A Visual Interface for Browsing and Summarizing Conversations

Shama Rashid[*]
Department of Computer Science
University of British Columbia

Abstract

In this paper I present a visual interface for generating focused summaries for human conversations. My interface provides the user a way to quickly explore a conversation to get an overview of its content and to interactively identify informative sentences as potential components of the summary. The candidate sentences are selected based on visual cues provided by mapping to an ontology containing nodes for the speakers of the conversation, sentence-level features like Dialog Acts (in short DAs, like decision, action, problem etc.) properties, and entities referred to in the conversation. In this course project, I have improved the design of the interface to enhance the usability based on information visualization principles; specifically addressing the issues of a better visual representation for the ontology information and easier exploration of the conversation transcript.

Terminology: Through out the scope of this project the terms 'utterance' and 'sentence' will be used interchangeably to indicate a segment of the conversation that can express a meaning independent of other segments. The people involved in the conversation will be referred to as 'participants' or 'speakers'. 'Entities' are noun phrases that appear in the conversation with a mid range document frequency. 'Dialog Acts' are sentence level features that specify whether a sentence expresses positive or negative opinions, a decision, an action or a problem.

1 Introduction

In our daily lives, we have conversations with people in many different modalities - emails, meetings, telephone, video-conferencing, instant messaging, blogs, forums etc. The Web has significantly increased the volume and the complexity of the conversational data generated through our day to day communication and has provided us with a rich source of readily available public discourse data (which has proved to be a challenge from NLP point of view due to its often highly informal nature that render text processing using language syntax, lexicographic and semantic rules inept). The ontology containing nodes for speakers, dialogue acts, and a list of entities referred to in the conversation is derived using classifiers and parsers based on generic features. So, the interface could be used to explore any conversational data provided (offline) in a text format.

With the high volume of conversational data being generated every day, summarization can be beneficial by creating concise overviews that aid quick access to the data. Extractive summary generated by selecting and concatenating the most informative sentences is the dominant approach in Natural Language Processing (NLP), since it can be considered a simple binary classification task of whether to include a particular sentence in the summary or not. However, the resultant summary is often unsatisfactory for users due to lack of coherency. The alternative approach is abstractive summary, which is generated by extracting and aggregating information from the conversation. This approach requires a natural language generation component and is preferred by users for coherency but the resultant summary sentences are often too generic and lack the details of the original conversation sentences. The objective of my interface is to provide a Visual Structured Summary that gives the user more control over choosing the topics she wants to appear in the concise resultant overview generated through interactive exploration, thus generating a focused summary.

I've been working mainly on the AMI meeting corpus [5]. The meeting conversations in the AMI corpus are structured as a series of 4 meetings the a-kickoff, b-conceptual design, c-detailed design, and d-evaluation meetings imitating product design cycle. In addition, the meetings have a fixed size group of participants playing very specific roles (like the project manager, the marketing expert, the interface designer and the user interface expert). The rigid format these meetings follow is not reflective of real conversational data but I have chosen to work on it since it is one of the most used and publicly available corpus in NLP research with different annotational data. Currently the interface can be used to analyze a single transcript, but in the future I'll extend it to analyze the whole series of transcripts together. I am currently extending the interface to other modes of conversation like emails, blogs, chats etc. and have also been working with the BC3 email corpus [18] and IBM technical blogs crawled off the web.

An example for mapping a sentence to the ontology can be found in Figure 1.

A: Let's go with a simple chip.
Speaker: A, who is the Project Manager
Entities: simple chip (only one for this example)
Dialog Acts: classified as decision and positive-subj

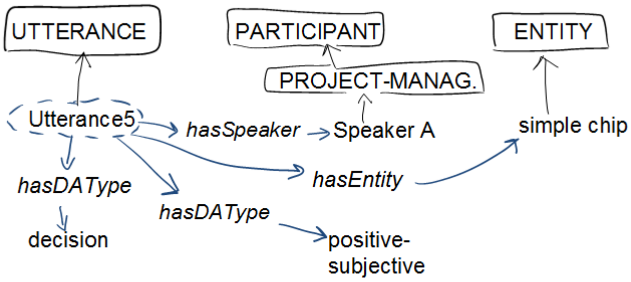


Figure 1: Example of mapping a sentence to an ontology

Each conversation thus can be considered as a dataset of sentence items. The attributes of each of these items would be binary variables (possible values 'yes' and 'no') indicating whether a particular sentence can be mapped to a particular concept on the ontology (DA, speaker and entity). Additionally, the time of utterance or the sequential order of a sentence in the conversation can be considered an ordinal value attribute based on which the dataset is sorted before display. For the above mentioned example item, the attribute 'ProjectManager' (say) would have a value 'yes' but the other speaker attribute columns would contain a 'no'.

[*] e-mail: shama@cs.ubc.ca

2 Related Work

(I am copying the portion on text or document browsing and summarizing in this section from [14].)

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 [24] 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 [24], 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 [9]. 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 [9] 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 [9] also considers features that are specific to conversations about designing a new product (see AMI corpus [5]), 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 [22] 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 [3] 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 [3] 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 [16] 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 [17] 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 [10], 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 [8], a means of graphically portraying the document's 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 [15] 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.

(The following portion of the section is literature review relevant to the interface redesign that I have implemented for the scope of this course project.)

The distribution of words or phrases in a text document has been explored using different techniques, the central theme of all of them being use of size to encode the frequency of occurrence of a word. Wordles [19] use tag-cloud-like displays that give careful attention to typography, color and composition. The font size of the words are linearly related to frequency of word in the text. Wordle makes compact use of space but cannot be used to compare contents of documents. TextPool [1] text collages adjust dynamically in response to user interaction and to changes in stream content like wire feeds. The salient terms from the stream are connected to their co-occurring terms by links whose lengths are scaled by the inverse of their co-occurrence, so that terms that are closely related and co-occur often are close to one another in the graph. TextPool uses brighter terms for recent topics and less bright ones for older topics, an idea which is similar to my design for the marker bar. TextArc [13] shows the distribution of words in a document using collocation and displays the entire text in one view. iBlogVis presented in [10] also uses a tag cloud representation of the heavily discussed topics. In contrast to most techniques, the Word Tree [11] uses the square root of the frequency of the word so that the area of the word is very roughly proportional to the frequency (except the variation created by word length).

The collocation of words based on lexical or syntactic relations is another aspect of text content analysis. The Word Tree provides a way to explore the phrases in a text in an interactive way. However, it does not provide any sort of overview of the text nor does it present an initial search term for viewers to start from i.e. there is no natural entry point to the text. The Phrase Net [7] interface allows users to input different regular expressions to analyze the text from different perspectives. In my work, the ontology concepts act as natural entry points, but I am not handling sentence level analysis of word relations. A common request from the Word Tree users was for the ability to click on an item in the visualization and see the places in the raw transcript where the item appears. This is the same as the highlighting of the entities in the transcript in my interface. Arc diagrams [20] are capable of representing complex patterns of repetition in string data with many different repeated sub-sequences and multiple scales of repetition. For each consecutive matching pair on the string, the corresponding intervals are connected with a translucent thick semi-circular arc. The height of the arc is thus proportional to the distance between the two sub-strings. The translucency allows to reveal highly repetitive structure. A similar effect is achieved through the distribution of markers along my marker bars.

The participatory information presented in the Ferret meeting browser [22] is similar to the distribution of markers along the Speaker marker bar in my interace. [21] uses a grid structure to show how the metadata of the text corresponds to a row in the text. Each paragraph in the document overview is lined up with a column in the grid and metadata concepts correspond to the rows. To show that a tag has been applied to a particular paragraph the cell where the appropriate column and row meet is shaded. The color of the cell is determined by the value of the applicability attribute. When a mouse hovers over a grid cell all relevant tags on the metadata tree are highlighted. The metadata is another example of facets of the conversation text and the grid is an alternative to our marker bar in showing distribution of concepts. The count beside the ontology concept labels, the marker bar, the sentence tags in the summary view are all examples of scented widgets [23] that provide cues for more informative data.

3 Solution

The design of my current interface for browsing and summarizing conversations has evolved through numerous iterations of applying HCI, NLP and Information Visualization principles. I had started with the interface presented in the subsection 'Stage 1 Design' and had modified it to the fully functional prototype described in 'Stage 2 Design' subsection. Within the scope of this course project, I have further modified the interface based on InfoVis visual encoding and interaction principles. The latest design is presented in subsection 'Stage 3 Design'.

3.1 Stage 1 Design

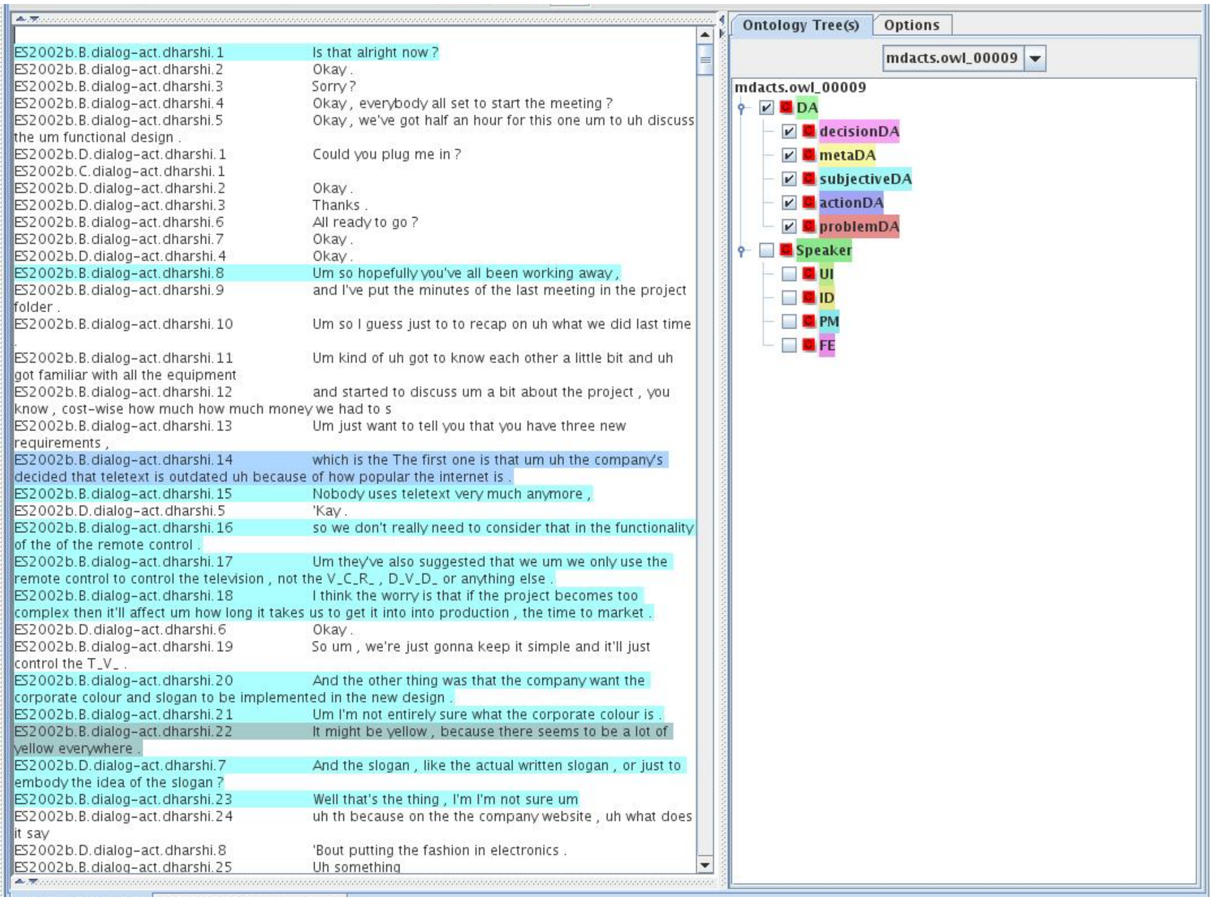
The details on an initial interface to aid creating visual structured summaries of conversations can be found in [4]. This interface relied on mapping the utterances of the conversation into an ontology that then could be used to search the conversation according to the annotation. The ontology in this interface contained nodes for participants of the conversation and properties of the utterance/sentence such as whether it was expressing a decision, a problem etc i.e. the DAs. The initial design contained two panels - a panel on the left for displaying the transcript and another on the right to show the ontology (see Fig 2). The transcript panel showed the sentences of the conversation one per row, ordered temporally, prefixed by the sentence identifiers. The ontology was presented in a tree-structure allowing multiple node selection using checkboxes juxtaposed to the node labels. Each node on the ontology was assigned a distinct color. 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 could be mapped in the selected nodes would be highlighted using the color assigned to the selected node. If a sentence could be mapped into multiple selected nodes, the highlight color would be a combination of the colors used for the original nodes.

3.2 Stage 2 Design

The stage 2 design (see Fig 3) addressed several limitations of the initial prototype presented in the previous subsection. It consisted of three integrated views - the Ontology View, the Transcript View and the Summary View. The details on the display and interaction design of this stage can be found in [14]. I am stating the solutions I provided at this stage to a few problems with the intial design from information visualization point of view below.

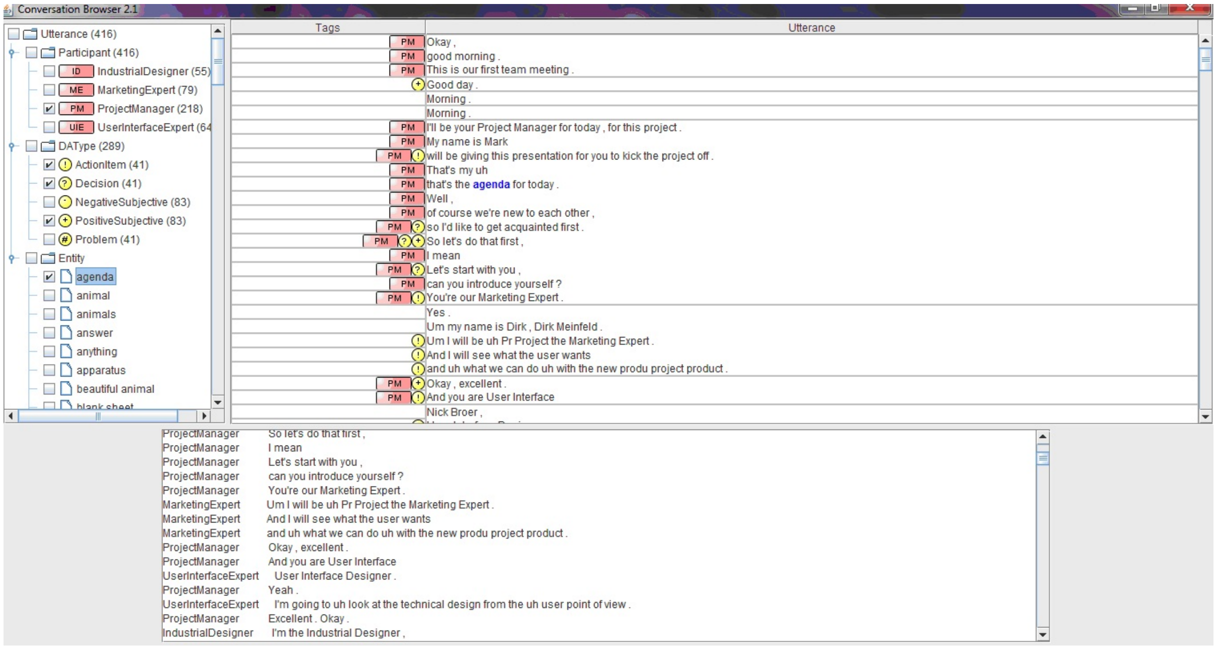
The accuracy of visual channels can be measured by how many level of information they can convey, whether they can be interpreted separately or are automatically merged. One of the major issues with the initial interface was the use of highly saturated colors to highlight sentences. In addition, the solution was non-scalable, as a sentence could be tagged with multiple labels and using a combination of the colors of the original nodes becomes perceptually indistinguishable very quickly. If we consider a sentence with just one Speaker tag and one DA type tag there is 4x5=20 possible colors that could be applicable, and human beings can hardly distinguish more than a dozen colors in such a situation. The actual situation is far worse as a sentence can be classified as multiple DA types simultaneously. Another conceptual problem with this solution was using colors to show two different categorical attributes (Speaker, DA type) when the color map for categorical + categorical attributes are perceptually inseparable. The solution I came up with in this stage was, instead of using color, I had added a column (named Tags) to the left of the transcript utterances, in which the (selected) mapping to the (knowledge concepts within) ontology of the corresponding utterance can be displayed.

I had also included entities mentioned in the conversation to the ontology representation. Searching the conversation using a particular keyword can only be used when user has previous knowledge about the content and want additional information on a particular entity. Representing a list of entities enables the user to perform a more refined search and browsing of the conversation. In addition, the entities also provides the user with a quick overview of the con-



```
ES2002b.B.dialog-act.dharshi.1        Is that alright now ?
ES2002b.B.dialog-act.dharshi.2        Okay .
ES2002b.B.dialog-act.dharshi.3        Sorry ?
ES2002b.B.dialog-act.dharshi.4        Okay , everybody all set to start the meeting ?
ES2002b.B.dialog-act.dharshi.5        Okay , we've got half an hour for this one um to uh discuss
the um functional design .
ES2002b.D.dialog-act.dharshi.1        Could you plug me in ?
ES2002b.C.dialog-act.dharshi.1
ES2002b.D.dialog-act.dharshi.2        Okay .
ES2002b.D.dialog-act.dharshi.3        Thanks .
ES2002b.B.dialog-act.dharshi.6        All ready to go ?
ES2002b.B.dialog-act.dharshi.7        Okay .
ES2002b.D.dialog-act.dharshi.4        Okay .
ES2002b.B.dialog-act.dharshi.8        Um so hopefully you've all been working away ,
ES2002b.B.dialog-act.dharshi.9        and I've put the minutes of the last meeting in the project
folder .
ES2002b.B.dialog-act.dharshi.10       Um so I guess just to to recap on uh what we did last time
.
ES2002b.B.dialog-act.dharshi.11       Um kind of uh got to know each other a little bit and uh
got familiar with all the equipment
ES2002b.B.dialog-act.dharshi.12       and started to discuss um a bit about the project , you
know , cost-wise how much how much money we had to s
ES2002b.B.dialog-act.dharshi.13       Um just want to tell you that you have three new
requirements ,
ES2002b.B.dialog-act.dharshi.14       which is the The first one is that um uh the company's
decided that teletext is outdated uh because of how popular the internet is .
ES2002b.B.dialog-act.dharshi.15       Nobody uses teletext very much anymore ,
ES2002b.D.dialog-act.dharshi.5        'Kay .
ES2002b.B.dialog-act.dharshi.16       so we don't really need to consider that in the functionality
of the of the remote control .
ES2002b.B.dialog-act.dharshi.17       Um they've also suggested that we um we only use the
remote control to control the television , not the V_C_R_ , D_V_D_ or anything else .
ES2002b.B.dialog-act.dharshi.18       I think the worry is that if the project becomes too
complex then it'll affect um how long it takes us to get it into into production , the time to market .
ES2002b.D.dialog-act.dharshi.6        Okay .
ES2002b.B.dialog-act.dharshi.19       So um , we're just gonna keep it simple and it'll just
control the T_V_ .
ES2002b.B.dialog-act.dharshi.20       And the other thing was that the company want the
corporate colour and slogan to be implemented in the new design .
ES2002b.B.dialog-act.dharshi.21       Um I'm not entirely sure what the corporate colour is .
ES2002b.B.dialog-act.dharshi.22       It might be yellow , because there seems to be a lot of
yellow everywhere .
ES2002b.D.dialog-act.dharshi.7        And the slogan , like the actual written slogan , or just to
embody the idea of the slogan ?
ES2002b.B.dialog-act.dharshi.23       Well that's the thing , I'm I'm not sure um
ES2002b.B.dialog-act.dharshi.24       uh th because on the the company website , uh what does
it say
ES2002b.D.dialog-act.dharshi.8        'Bout putting the fashion in electronics .
ES2002b.B.dialog-act.dharshi.25       Uh something
```

Figure 2: Stage 1 design



Tags	Utterance
PM	Okay ,
PM	good morning .
PM	This is our first team meeting .
+	Good day .
	Morning .
	Morning .
PM	I'll be your Project Manager for today , for this project .
PM	My name is Mark
PM !	will be giving this presentation for you to kick the project off .
PM	That's my uh
PM	that's the **agenda** for today .
PM	Well ,
PM	of course we're new to each other ,
PM ?	so I'd like to get acquainted first .
PM ? +	So let's do that first ,
PM	I mean
PM ?	Let's start with you ,
PM	can you introduce yourself ?
PM !	You're our Marketing Expert .
	Yes .
	Um my name is Dirk , Dirk Meinfeld .
!	Um I will be uh Pr Project the Marketing Expert .
!	And I will see what the user wants
!	and uh what we can do uh with the new produ project product .
PM +	Okay , excellent .
PM !	And you are User Interface
	Nick Broer ,

```
ProjectManager        So let's do that first ,
ProjectManager        I mean
ProjectManager        Let's start with you ,
ProjectManager        can you introduce yourself ?
ProjectManager        You're our Marketing Expert .
MarketingExpert       Um I will be uh Pr Project the Marketing Expert .
MarketingExpert       And I will see what the user wants
MarketingExpert       and uh what we can do uh with the new produ project product .
ProjectManager        Okay , excellent .
ProjectManager        And you are User Interface
UserInterfaceExpert    User Interface Designer .
ProjectManager        Yeah .
UserInterfaceExpert    I'm going to uh look at the technical design from the uh user point of view .
ProjectManager        Excellent . Okay .
IndustrialDesigner     I'm the Industrial Designer ,
```

Figure 3: Stage 2 design with 3 integrated view - the Ontology View (left), the Transcript View (right), the Summary View (bottom)

tent of the whole conversation without browsing the transcript. I had also included count within parentheses beside the labels on the ontology tree as information scent. For leaf nodes, the count indicates how many sentences were mapped to this node and imply its relevance for the summary.

To take advantage of visual popout, I had decided to use icons associated with the ontology concepts instead of text labels as representation of tags in the Tags column. I had used pink rectangles for the Speaker type tags and yellow circles for the DA type tags to make the ontology core concepts distinguishable using color and shape channel. A word can have different meanings in different contexts, that's why instead of showing the entity tags in the Tags column, I was highlighting them in bold blue font within the transcript sentences.

I had included gridlines in light gray in the transcript to make the separation of the temporally ordered sentences (one per row) apparent. In the Utterance column, the sentences spoken by a speaker subsequently without intervention from another speaker (know as turns in NLP terms) were grouped using containment within a larger grid box.

The summary view was a new addition at this stage that worked as a filtered view of the sentences that can be mapped to the nodes currently selected on the ontology. Although these sentences can be inspected in the context of the transcript, they may be highly dispersed and the length of the conversation may make it impossible to display them in a satisfactory way within the currently viewable portion of the transcript. The summary view is linked to the transcript view as well. Clicking a sentence in the summary view highlights the corresponding sentence in the transcript view (along with the two preceding sentences and the two subsequent sentences to make the highlight easier to spot) and also adjusts the viewport on the Transcript view to show the highlighting by auto-scrolling.

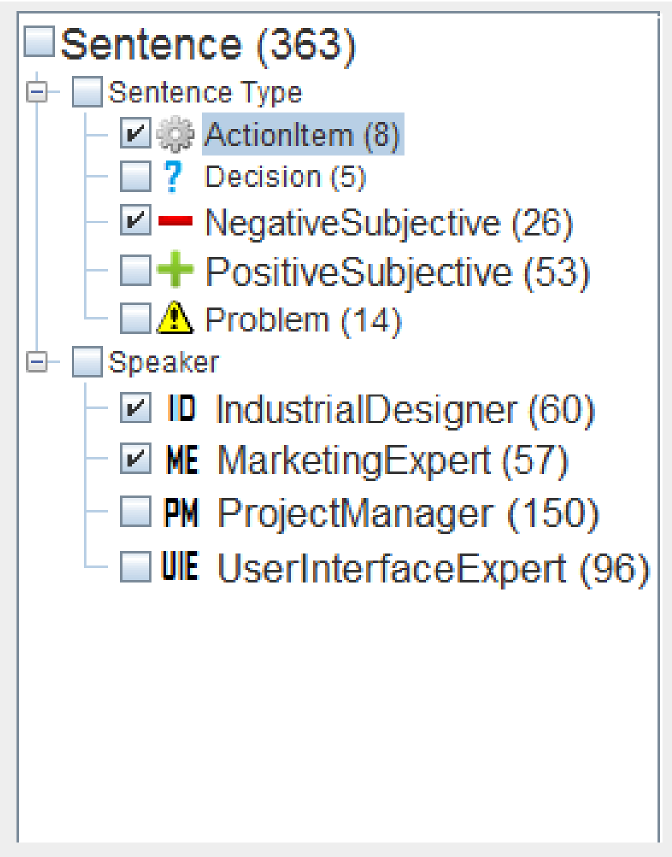
The summary view is also an important addition to the interface since it decouples the task of identifying informative sentences and the task of generating a focused summary for those selected sentences. This makes it possible to choose either an extractive or an abstractive [12] approach for the generation task. A user after she has 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 her current information needs). For instance, she 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*).

3.3 Stage 3 Design

Most of the basic concepts from the Stage 2 design have been preserved in this stage. However, I have modified the interface to be more visually representative of the data using InfoVis principles (see Fig 4). I have improved the data encoding by making efficient use of more visual channels and have provided new ways of interaction. I have also changed the layout of the multiple views to make better use of screen real estate. The details of my recent modifications and the rationale behind them are enumerated in the following subsections.

3.3.1 Data Encoding

I am now using the shape, color, size, and position visual channels for encoding of data attributes as opposed to minimally using shape and color to code the core concepts (the pink rectangle and yellow circle shaped icons) in the previous design stage. The first two channels, shape and color, can be used efficiently for a relatively limited number of levels before they start loosing distinguishability. On the other hand, theoretically, size and position can be used to encode a much larger number of levels but they take up screen space which is a limited resource. Although spatial position is the most efficient channel to encode all types of data (quantitative, ordinal or categorical), text itself has an inherent linear order and displaying text in a legible manner puts it up as a contender for screen space. The nature of each of these data channels and their efficiency measured in terms of accuracy, discriminability, separability, and the ability to provide visual popout have been taken into account while deciding the encoding of my data attributes (speaker, DA type, entity etc.).



□ Sentence (363)
- □ Sentence Type
  - ☑ ActionItem (8)
  - □ ? Decision (5)
  - ☑ − NegativeSubjective (26)
  - □ + PositiveSubjective (53)
  - □ ⚠ Problem (14)
- □ Speaker
  - ☑ ID IndustrialDesigner (60)
  - ☑ ME MarketingExpert (57)
  - □ PM ProjectManager (150)
  - □ UIE UserInterfaceExpert (96)

Figure 5: Ontology View concept encoding using size, shape and color channels

The count within parenthesis beside the node labels on the ontology tree act as information scents; a higher frequency indicating repeated occurrence of that particular type within the items of the dataset. For a better visual representation of this information scent, in this stage, I have scaled the font size of the labels (see Fig 5) i.e. now I am using a larger font for nodes with larger counts to make them stand out more. However, the frequency distribution for the three core nodes DA type, Speaker and Entity are not the same. Entities in a conversation rarely appear more than 5 or 6 times. On the other hand, since each of the sentences can be mapped to a speaker, for a sizeable conversation (around 1000 sentences) each of the speaker nodes would have hundreds of sentences mapped to it. The DA type leaf nodes have a distribution similar to speakers. To keep the labels legible I had to impose a minimum and maximum font size for the scaling. Given these boundary conditions, using a linear function to scale the fonts for the leaf nodes was not possible as it would assign the maximum font size for most of the nodes under the Speaker and DA type core nodes. Using a logarithmic function would not provide a satisfactory solution since it would not reflect the variation in the frequency of the entities. As a solution to this completely different distribution of nodes under the core nodes, I am displaying the entities in a tag cloud format (see Fig 6) separate from the Ontology View. This has allowed me to use two different scaling functions for the labels in the two different views; a logarithmic scaling function for the Ontology View containing nodes for the DA type and speaker concepts and a linear scaling function for the Entity View representing the entity concepts.

The Entity View is a textual collage of the list of entities referred to in the conversation represented in a tag cloud format, in a rectangular tag arrangement. The entities are listed in a sequential line-by-line layout. It is possible to select multiple entities simultaneously from the list and the color channel is used to differentiate the set of entities that are selected from the unselected ones. I am using black for the unselected entities, blue for the selected ones and white on mouse hover over an entity tag. To make it clear to the user that there is no correlation between the sizes of the labels in the two views (Entity and Ontology) since the scaling function are different I have also set the background color of the Entity View different from the Ontology View. Since I am using color as one of the important visual encoding channels for the DA type concepts in the Ontology View (discussed below) I have kept the background color of the Ontology View white (see Fig 5 and Fig 6).

The DA types in my interface is a fixed size set (size is 5). Although this set could be extended if I used other supervised classifiers trained on annotated data, for the extant of my project the list is unchangeable, making it possible to use shape and color channels. In the stage 2 design I was using pink rectangles with abbreviation of the speaker name as icons for the Speaker nodes, and yellow circles with characters like '+', '-' etc. to indicate PositiveSubjectivism, NegativeSubjectivism etc. for the DA type nodes. There is a strong domain convention to use red for negative sentiments and green for positive sentiments, which I have followed for the redesigned icons and I have used luminance to counter the limitation of using red-green hues for color-deficient users. I have made the other DA type icons of distinctly different colors. I have also used the shape channel for the DA type concepts. So, instead of using a circle containing a '+' or '-' sign I am now using a green '+' or a red '-' shaped icon. For the other DA type nodes, I have used the shape of the most common icon found when I googled using the keywords (see Fig 5).

Although the speakers in the AMI meeting conversations are fixed, this is not the case with the email and blog discussions where the group of participants is of variable length. That is why shape and color are not scalable solutions for the speaker nodes. Also, there is no intuitive mapping from a person's name to shape or color and users of the interface are interested in knowing exactly which participant is saying what rather than just identifying a change of speaker. As such, I have decided to use the abbreviations of the speaker names as representative glyphs for the concepts (see Fig 5).

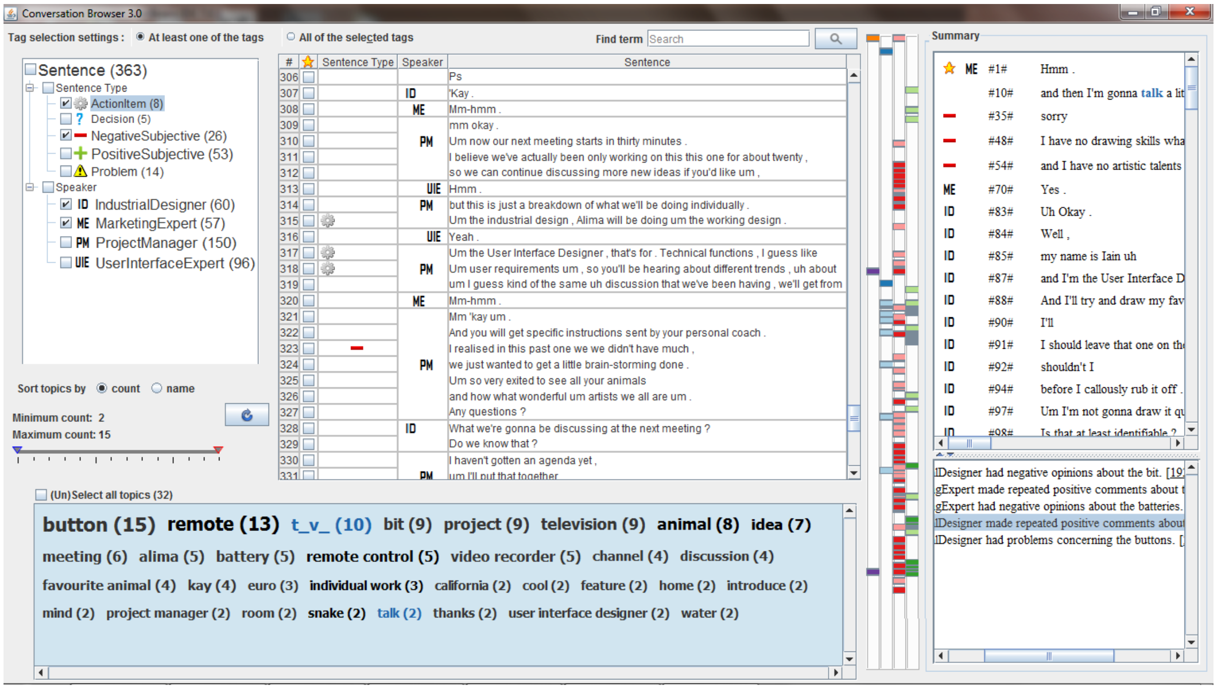
In the transcript view, I have split the column 'Tags' encasing both DA type and speaker to two separate columns, the 'Sentence Type' column and the 'Speaker' column (see Fig 7). This has enabled me to incorporate the speaker as a turn parameter which has two-fold advantages. Firstly, the icon for a particular speaker now appears only for the middle sentence of a turn, this reduces the time the user would have spent in verifying who is the speaker if it appeared for every line of text. Secondly, although speaker is still a filtering criteria selectable from the Ontology View, I am displaying the speaker icons all the time in the Transcript View to help users maintain orientation. Separating the columns reinforce the behavioural differences between the DA type and speaker concepts.

I am also redundantly coding these attributes using spatial positioning in the Trancript View. In the 'Sentence Type' and the 'Speaker' columns, icons for specific leaf nodes of these categories appear in specific locations. This solution for the 'Speaker' tag is applicable for the AMI meeting dataset only and cannot be extended to the email and blog conversations due to their varying list of participants (the number of participants in blog conversation can even reach upto hundreds).

In the stage 2 design, the extractive summary sentences in the Summary View were prefixed by the speaker of the sentence. This lacked the information why a sentence was selected to appear in the summary in the first place (unless the user was only exploring the Speaker subtree). In the new design, I have prefixed the tags for the selected ontology concepts that can be applied to the sentence to make the summary view self-contained without looking up the transcript. Also, I have add the line number for the sentences in the conversation in the transcript view as well as in the summary view. This provides the user an idea whether the sentences being selected are concentrated at a portion of the conversation indicating a possible topic shift. Providing these types of information scent is even more important for abstractive summary approach since for this approach the sentences are aggregations of a set of transcript sentences and without the cues the user has to rely completely on the quality of the abstractive summary generation component, which like any other machine learning approach, has a degree of uncertainty involved. For abstractive summary sentences, I have provided a list of line numbers for the component transcript sentences after each summary sentence (see Fig 8).

3.3.2 Interaction Design

Despite the visual cues provided as tags for the transcript sentences and the link between the summary view and the transcript view in the stage 2 design, the user still had to spend considerable time scrolling through the transcript since normally inspection of the context sentences is mandatory to fully understand the relevance of the tagged sentence and sometimes the tagged sentences are widely spread out through out the conversation length. I have added marker bars (see Fig 9) as a new mechanism for interacting with the transcript view to reduce this scroll time. I have decided to use 4 parallel marker bars instead of just one to reduce visual clutter. Now, there is one marker bar dedicated to each of the ontology core concepts (speaker, DA type and entity) and along each of these bars the most recently added tag of the type is shown in a highly saturated color while the older tags are shown in less saturated shades of the same color (blue for entity, red for participant and green for DA type). I have chosen the two shade encoding assuming that users would be adding a single concept (or a relatively small number of concepts) at a time as a filtering criteria and would want to inspect the effect of latest addition further before proceeding to adjusting the filtering criteria to satify her information need. There is a fourth marker bar along which (orange colored) markers appears for the sentences the user identifies as important (despite not being tagged). This allows the user to easily find those sentences later to re-inspect them. The mapping from the Summary View to the Transcript View (for abstractive summary) is also shown as (purple colord) markers along this bar. Although the markers on the different bars do not interact with each other, the bars are placed very close to each other; so, I have used a colorbrewer qualitative palette (colorbrewer2.org) to choose the colors of the markers. I have also set the background of the marker bars to a very light shade of gray so that the effect of the color coding is not interfered with. Due to scalability issues, only the core level concepts could be color coded. To distinguish leaf level concepts I am using the tooltip text of the markers. So, whenever the user hovers the mouse pointer over a particular marker, the corresponding concept is shown as a tooltip text. This can be considered as a rudimentary details on demand interaction where as the distribution of markers along the bars act as an overview of the document from the perspective of the concept's occurence along



Conversation Browser 3.0

Tag selection settings : ● At least one of the tags ○ All of the selected tags

Find term Search

□ Sentence (363)
- □ Sentence Type
  - ☑ ⚙ ActionItem (8)
  - □ ? Decision (5)
  - ☑ — NegativeSubjective (26)
  - □ + PositiveSubjective (53)
  - □ ⚠ Problem (14)
- □ Speaker
  - ☑ ID IndustrialDesigner (60)
  - ☑ ME MarketingExpert (57)
  - □ PM ProjectManager (150)
  - □ UIE UserInterfaceExpert (96)

Sort topics by ● count ○ name

Minimum count: 2
Maximum count: 15

#	☆	Sentence Type	Speaker	Sentence
306	□			Ps
307	□		ID	'Kay .
308	□		ME	Mm-hmm .
309	□		PM	mm okay .
310	□			Um now our next meeting starts in thirty minutes .
311	□			I believe we've actually been only working on this this one for about twenty ,
312	□			so we can continue discussing more new ideas if you'd like um ,
313	□		UIE	Hmm .
314	□		PM	but this is just a breakdown of what we'll be doing individually .
315	□	⚙		Um the industrial design , Alima will be doing um the working design .
316	□		UIE	Yeah .
317	□	⚙	PM	Um the User Interface Designer , that's for . Technical functions , I guess like
318	□	⚙		Um user requirements um , so you'll be hearing about different trends , uh about
319	□			um I guess kind of the same uh discussion that we've been having , we'll get from
320	□		ME	Mm-hmm .
321	□		PM	Mm 'kay um .
322	□			And you will get specific instructions sent by your personal coach .
323	□	—		I realised in this past one we we didn't have much ,
324	□			we just wanted to get a little brain-storming done .
325	□			Um so very exited to see all your animals
326	□			and how what wonderful um artists we all are um .
327	□			Any questions ?
328	□		ID	What we're gonna be discussing at the next meeting ?
329	□			Do we know that ?
330	□		PM	I haven't gotten an agenda yet ,
331	□			um I'll put that together

Summary

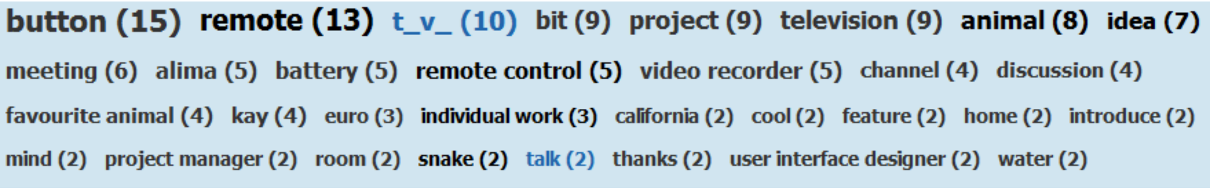
☆	ME	#1#	Hmm .
		#10#	and then I'm gonna talk a lit
—		#35#	sorry
—		#48#	I have no drawing skills wha
—		#54#	and I have no artistic talents
ME		#70#	Yes .
ID		#83#	Uh Okay .
ID		#84#	Well ,
ID		#85#	my name is Iain uh
ID		#87#	and I'm the User Interface D
ID		#88#	And I'll try and draw my fav
ID		#90#	I'll
ID		#91#	I should leave that one on th
ID		#92#	shouldn't I
ID		#94#	before I callously rub it off .
ID		#97#	Um I'm not gonna draw it qu
ID		#98#	Is that at least identifiable ?

lDesigner had negative opinions about the bit. [19
gExpert made repeated positive comments about t
gExpert had negative opinions about the batteries.
lDesigner made repeated positive comments about
lDesigner had problems concerning the buttons. [

□ (Un)Select all topics (32)

button (15) remote (13) t_v_ (10) bit (9) project (9) television (9) animal (8) idea (7)
meeting (6) alima (5) battery (5) remote control (5) video recorder (5) channel (4) discussion (4)
favourite animal (4) kay (4) euro (3) individual work (3) california (2) cool (2) feature (2) home (2) introduce (2)
mind (2) project manager (2) room (2) snake (2) talk (2) thanks (2) user interface designer (2) water (2)

Figure 4: Stage 3 design based on InfoVis principles

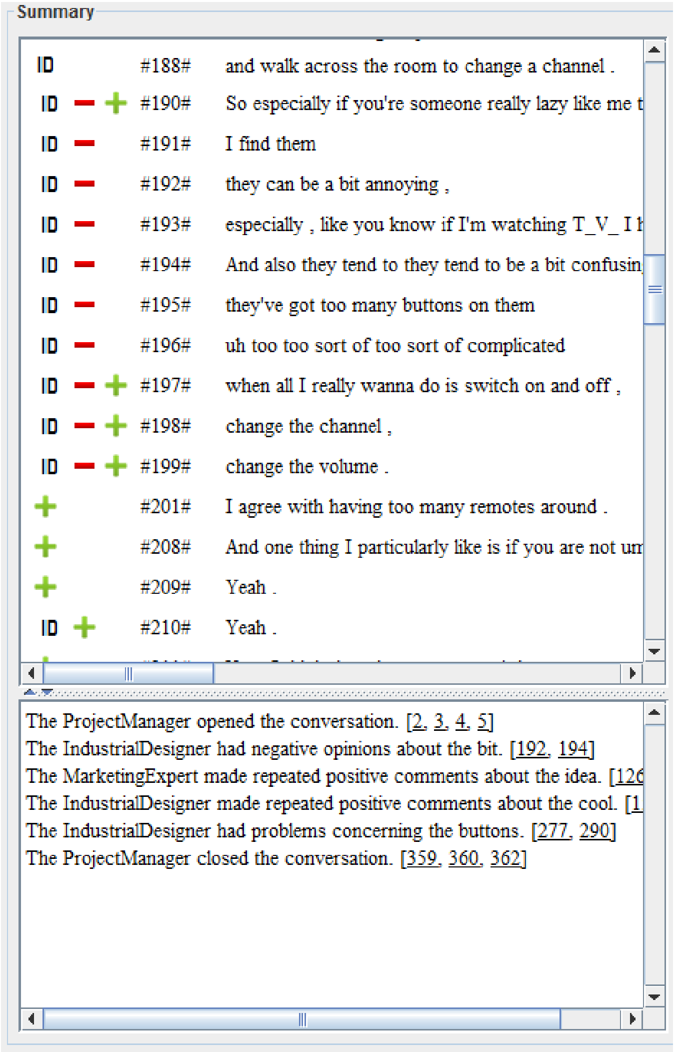


button (15) remote (13) t_v_ (10) bit (9) project (9) television (9) animal (8) idea (7)
meeting (6) alima (5) battery (5) remote control (5) video recorder (5) channel (4) discussion (4)
favourite animal (4) kay (4) euro (3) individual work (3) california (2) cool (2) feature (2) home (2) introduce (2)
mind (2) project manager (2) room (2) snake (2) talk (2) thanks (2) user interface designer (2) water (2)

Figure 6: Entity cloud using size and color for encoding and postion as sort order

#	★	Sentence Type	Speaker	Sentence
257	□	+	PM	Mm .
258	□	? + ⚠	UIE	and if um if you'd save the more complicated functions maybe for separate
259	□		ME	Mm-hmm .
260	□		PM	Okay ,
261	□	? + ⚠		so maybe have like one remote that has the main functions on , off , channel
262	□		UIE	Um .
263	□		PM	Because that is one thing that um remotes tend to have buttons that the
264	□	+	UIE	Yeah .
265	□			Mm .
266	□		PM	So like you have to have them somewhere ,
267	□		ID	Yeah .
268	□		PM	'cause you're gonna m need those special functions occasionally .
269	☑		UIE	Right .
270	□		PM	Um but not necessarily on the m the normal remote .
271	□		ID	Can I ask ,
272	□	⚠		are we designing a remote control for a television only
273	□		PM	Good question .
274	□		ID	um , and if if this device is just to be used for the television
275	□	⚠		would we even technically be possible to include video recorder functions on
276	□		PM	I don't know that yet .
277	□	⚠	ID	Um or should we just stick to just stick to having television television related
278	□		UIE	Mm .
279	□		PM	It's a good question . Um .
280	□	⚙ ⚠		I'll look into that .
281	□		ME	Mm-hmm hmm .
282	□		PM	If I can .

Figure 7: Transcript View with separate columns for Speaker and DA type; spatial position as encoding and Speaker turn parameter



Summary

ID	#188#	and walk across the room to change a channel .
ID − +	#190#	So especially if you're someone really lazy like me t
ID −	#191#	I find them
ID −	#192#	they can be a bit annoying ,
ID −	#193#	especially , like you know if I'm watching T_V_ I h
ID −	#194#	And also they tend to they tend to be a bit confusin
ID −	#195#	they've got too many buttons on them
ID −	#196#	uh too too sort of too complicated
ID − +	#197#	when all I really wanna do is switch on and off ,
ID − +	#198#	change the channel ,
ID − +	#199#	change the volume .
+	#201#	I agree with having too many remotes around .
+	#208#	And one thing I particularly like is if you are not un
+	#209#	Yeah .
ID +	#210#	Yeah .

The ProjectManager opened the conversation. [2, 3, 4, 5]
The IndustrialDesigner had negative opinions about the bit. [192, 194]
The MarketingExpert made repeated positive comments about the idea. [126
The IndustrialDesigner made repeated positive comments about the cool. [1
The IndustrialDesigner had problems concerning the buttons. [277, 290]
The ProjectManager closed the conversation. [359, 360, 362]

Figure 8: Summary View with information scent

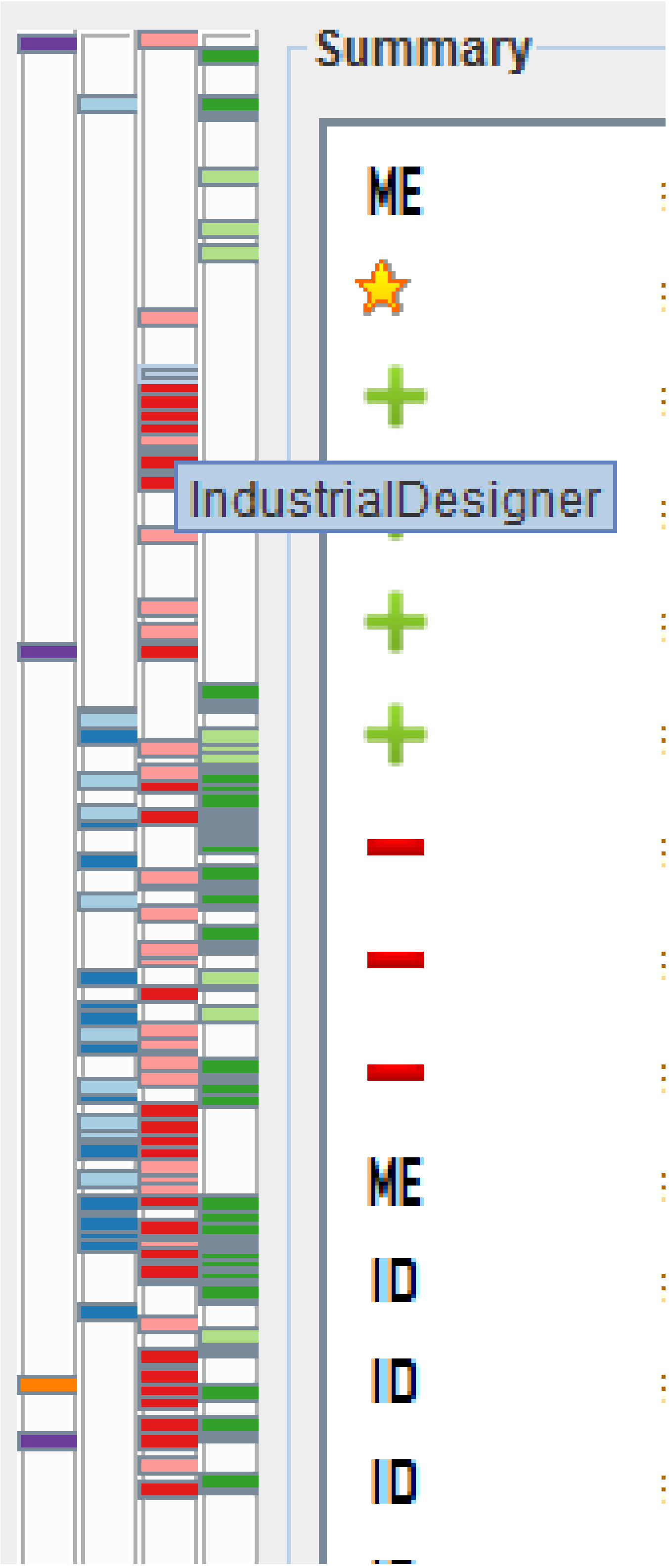


Figure 9: Marker bar with color encoding core concept and tooltip showing leaf-level concept

the conversation length. Clicking on a marker auto-scrolls the transcript and makes the part of the transcript corresponding to the tag associated with that marker visible to the user (using highlighting), this is an instance of change of viewport type of interaction).

The Entity View is a useful entry point to the conversation providing a list of precompiled keywords. However, the long list of entities impose cognitive load on the user. To enable users to concentrate on a subset of the entities I have listed them in a descending order of the frequency count and have provided a range slider where the user can specify a minimum and a maximum count for the list of entities. This fades out the entities falling outside the range selected and thus narrow down the search scope. Also, I have provided a control to change the sort order of the entities to alphabetical listing (an important aspect of text data). This is an example of interaction to change spatial position of the elements according to sort order (see Fig 6 and Fig 10).

The entity cloud is a list of the noun phrases in the conversation extracted automatically; hence, it cannot be taken as a complete enlistment of entities. Also, some adverb and verb forms are strong cues for conversational direction. For example, to find a decision taken in a conversation a user could lookup the sentences mapped to the 'Decision' DA type node or look for keywords like 'finally'. In addition to browsing the conversation in a structured way using the entity cloud and the ontology tree, I've provide a keyword search option to the user to increase the flexibility of search. When the user types in a keyword and clicks the search button it takes the user to the first occurrence of the term in the transcript and highlights the corresponding sentence. The subsequent occurrences of the term can be identified by further clicking the search button. A dialog box is shown with a message relating that there is no more occurrences of the search term once the user has reached the end of the transcript. This is similar to the search functionality of basic text editors (see Figure 11).

3.3.3 Layout Design

The design presented in stage 2 was also lacking in efficient use of screen space. In this phase, I have attempted a better layout of the multiple views (see Fig 4) in terms of use of screen real estate and usage of each of the views. I have also tried to place the controls and interaction mechanisms for a particular view close to it.

I have placed the array of marker bars beside the Transcript View to emphasize their role as an interaction mechanism for browsing the transcript. I have also made the bars use the full length of the screen space available since even after providing separate bars for the ontology concepts there is still a considerable amount of clutter induced by overlapping markers due to space contention. In laying out the parallel bars from left to right, I have taken into consideration of the usage of the particular core concepts (according to the pilot study I have conducted before). Since users find the Entity concept most important for identifying informative sentences the bar for it is closest to the Transcript View, then I have placed the Speaker marker bar and the DA type marker bar. I have given the fourth marker bar for the override and summary sentence map highest priority and have set it right beside the transcript view for two reasons; due to the small number of markers that are placed along it, it is easy for the user to overlook it if it is placed farther amidst the cluttering of colors on the other three bars. Also, the override control is a direct reflection of the user's current information need which is what we are striving to satisfy using this interface. Similarly, the range slider and sorting controls for the Entity View are placed closer to it.

The filtering controls like the Ontology View and the Entity View have been placed towards the left side of the interface. The summary View has been moved to the right side of the interface to emphasize its role as a result of the interaction (even though the summaries can be used as interaction mechanisms themselves). This places the Transcript View at the centre of the layout in keeping with the notion of centrality of the conversation transcript to the task at hand.

4 Implementation Approach

The ontology mapping has been done in stage 1 [12] and all data parsing, formatting has been done in stage 2 using Python scripts and the formatted files were stored offline to reduce data loading time. The natural language generation component used to derive the abstractive summary is also based on Java and SimpleNLG api and uses the sentence aggregation technique described in [12]. For the scope of this course project, I have made changes to the front interface which has been built using Java Swing and AWT components. Since the entire front-end application has been developed using Java, it is platform independent and free-source. The detailed implementation approach for the stage 3 redesign is as follows.

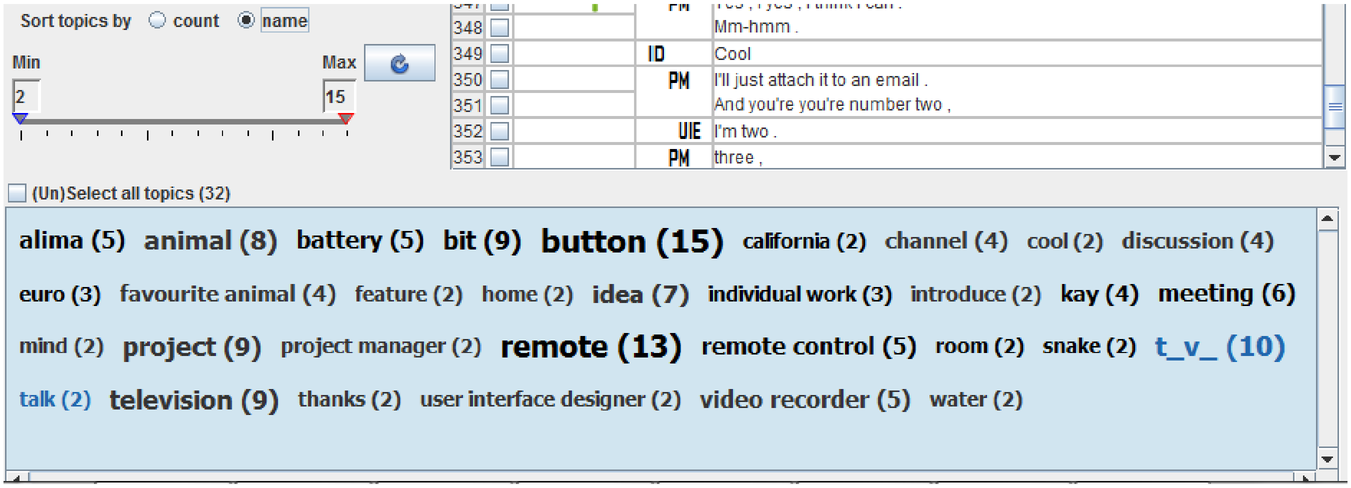
I have built the marker bar and the entity cloud visualization and interaction widgets from scratch using Java Swing components. For the Entity View, I am using a freely available data structure (package org.mcavallo.opencloud) to maintain the list of entities and their corresponding scaling weights. The range slider control for the Entity View is based on a code snippet off the web where I have customized the look and feel for my application. I have also modified the codes for the Transcript View and the Summary View using Swing component properties. The keyword search box for the Transcript View is also based on code snippet copied from the web; I have only implemented the functionality for this control.

5 Scenario of Use

One of the uses of the interface I have presented here would be as an analytical tool by the NLP experts. Almost all NLP applications provide results in a flat text format which is incomprehensible through manual inspection; and as such, statistical measures are used to verify the performance of these applications. However, statistical measures are not exhaustive and some patterns are easier to identify using human perception. I am relating such an anecdote from the early development phases of the interface. Right after adding in the count for the ontology concepts (see Fig 5), it became clear that the counts for the 'PositiveSubjective' and the 'NegativeSubjective' nodes seemed to be highly correlated. To verify whether that was a coincidence or not, I loaded different conversational test data on the interface and also checked one of the input files manually to make sure the problem was not due to a programming 'bug' on the interface end. It turned out that the error was introduced during a recent modification to the background ontology mapping application. It became apparent that the interface could be a useful verification tool as NLP applications are highly dependent on annotating data using different aspects. An NLP expert could easily extend the ontolgy based on these annotation concepts and use the interface to verify whether the algorithm is behaving in the way it is expected to.

The summary generated interactively using the interface could also be helpful for satisfying specific queries of a user on a particular conversation thread. Consider a scenario where an employee has recently joined a product designing company. Automatic meeting abstracts would allow this new personnel to prepare for an upcoming meeting or review the decisions made by a previous group. This person could be specifically looking for the commonly-used functions of the remote control that the group was designing. Using my interface this new employee could easily browse the transcripts (may be generated by Automatic Sound Recognition or as a part of meeting minutes) and find out what was the final decision, whether there was any alternatives the team considered, and what was the reasoning for and against any alternatives.

Given the above task description, the user, aided by my interface, might first skim through the entities in the tag cloud and select enti-



Sort topics by ○ count ● name

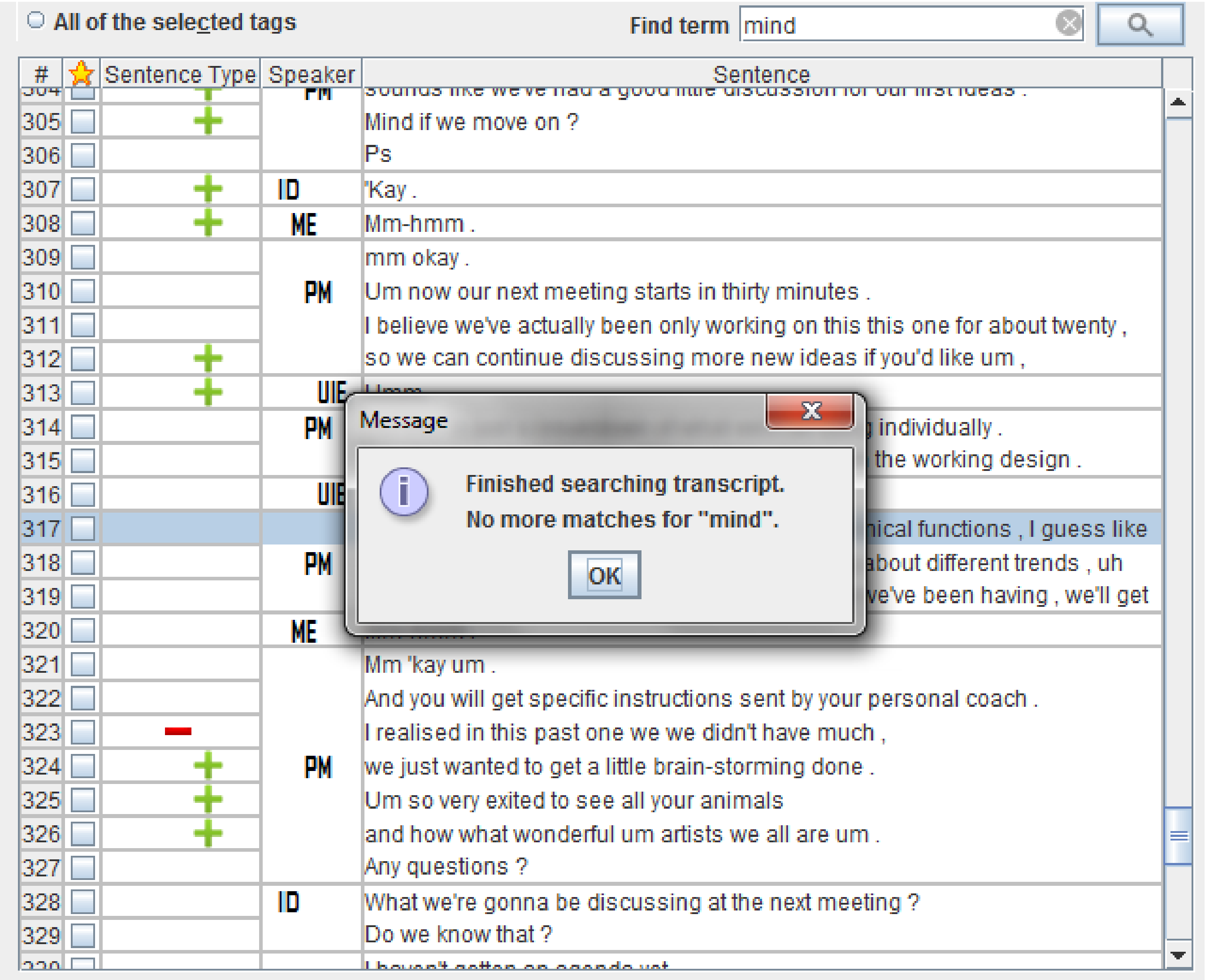
Min 2 Max 15

#		Sentence Type	Speaker	Sentence
348	□			Mm-hmm .
349	□		ID	Cool
350	□		PM	I'll just attach it to an email .
351	□			And you're you're number two ,
352	□		UIE	I'm two .
353	□		PM	three ,

□ (Un)Select all topics (32)

alima (5) animal (8) battery (5) bit (9) button (15) california (2) channel (4) cool (2) discussion (4)
euro (3) favourite animal (4) feature (2) home (2) idea (7) individual work (3) introduce (2) kay (4) meeting (6)
mind (2) project (9) project manager (2) remote (13) remote control (5) room (2) snake (2) t_v_ (10)
talk (2) television (9) thanks (2) user interface designer (2) video recorder (5) water (2)

Figure 10: Entities sorted by name using control



○ All of the selected tags

Find term mind

#	☆	Sentence Type	Speaker	Sentence
305	□	+		Mind if we move on ?
306	□			Ps
307	□	+	ID	'Kay .
308	□	+	ME	Mm-hmm .
309	□			mm okay .
310	□		PM	Um now our next meeting starts in thirty minutes .
311	□			I believe we've actually been only working on this this one for about twenty ,
312	□	+		so we can continue discussing more new ideas if you'd like um ,
313	□	+	UIE	
314	□		PM	individually .
315	□			the working design .
316	□		UIE	
317	□			nical functions , I guess like
318	□		PM	about different trends , uh
319	□			ve've been having , we'll get
320	□		ME	
321	□			Mm 'kay um .
322	□			And you will get specific instructions sent by your personal coach .
323	□	−		I realised in this past one we we didn't have much ,
324	□	+	PM	we just wanted to get a little brain-storming done .
325	□	+		Um so very exited to see all your animals
326	□	+		and how what wonderful um artists we all are um .
327	□			Any questions ?
328	□		ID	What we're gonna be discussing at the next meeting ?
329	□			Do we know that ?

Message

Finished searching transcript.
No more matches for "mind".

Figure 11: Keyword based search box

ties like 'remote control' and other entities like 'button' or 'lcd' that seem related to remote control designs (see Fig 6). This would highlight those entities in the transcript and markers along the marker bar would appear (see Fig 4). At this point the user could employ two different approaches to inspect the sentences further; she could click on each marker (see Fig 9) and read the corresponding portion of the transcript, or she could look at the filtered sentences in the summary view (see Fig 8) and make informed choice on which sentences to inspect further in the transcript. The link between the summary view and the transcript view would reduce the scroll-time if she decided to use this approach. Alternatively, when using the marker bar as the main form of interaction she could look at the distribution of the selected entities along the bar and first try to inspect the part of the conversation where the relevant entity keywords appear close to each other i.e. where the markers for the phrases 'remote control' and 'button' are concentrated. The tooltip text provided for the markers would help to identify such regions.

To narrow down search scope for the alternatives considered and the reasoning behind them the user might want to select the 'PositiveSubjective', or the 'NegativeSubjective' on the ontology view. Further, the user might even employ her real world knowledge that ultimately a decision would be made by the 'ProjectManager' and concentrate on 'Decision' type sentences by that particular speaker. To find out what other topics were highly discussed, the user might want to use the range slider provided for the entity cloud and set it to retain entities with higher counts in the conversation (see Fig 12). This would significantly reduce the number of elements shown on the entity cloud and enable the user to concentrate on a much smaller set of entities and try to correlate them with the task on hand. While looking through the transcript she might remember she had seen the word 'battery' being mentioned in the conversation at some point and that might be relevant to the power source feature of the 'remote control'. She could then type in the phrase 'battery' in the search box provided for the transcript view to look at all occurrences of the term, one at a time in the temporal order they appear (see Fig 11). On the other hand, she could use the 'Sort by name' control for the entity cloud and check whether the term 'battery' is one of the keywords. The alphabetical ordering of the entities in this mode would make the searching task faster and exhaustive for the user (see Fig 10). If 'battery' does appear on the entity cloud (is identified correctly by my background application) the user could then select it and resort to further inspection using the marker bar.

6 Discussion and Future Work

The abstractive summary is generated based on a natural language generation component. Presently, the ontology concepts can be inferred from the sentences generated but extracting them would be computationally expensive and redundant since they are already being done once in the NLG component. In the future, I shall modify the summary generation application to output the DA type, speaker and entity annotations as well as the list of summary sentence to transcript sentence map and I'll include information scents similar to the extractive summary for the abstractive summary as well. Due to time constraints, I was not able to implement the functionality of linking the individual component sentences in the list to the transcript, which is an easy extension for the future. Recurring patterns or phrases are common among individual posts in an email or blog thread or among the series of meetings in the AMI corpus. An overview of this type of repeatitions may be helpful to the users to satisfy complex query needs. A way to analyze such recurring patterns has been discussed in [2]. The check box override control could be extended in functionality in the reverse direction to exclude one of the tagged sentences from the generated summary. There is a number of ways the entities could be clustered based on their semantic relationships or collocation. The concepts presented in PhraseNet and WordTree could be extended to visualize such groupings of entities referred to in the conversation.

The screen real estate could be put to better use by using dynamic layout. In that case, some of the views and controls could be kept collapsed while the user is focusing on conversation analysis using particular widgets. I am also thinking about a more intelligent placement algorithm for the markers on the marker bar to reduce clutter. I need to find a balance between efficient display of the tags and computational cost involved for more sophisticated placement strategy.

There is work to be done to improve the performance of the interface in terms of interaction time. To achieve this I am redesigning the code structure to make it more optimal and to use data structures that have a better access time. I am also going to run some of the more computationally expensive code segments on a separate thread to leave the GUI rendering thread free to handle the graphics and other interactions of the user while the computation is performed.

Finally, the efficiency and usability of the current design can be assessed only through a formative user study. I am currently designing an experiment for this purpose and plan to carry out the study in near future.

In conclusion, I have presented an interface for browsing human conversations and summarizing them using generic concepts represented in an ontology. The main contribution of my project (the entire scope of my thesis) is the extensibility of the approach to multi-modal conversations. The redesign implemented within the scope of this course project will provide the user with a more efficient interaction mechanism and will reduce the time required to identify informative sentences within the transcript.

Acknowledgements

I've been working on this project as my M.Sc. thesis under the supervision of Dr. Giuseppe Carenini since September, 2010. I've also received valuable feedback on the interface from Dr. Raymond Ng and Dr. Gabriel Murray over the length of that time. I would like to thank them all for their guidance.

References

[1] C. Albrecht-Buehlera, B. Watson, and D. A. Shamma. TextPool: Visualizing Live Text Streams. In *InfoVis*, 2004.

[2] D. Angus, A. Smith, and J. Wiles. Conceptual Recurrence Plots: Revealing Patterns in Human Discourse. In *IEEE transactions on Visualization and Computer Graphics, Volume 17 Issue 12*, 2011.

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

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

[5] 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.

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

[7] F. B. V. Frank van Ham, Martin Wattenberg. Mapping Text with Phrase Nets. In *Infovis*, 2009.

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

[9] 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.

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

[11] F. B. V. Martin Wattenberg. The Word Tree, an Interactive Visual Concordance. In *Infovis*, 2008.

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

[13] W. B. Paley. TextArc: Showing Word Frequency and Distribution in Text. In *InfoVis Poster Compendium*, 2002.

[14] S. Rashid and G. Carenini. An Ontology-based Visual Interface for Browsing and Summarizing Conversations. In *Proc. of VISSW workshop, Palo Alto, California, USA*, 2011.

[15] 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 IV, New Jersey, USA*, 2009.

[16] 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.

[17] G. Tur, A. Stolcke, L. Voss, S. Peters, D. Hakkani-T*ür,J. Dowding,B. Favre,R. Fernández,M. Frampton,M. Frandsen,C. Frederickson,M. Graciarena,D. Kintzing,K. Leveque,S. Mason,J. Niekrasz,M. Purver,K. Ried*

[18] 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.

[19] F. B. Viegas, M. Wattenberg, and J. Feinberg. Participatory Visualization with Wordle. In *InfoVis*, 2009.

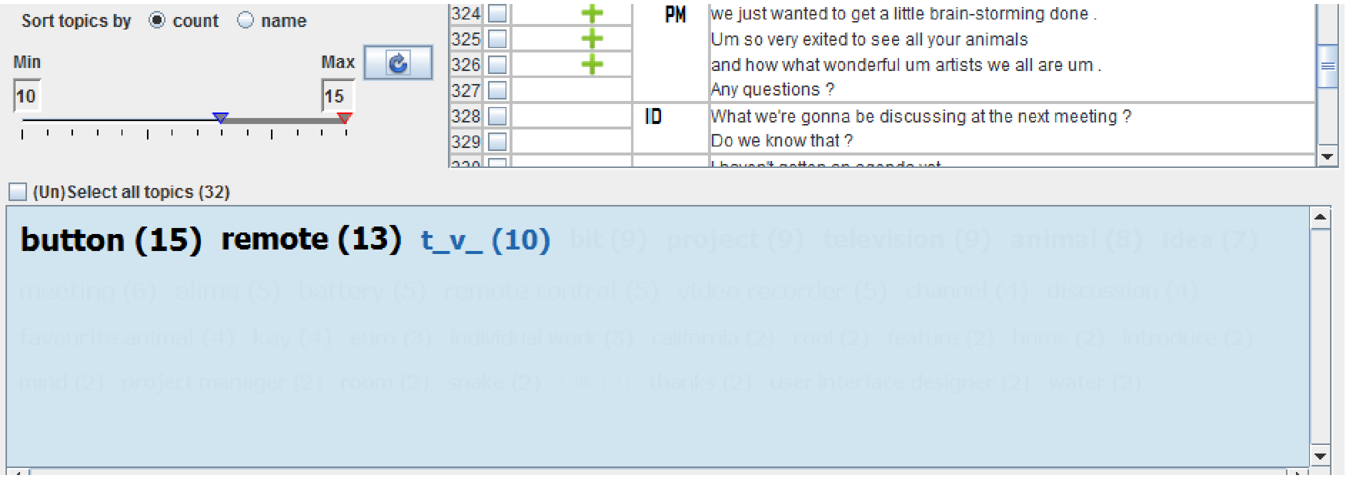
[20] M. Wattenberg. Arc Diagrams: Visualizing Structure in Strings. In *Infovis*, 2002.

[21] M. Weiss-Lijn, J. T. McDonnell, and L. James. Supporting Document Use Through Interactive Visualization of Metadata. In *Infovis*, 2001.

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

[23] J. H. Wesley Willett and M. Agrawala. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. In *Infovis*, 2007.

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



Sort topics by ● count ○ name

Min 10 Max 15

324	□	+	PM	we just wanted to get a little brain-storming done .
325	□	+		Um so very exited to see all your animals
326	□	+		and how what wonderful um artists we all are um .
327	□			Any questions ?
328	□		ID	What we're gonna be discussing at the next meeting ?
329	□			Do we know that ?
330	□			I haven't gotten an agenda yet

□ (Un)Select all topics (32)

button (15) remote (13) t_v_ (10) bit (9) project (9) television (9) animal (8) idea (7)
meeting (6) alima (5) battery (5) remote control (5) video recorder (5) channel (4) discussion (4)
favourite animal (4) kay (4) euro (3) individual work (3) california (2) cool (2) feature (2) home (2) introduce (2)
mind (2) project manager (2) room (2) snake (2) talk (2) thanks (2) user interface designer (2) water (2)

Figure 12: Range slider and its effect on entity view