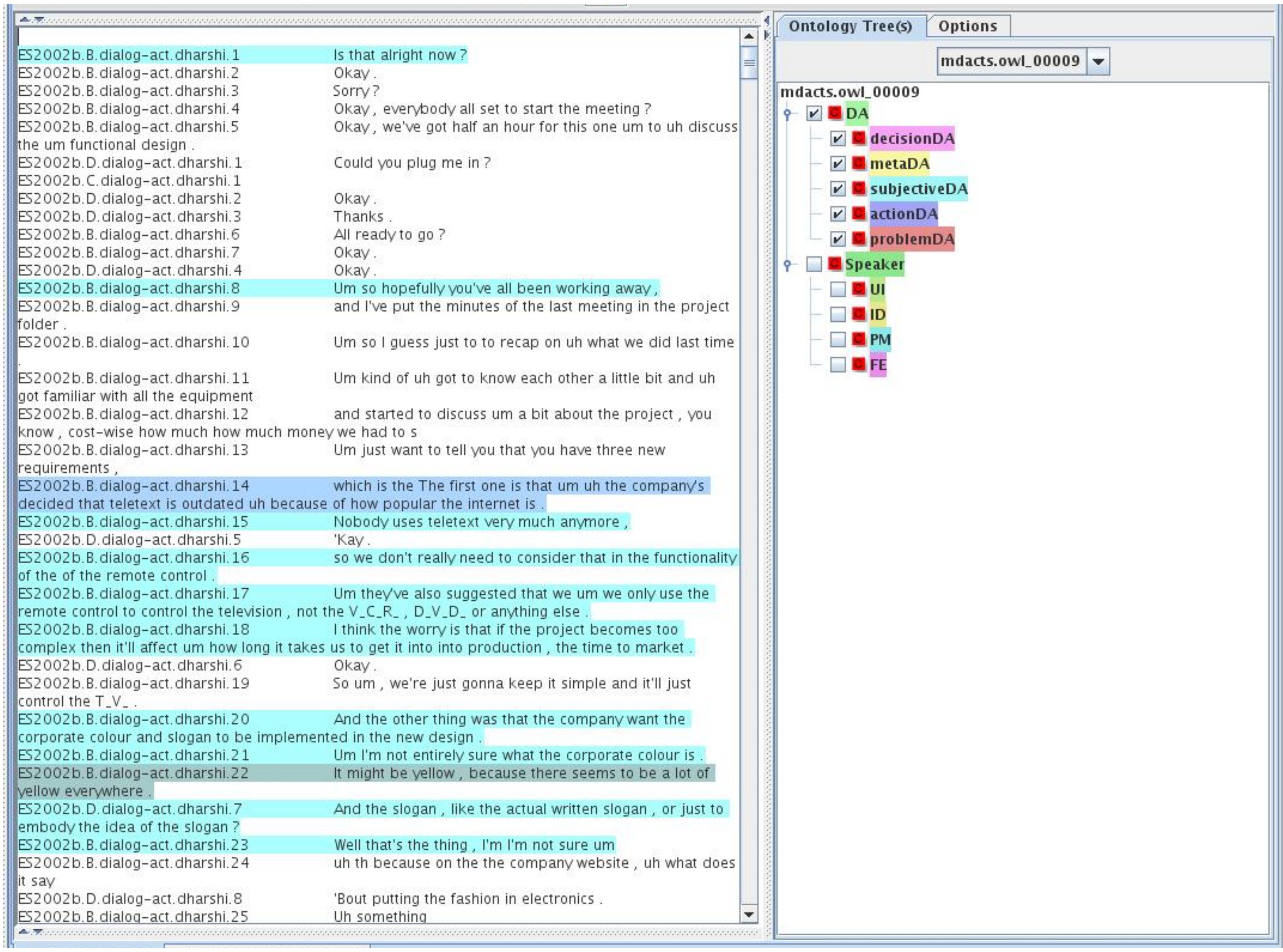

Figure 1. Initial prototype for generating visually structured summary of human conversations, as presented in [2]

ing a particular keyword is suitable only when users already have an idea about the content and want additional information on a particular entity. Our assumption is that representing entities on the ontology tree will not only enable the users to perform a more refined search and browsing of the conversation, but the entities would also provide them with a quick overview of the content of the whole conversation. Secondly, we have provided a satisfactory solution to the problem of highlighting the sentences mapped to nodes selected by the users in the ontology. Instead of using color (a non-scalable solution that we initially explored) we have added a column to the left of the interface layout, in which the (selected) mapping to the (knowledge concepts within) ontology of the corresponding utterance can be displayed. The third extension, the summary view (discussed in details at Display Design section), is the most critical one, as it opens the door to a possibly highly beneficial integration of structured visual (extractive) summaries and abstractive focused summaries. Our hypothesis is that the users, after they have inspected the conversation through the mapping to the ontology, may wish to generate summaries covering only some aspects of the conversation (which are especially relevant to their current information needs). For instance, users may need a summary of all the *positive and negative* comments that were expressed in the conversation about two particular entities (e.g., *new design* and *interface layout*). The new interface allows the users to trigger the generation of such summaries, which will be shown in the bottom panel (see Fig 2). Most importantly, these summaries can be generated either by extraction or by abstraction; in the latter case by applying techniques presented in [18].

We have developed our prototype using instances of meeting conversations from the AMI corpus [4]. We are currently extending the interface to Web-based modes of conversation like emails, blogs, chats etc. and working on this with the BC3 corpus [27]. As we move to asynchronous conversations, one additional complication is that the conversational structure is not linear anymore, but can be a tree or a graph. We are currently exploring how the interface can be extended to deal with more complex conversational structures. In contrast, the mapping of sentences to the ontology can be easily transfered from meetings to other conversational modes, as it relies on very general methods. The mapping is performed by first identifying all the entities referred to in the conversation (via syntactic parsing), and then by utilizing classifiers relating to a variety of sentence-level phenomena such as *decisions* and *subjective sentences*. High classification accuracy is achieved by using a very large feature set integrating conversation structure, lexical patterns, part-of-speech (POS) tags and character n-grams.

In this paper we will first describe related work. Then we will present the process of deriving an ontology and mapping sentences to it. After that, we shall discuss our interface to display the conversation to the users for interactive exploration of the data with the help of the ontology.

RELATED WORK

In HCI and NLP different approaches have been proposed to support the browsing and summarization of data/documents with the aid of an interactive interface. Here, we focus on the ones that are more critical for our current and future work.

The idea of using an ontology to explore data in an orderly manner is not novel. For instance, the Flamenco [30] and the Mambo [6] systems make use of hierarchical faceted metadata for browsing through image or music collections. In our approach we adopt similar techniques to support the exploration of conversations. More specifically, in Flamenco [30], while navigating an image collection along conceptual dimensions or facets (e.g. date, theme, artist, media, size, color, material etc.), every facet hyperlink that can be selected to derive a new result set is displayed with a count as an indicator of the number of results to expect i.e. the count works as a query preview. Similarly, we have included a count beside each node of the ontology to indicate the number of sentences in the conversation that have been mapped to it. Another idea we have borrowed from the Flamenco and Mambo systems is to use summary paths to simplify the user interaction with the ontology. In Flamenco, different paths may lead to a collection of images at a particular time; so Flamenco uses a summary path along the top of the interface to show exactly which path was taken and uses links along this path to retract to a previous decision along the path. Similarly, the Mambo system provides breadcrumb style filter history, which gives an interactive overview of the active facet filter. In our interface, to facilitate the inspection of a possibly large ontology, nodes can be minimized (i.e., their children are hidden). So, it may happen that the set of tags selected by the users is not fully visible. To address this problem, we are working on including a summary of the ontology node selection at the top of our interface, as it is done in Flamenco and Mambo.

An extractive approach for generating a decision-focused summary suitable for debriefing tasks has been proposed in [12]. This type of summary includes only 1-2% of a meeting recording related to decision making. In addition to the transcripts, the interface takes advantage of the audio-video recordings to better understand decision points. While the interface in [12] makes use of only dialog acts for focused summary generation, ours additionally uses speaker and entity information. Furthermore, we are not limited to extractive techniques as we are also exploring focused summarization by abstraction. The interface proposed in [12] also considers features that are specific to conversations about designing a new product (see AMI corpus [4]), in which you typically do not have only a single meeting but a series of meetings, the kickoff, the conceptual design, the detailed design, and the evaluation meetings. While we also aim to consider series of related conversation we intend to do it in a general way, i.e., not being limited to conversations about designing a product.

The Ferret Meeting Browser [28] provides the ability to quickly find and play back a combination of available audio, video, transcript and projected display segments from a meeting side by side for comparison and inspection synchronously and allows navigation by clicking on a vertical scrollable timeline of the transcript. Users can zoom into particular places of interest by means of a button and by zooming out they get an overview of the meeting in terms of who talked the most, what meeting actions etc. In the future, we'll extend our interface to include an overview of the conversation integrating ideas from the following projects.

The Meeting Miner [1] aids browsing multimodal meeting through recordings of online text and speech collaborative meetings using timeline navigators of content of edits as the main control for browsing. In addition, it can retrieve a set of speech turns spread throughout the conversation focused on particular keywords that can be selected from a list of automatically generated keywords and topic. The users can also navigate to the audio segments that have been identified as relevant using the audio timeline for random access of the file. The Meeting Miner [1] automatically identifies a set of potential keywords and the users can decide to view these in alphabetical order, ranked by term frequency or simply by time of appearance in the conversation. A similar concept has been discussed in the future work of FacetMap [25] where the authors mention implementing the ability to dynamically order the facets, such as by count, alphabetically by label, by usage, or by some specific facet ordering. The entities on the ontology tree of our interface are equivalent to Meeting Miner's keyword panel entries and we are currently listing the entities in alphabetical order; but a different ordering based on the count etc. may prove more helpful to the users.

The CALO meeting assistant [26] is used for capturing audio signals and optional handwriting recorded by digital pens for distributed meetings. During the meeting the system automatically transcribes the speech to text and the participants are fed back a real-time transcript to which annotations can be attached. At the end of the meeting the system performs further semantic analysis on the transcript like dialog act segmentation and tagging, topic identification and segmentation, question-answer pair identification, addressee detection, action item recognition, decision extraction and summarization. The result of this analysis is made available via a web-based interface. The off-line meeting browser interface displays the meeting transcript segmented according to dialog acts. Each dialog act is shown along side its start time, speaker, and a link for streaming audio feedback for the transcript segment (in case the users want to overcome any speech transcription errors). The CALO browser also provides the users views of the extractive summary of the meeting and above mentioned annotations in separate tabs. A lot of the annotations provided by the CALO system overlap with our segmentation of the transcript and knowledge concepts represented in the ontology tree but the CALO browser provides more flexibility by providing the users means to attach their own annotations, which is an interesting direction we could explore in our future prototypes. Our interface differs from CALO by providing a way to focus on the users' particular information need by referring to the ontology and by providing an option to generate abstractive or extractive summaries.

In iBlogVis [13], the authors use social interaction cues like comment length, number of comments, regular commenters etc. and content cues like topics of a blog, blogger's posting habits etc. to provide the users with an overview of a blog archive and to support them in deciding which entry to read. The font size of a tag for blog topic representation indicates its popularity, a concept that we shall employ in the future for our textual collage representation of conversation content. iBlogVis uses the idea of read wear [10], a means of graphically portraying the documents readership history, to help users keep track of entries that have been read, have not been read, or the one that is currently being read using different colors. Similarly, we are currently working to provide users an option to log the current ontology settings so that they can keep track of the combinations tried before.

MostVis [24] uses a multiple co-ordinated view for browsing a catalog for multimedia components in a car. Besides the textual label of each node in the catalog node-link tree representation there is an additional icon representing element type (car series, function block, functions, parameters etc.). This is similar to our use of a short string representation or icon beside the ontology tree nodes. MostVis also has a history window with undo and redo button where an entry is logged every time an expansion or minimization of the node-link tree occurs. We are exploring how a similar mechanism could be added to our interface.

MAPPING SENTENCES TO ONTOLOGY

Our summarization method relies on mapping the sentences in a conversation to an ontology written in OWL/RDF (Web Ontology Language/Resource Description Framework), a widely used open-source standard develop tool compatible with the architecture of the Semantic Web in particular.

Our ontology contains three core upper-level classes: Participant, Dialog Act (DA) types and Entities. When additional information is available about participant roles in a given domain, Participant subclasses can be utilized. For our AMI meeting scenarios the Participant class consisted of four subclasses ProjectManager (PM), IndustrialDesigner (ID), UserInterfaceExpert (UIE) and MarketingExpert (ME). The DA-Type class, on the other hand, contains subclasses decisions, actions, problems, positive subjective sentences, and negative subjective sentences.[1] The entities in a conversation are noun phrases with mid-range document frequency. Our ontology is populated with the instance data for a given conversation that the user is attempting to browse and summarize.

Our definition of entities is similar to the definition of concept as defined by Xie et al. [29], where n-grams are weighted by *tf.idf* scores, except that we use noun phrases rather than any n-grams. We use mid-range document frequency instead of *idf* as in [5], where the entities occur in between 10% and 90% of the documents in the collection. We do not currently attempt coreference resolution for entities; recent work has investigated coreference resolution for multi-party dialogues [16, 8], but the challenge of resolution on such noisy data is highlighted by low accuracy (e.g. F-measure of 21.21) compared with using well-formed text (e.g. monologues).

Using a number of supervised classifiers trained on labelled decision sentences, action sentences etc. the sentences in the conversation were mapped to our ontology. The classifiers have been evaluated both on meeting and email data, the AMI [4] and BC3 [27] corpus respectively, as mentioned in [17] and found to perform well on both sets of data. The flexibility of our mapping approach is that it only relies on generic conversational features and can therefore be applied to a multi-modal conversation, for example a conversation that spans both an email thread and a meeting. We used a feature set related to generic conversational structure, which include sentence length, sentence position in the conversation and in the current turn, pause-style features, lexical cohesion, centroid scores, and features that measure how terms cluster between conversation participants and conversation turns. Despite using generic features the classifiers achieve similar results to [11, 21, 20, 22], that rely on meeting-specific or email-specific features (e.g., prosody for meetings). Details can be found in [2].

For AMI corpus, a particular conversation for a meeting consists of utterances that have the following format:

```
<Utterance rdf:about="#TS3012a.A.dialog-act.vkaraisk.14">
    <rdf:type rdf:resource="&owl;Thing"/>
    <hasSpeaker rdf:resource="#ProjectManager"/>
    <hasDAType rdf:resource="#NegativeSubjective"/>
    <begTime>21.66</begTime>
    <endTime>22.74</endTime>
</Utterance>
```

The above utterance is a negative subjective statement or a negative comment made by the ProjectManager at the meeting. The beginning time of utterance is used to temporally order the whole conversation and the unique identifier of the Utterance object is used to match the utterance with the actual sentence being said and thus any relevant entities.

[1]Our classifiers are designed for identifying five subclasses of the DA-type class, namely decisions, actions, problems, positive subjectives, and negative subjectives; but we could easily include additional classifiers to identify other types of dialog acts according to the information need.

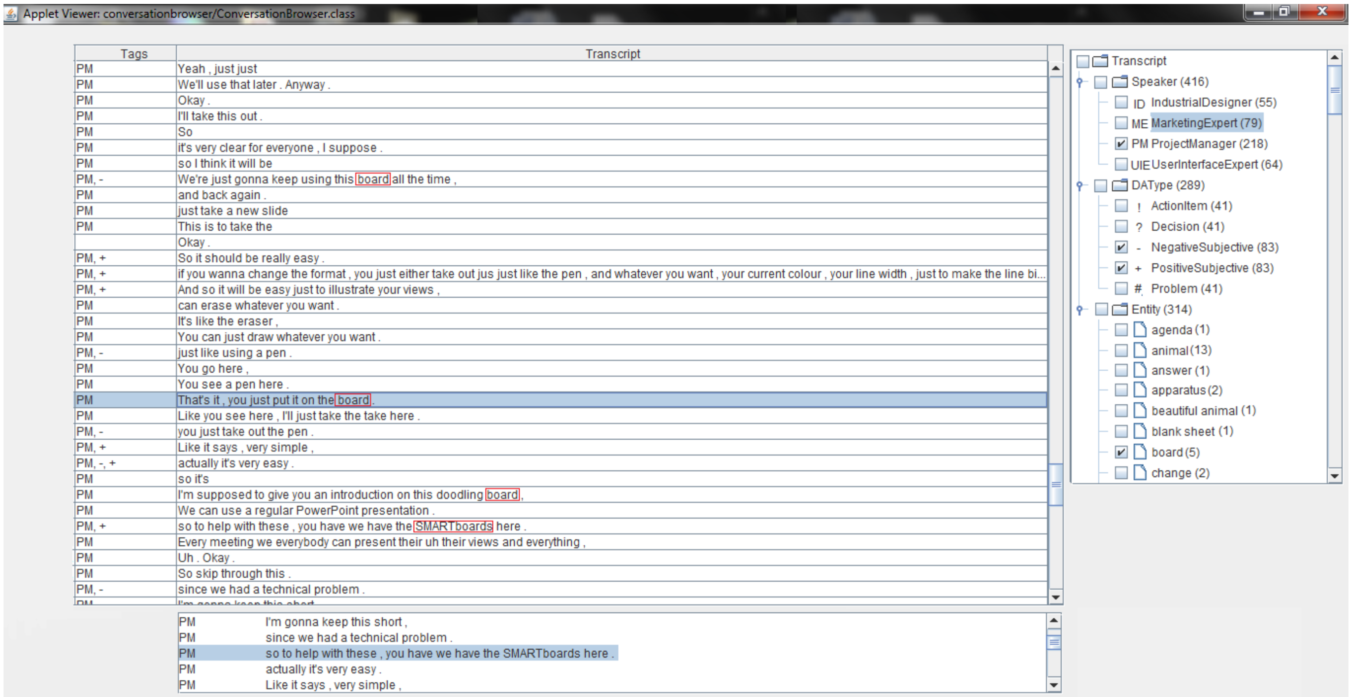

Figure 2. A refinement of the visual interface in Fig 1, intended for both browsing and summarizing conversations and consisting of three integrated views - the Ontology View (right), the Transcript View (middle) and the Summary View (bottom)

DISPLAY DESIGN

Once the ontology is populated with the participants, DA types and entities of a particular conversation, the transcript of the conversation is displayed ordered temporally. The design of the interface is intended to satisfy two key goals. The first goal is to support the exploration of the conversation through annotating the discourse with an ontology. This is achieved by allowing the users to select subclasses from the ontology that seem promising to fulfill their particular information needs and by allowing them to inspect the sentences that are associated to those subclasses in the context of the whole transcript. The second goal is to support the generation of focused summaries that cover only aspects of the conversations which are especially relevant to the users. This is achieved by allowing the users to select classes of sentences that they find particularly informative and that should be included in the summary (include verbatim for an extractive summary vs. include their content for an abstractive one).

In this section we discuss in more detail how the achievement of these two key goals is supported by our novel display, shown in Figure 2. Our visual interface consists of three integrated views, the Ontology View (right), the Transcript View (middle) and the Summary View (bottom). Contrast this with the simpler interface presented in [2] (see Fig 1). Our interface does not feature audio-video data streams in addition to transcripts as in Meeting Miner [1] or Ferret [28] because we have designed it to explore and summarize multi-modal conversations in general. The prototype was developed using Java Swing components and Jena, an open source Java framework that provides a programmatic environment for building semantic web applications.

The Ontology View

The ontology view provides a structured way for the users to explore all the relevant concepts in the conversation and their relations. It contains a tree hierarchy with core nodes Speaker, DAType (Dialogue Act Type), and Entity. Conceptually the top node in the ontology tree represents all the utterances or sentences in the conversation, while any other node represents a subset or subclass of those sentences that satisfies a particular property. For instance, the node ProjectManager (PM) represent all the sentences uttered by the PM, while the node ActionItem represents all the utterances that were classified as containing an action item. The Entities core node, on the other hand, does not represent all the noun phrases detected in the conversation but only the ones deemed important on the basis of their frequency and the ones that are associated with messages in the conversation.

As shown in Fig. 2, like in [2], the nodes each have a check box and a label. Additionally we have included a count within parentheses beside the labels. For leaf nodes, the count indicates how many sentences were mapped to this node and imply its relevance for the summary; for non-leaf nodes this is just the sum of all its descendant leaf node count; a count is an implication of the node's relevance. For Speaker subtree nodes the sets are mutually exclusive and these counts give a sense how dominant a role was in this particular meeting. For Entity and DAType subtrees, the sets can be overlapping and the counts at the non-leaf core nodes indicate the extent of overlap. Our initial prototype displays the entities in an alphabetical order but we could use the counts to order the entities on the ontology tree as an indicator to their relevance.

The Transcript View

The transcript view is designed to allow the users to inspect the whole conversation as well as the mapping of each sentence into the ontology. This view has two columns - Transcript and Tags. The Transcript column displays the whole conversation one sentence per row, while keywords and icons for the nodes in the ontology to which each sentence was mapped to are shown in the corresponding Tags column (to the left of the Transcript column), in case of selected nodes under the Speaker and DAType core nodes; or highlighted in the Transcript column, in case of nodes under Entity core node (refer to the Interaction Design Section for a user scenario). We have decided to display the entities highlighted in the transcript instead of mentioning them in the Tags column so that the users could inspect them in their actual context. Also, adding a number of long noun phrases to the Tags column would have widened that particular column space making it difficult for the users to inspect both the Tags and Transcript columns at the same time.

The Transcript View is scrollable both vertically and horizontally which can be used to inspect a sentence in its context i.e. its position in the conversation. A sentence may convey additional information in conjunction with its surrounding sentences. For example, when users inspect the sentence 'That's it, you just put it on the board.' mentioning the entity 'board' in its context, they may decide to include the entity 'pen' for further investigation since the 'it' in the sentence refers to the 'pen' that appeared in a preceding sentence.

The Summary View

The summary view is a text area where the candidate summary of the conversation appears for user assessment. The summary is based on sentences selected from the transcript using the criteria set from the ontology tree and is generated using either extraction or abstraction as in [18]. The summary view provides an easier way to assess the conversation overview on a particular information need without scrolling through the whole transcript. When the summary is extractive, to support the users in interpreting the summary in the context of the whole transcript, each sentence in the summary view is prefixed with a keyword indicating the speaker of the sentence. We are currently exploring how to provide a similar support for abstractive summaries.

INTERACTION DESIGN

The list of entities gives the users an idea of the conversation content without requiring them to browse the whole transcript. Furthermore, by selecting a few entities out of the list the users can satisfy particular information needs on the direction the conversation took regarding those particular entities. For instance, a user may be interested in all the comments made by the ProjectManager on the 'board' and whether these comments were positive or negative. To achieve this goal, the user would select the node 'board' under the Entity sub-tree, the node 'ProjectManager' under the Speaker core node and 'PositiveSubjective' and 'NegativeSubjective' nodes under the DA Type core node. As shown in figure 2, this would display the keywords 'PM', '+' and '-' in the Tags column of sentences that map to each of these nodes and would also highlight every occurrence of the word 'board', providing the user scope for closer inspection by scrolling through the transcript.

When the transcript view is generated for a conversation, the Tags column is initially empty and all the nodes on the ontology tree in the ontology view are shown and are de-selected. If for a particular conversation the ontology is too large, the users can expand/minimize nodes they are/are-not interested in, as in any standard outline based interface. Once the users select a node (or de-selects an already selected node) on the ontology tree, the keyword or icon associated with that node appears in (or disappears from) the Tags column of all the rows that contain sentences that can be mapped to that particular node. Once the users have selected the nodes of interest from the ontology tree, they can scroll through the transcript view and select sentences that appear to be promising for generating a focused summary. When they choose a sentence, all sentences that have the exact same tag set associated are detected and the summary view is updated with the summary generated based on these sentences. These sentences are included verbatim for an extractive summary, while an abstractive summary will summarize their content. The generated summary is also interactive, for the extractive case, clicking on a sentence in the summary view highlights it and also re-focuses the vertical scroll bar position on the transcript view to show the context of that particular sentence. For the abstractive case, we are still investigating ways to provide a similar functionality, which needs to take into account that there is no one-to-one mapping between the sentences in the summary and the ones in the transcript.

CONCLUSION AND FUTURE WORK

This paper presents a visual interface that not only allows users to explore a conversation using a mapping to an ontology, but also allows them to interactively generate focused summaries of human conversation. The classifiers for the mapping phase are not dependent on features specific to any particular mode of conversation which makes this approach extensible to multi-modal conversations. We are currently working to extend our interface to asynchronous conversations like emails and blogs, which entails the additional complication of having a non-linear conversational structure (i.e., a graph). In the future, we plan to investigate coreference resolution for the noun phrases in the conversation, this would also deal with synonymy of entities.

Our interface supports the generation of focused summaries by allowing the users to select classes of sentences that they find particularly informative and that should be included in the summary. If these sentences are included verbatim, we generate extractive summaries, whereas if the content of those sentences is extracted and aggregated, we generated abstract summaries. There is evidence that human abstractors at times use sentences from the source documents nearly verbatim in their own summaries, justifying this approach to some extent [14]. However, other studies also show that users usually prefer concise abstracts and find them more coherent[9, 15, 19]. In our interface, we can generate both types of summaries, so it represents an ideal environment in which to explore the pros and cons of these two methods and the possible benefits of their integration.

To assist users to decide on the informativeness of a set of sentences chosen according to a criteria set on the ontology tree, in the future we would also provide users with a task history. Depending on the size of the Entities subtree, it may be prohibitively time-consuming for the users to recreate a previously examined criterion by reselecting all of the relevant nodes since the userw would have to recall and find the exact nodes selected before and re-select them. A task history may record such promising criteria so that users can re-assess them later on using a single selection from this history view.

Our current prototype only shows a local view of the conversation, with fewer than fifty sentences. We plan to provide a second visualization that shows a global view of the entire conversation and possibly of the entire corpus of conversations (as in [13]). A possible approach could be to display a textual collage like tag clouds or word clouds in the overview window of the list of entities to give users a sense of the content of the conversation or corpus transcript without browsing. We could also show the representation of speaker participation information as a vertical scrollable timeline as in [28].

Finally, before engaging in a second redesign exercise, we plan to run a formative user evaluation of the interface using objective task scores.

REFERENCES

1. M.-M. Bouamrane and S. Laz. Navigating Multimodal Meeting Recordings with the Meeting Miner. In *Proc. of FQAS, Milan, Italy*, 2006.

2. G. Carenini and G. Murray. Visual Structured Summaries of Human Conversations. In *Proc. of IVITA 2010, HongKong, China*, pages 41–44, 2010.

3. G. Carenini, R. Ng, and X. Zhou. Summarizing Email Conversations with Clue Words. In *Proc. of ACM WWW 2007, Banff, Canada*, 2007.

4. J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The AMI Meeting Corpus: A Pre-Announcement. In *Proc. of MLMI 2005, Edinburgh, UK*, pages 28–39, 2005.

5. K. Church and W. Gale. Inverse Document Frequency IDF: A Measure of Deviation from Poisson. In *Proc. of the Third Workshop on Very Large Corpora*, pages 121–130, 1995.

6. R. Dachselt and M. Frisch. Mambo : A Facet-based Zoomable Music Browser. In *Proc. of MUM, Oulu, Finland*, 2007.

7. M. Galley. A Skip-Chain Conditional Random Field for Ranking Meeting Utterances by Importance. In *Proc. of EMNLP 2006, Sydney, Australia*, pages 364–372, 2006.

8. S. Gupta, J. Niekrasz, M. Purver, and D. Jurafsky. Resolving "*You*" in Multi-Party Dialog. In *Proc. of SIGdial 2007, Antwerp, Belgium*, 2007.

9. L. He, E. Sanocki, A. Gupta, and J. Grudin. Auto-summarization of audio-video presentations. In *Proc. of ACM MULTIMEDIA '99, Orlando, FL, USA*, pages 489–498, 1999.

10. W. C. Hill, J. D. Hollan, D. Wroblewski, and T. McCandless. Edit Wear and Read Wear. In *Proc. of SIGCHI, Monterey, California, USA*, 1992.

11. P.-Y. Hsueh, J. Kilgour, J. Carletta, J. Moore, and S. Renals. Automatic Decision Detection in Meeting Speech. In *Proc. of MLMI 2007, Brno, Czech Republic*, 2007.

12. P.-Y. Hsueh and J. D. Moore. Improving Meeting Summarization by Focusing on User Needs : A Task-Oriented Evaluation. In *Proc. of IUI, Florida, USA*, 2009.

13. Indratmo, J. Vassileva, and C. Gutwin. Exploring Blog Archives with Interactive Visualization. In *Proc. of AVI, Naples, Italy*, 2008.

14. J. Kupiec, J. Pederson, and F. Chen. A Trainable Document Summarizer. In *Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA*, pages 68–73, 1995.

15. K. McKeown, J. Hirschberg, M. Galley, and S. Maskey. From Text to Speech Summarization. In *Proc. of ICASSP 2005, Philadelphia, USA*, pages 997–1000, 2005.

16. C. Muller. Resolving *It*, *This* and *That* in Unrestricted Multi-Party Dialog. In *Proc. of ACL 2007, Prague, Czech Republic*, 2007.

17. G. Murray and G. Carenini. Interpretation and Transformation for Abstracting Conversations. In *North American ACL, Los Angeles, CA, USA*, 2010.

18. G. Murray, G. Carenini, and R. Ng. Generating Abstracts of Meeting Conversations: A User Study. In *Proc. of INLG, Dublin, Ireland*, 2010.

19. G. Murray, T. Kleinbauer, P. Poller, S. Renals, T. Becker, and J. Kilgour. Extrinsic Summarization Evaluation: A Decision Audit Task. In *Proc. of MLMI 2008, Utrecht, the Netherlands*, 2008.

20. G. Murray and S. Renals. Detecting Action Items in Meetings. In *Proc. of MLMI 2008, Utrecht, the Netherlands*, 2008.

21. M. Purver, J. Dowding, J. Niekrasz, P. Ehlen, and S. Noorbaloochi. Detecting and Summarizing Action Items in Multi-Party Dialogue. In *Proc. of the 9th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium*, 2007.

22. S. Raaijmakers, K. Truong, and T. Wilson. Multimodal Subjectivity Analysis of Multiparty Conversation. In *Proc. of EMNLP 2008, Honolulu, HI, USA*, 2008.

23. O. Rambow, L. Shrestha, J. Chen, and C. Lauridsen. Summarizing Email Threads. In *Proc. of HLT-NAACL 2004, Boston, USA*, 2004.

24. M. Sedlmair, C. Bernhold, D. Herrscher, S. Boring, and A. Butz. MostVis: An Interactive Visualizing Supporting Automotive Engineers in MOST Catalog Exploration. In *Proc. of InfoVis09, New Jersey, USA*, 2009.

25. G. Smith, M. Czerwinski, B. Meyers, D. Robbins, G. Robertson, and D. S. Tan. FacetMap: A Scalable Search and Browse Visualization. In *Proc. of InfoVis, Baltimore, Maryland, USA*, 2006.

26. G. Tur, A. Stolcke, L. Voss, S. Peters, D. Hakkani-Tr, J. Dowding, B. Favre, R. Fernndez, M. Frampton, M. Frandsen, C. Frederickson, M. Graciarena, D. Kintzing, K. Leveque, S. Mason, J. Niekrasz, M. Purver, K. Riedhammer, E. Shriberg, J. Tien, D. Vergyri, and F. Yang. The CALO Meeting Assistant System. In *IEEE Transaction on Audio, Speech and Language Processing*, 2010.

27. J. Ulrich, G. Murray, and G. Carenini. A Publicly Available Annotated Corpus for Supervised Email Summarization. In *Proc. of AAAI EMAIL-2008 Workshop, Chicago, USA*, 2008.

28. P. Wellner, M. Flynn, and M. Guillemot. Browsing Recorded Meetings with Ferret. In *Proc. of MLMI 2004, Martigny, Switzerland*, pages 12–21, 2004.

29. S. Xie, B. Favre, D. Hakkani-Tür, and Y. Liu. Leveraging Sentence Weights in a Concept-based Optimization Framework for Extractive Meeting Summarization. In *Proc. of Interspeech 2009, Brighton, England*, 2009.

30. K.-P. Yee, K. Swaringen, K. LI, and M. Hearst. Faceted Metadata for Image Search and Browsing. In *Proc. of CHI, Boston, USA*, 2003.

31. L. Zhou and E. Hovy. Digesting Virtual “Geek” Culture: The Summarization of Technical Internet Relay Chats. In *Proc. of ACL 2005, Ann Arbor, MI, USA*, 2005.

32. X. Zhu and G. Penn. Summarization of Spontaneous Conversations. In *Proc. of Interspeech 2006, Pittsburgh, USA*, pages 1531–1534, 2006.