Meeting Structure Annotation: Data and Tools

Alexander Gruenstein
Spoken Language Systems
MIT Computer Science and
Artificial Intelligence Laboratory
Cambridge, MA 02139, USA
alexgru@csail.mit.edu

John Niekrasz **Matthew Purver**
Center for the Study of Language and Information
Stanford University
220 Panama Street
Stanford, CA 94305, USA
{niekrasz,mpurver}@csli.stanford.edu

Abstract

We present a set of annotations of *hierarchical topic segmentations* and *action item subdialogues* collected over 65 meetings from the ICSI and ISL meeting corpora, designed to support automatic meeting understanding and analysis. We describe an architecture for representing, annotating, and analyzing multi-party discourse, including: an ontology of multimodal discourse, a programming interface for that ontology, and an audiovisual toolkit which facilitates browsing and annotating discourse, as well as visualizing and adjusting features for machine learning tasks.

1 Introduction

The automatic processing and understanding of multiparty meetings has emerged recently as a major area of research. Technically, meetings present many interesting multidisciplinary challenges; for instance, they have multiple interacting participants and contain spontaneous speech, movement, and gesture. Commercially, they are interesting as they often involve important decisions, yet they are usually poorly documented. Several major projects studying meetings are underway, including Mapping Meetings,[1] M4,[2] AMI,[3] ISL,[4] IM2,[5] and CHIL.[6]

In this paper, we view meetings from the perspective of building meeting understanding components which comprise part of the *cognitive personal office assistant* being designed for the CALO project.[7] The types of assistance envisioned include summarizing the meeting, actively bringing attention to relevant documents, and helping the collaborative creation of documents in the course of the meeting. Additionally, the content of meetings will be presented in a *meeting browser* which allows a user to browse a top-level summary, locate pertinent portions, and "drill down" into more detailed structure as desired.

In order to summarize meeting structure in a useful way, it is therefore critical to first understand what sort of structure best assists humans in browsing or reviewing the contents of meetings. With this in mind, we describe an *application-driven* approach undertaken to annotate a set of meetings with relatively coarse structural annotations with the hopes of spurring development of automatic structural segmentation algorithms in this difficult domain. In the first half of the paper, we describe a new set of annotations of the ICSI (Janin et al., 2003) and ISL (Burger et al., 2002) meeting corpora that mark *hierarchical topic segmentation* and *action items*, and then analyze inter-annotator agreement.

The remainder of the paper discusses an architecture developed in the course of the project for both collecting annotations over, and performing research tasks involving, multi-party discourse. In particular, we discuss an *ontology of multimodal discourse*, along with its corresponding *ontology programming interface*. We then present an *audiovisual toolkit* built using this programming interface, which was in turn used to develop the tool used to perform the annotations, as well as several other tools designed for manipulating meetings.

The annotations and tools described in this paper are at `http://godel.stanford.edu` under *Software*.

2 Annotation Motivations and Schema

We focus on two types of discourse structure annotations. The first, *topic segmentation*, breaks the discourse up into a (hierarchical) sequence of topics. The second, *action item subdialogues*, marks particular utterances as

[1] http://labrosa.ee.columbia.edu/mapmeet/
[2] http://www.m4project.org
[3] http://www.amiproject.org
[4] http://penance.is.cs.cmu.edu/meeting_room/
[5] http://diuf.unifr.ch/im2/
[6] http://chil.server.de/
[7] http://www.ai.sri.com/project/CALO

being relevant to the discussion or assignment of action items. In this section, we describe our motivations in studying these phenomena, related work, and the iterative process by which we refined an application-driven annotation schema.

We worked with the ICSI Meeting corpus (Janin et al., 2003) and the ISL Meeting Corpus (Burger et al., 2002) because both contain high-quality close-talking microphone recordings of conversational speech in a meeting environment, as well as word-level transcriptions and utterance-level timing information. We focused mainly on the ICSI corpus because its contents most closely matched our task of processing fairly informal, office-style meetings. In addition, extensive annotations have already been completed on the ICSI corpus, including: dialogue acts (Shriberg et al., 2004), "hot spots" (Wrede and Shriberg, 2003), and some work on topic segmentation (Galley et al., 2003; Carletta and Kilgour, 2004).

2.1 Topic Segmentations

A significant challenge in spoken discourse segmentation is providing a concrete definition of the problem – the desired concepts of both *topic* and *segmentation*. To that end, we first briefly discuss the conceptualizations – and motivations behind those conceptualizations – that have arisen in the related fields of segmenting text and monologue. We then discuss previous work in segmenting discourse, our own motivations, and finally (in section 2.1.1) outline an annotation schema derived from these motivations.

Text and Monologues The segmenting of text documents is often motivated by information retrieval tasks – for instance, so that a single appropriate segment can be returned matching a query. In some cases, topic boundaries are hand-annotated, as in (Hearst, 1994). However, topic boundaries are often artificially created by concatenating multiple articles together, as in (Galley et al., 2003; Choi, 2000). Moreover, since text is written linearly, usually with clearly punctuated boundaries in the form of sentences and paragraphs, it is natural to assume that topic boundaries will occur at such places. Thus, such "natural" boundaries both define and limit the search space. In addition to text, there has been much research in segmenting *non-conversational speech*; essentially monologues or series of monologues. For example, much work has been done on automatically segmenting broadcast news, *e.g.* (Tür et al., 2001; Beeferman et al., 1999; Allan et al., 1998).

The tasks of segmenting text and monologue are similar in that both tend to have fairly well defined topic structure. In the case of artificial text corpora created through concatenation, topic boundaries can be objectively defined over the concatenated article boundaries. News broadcasts tend to consist primarily of scripted speech – with little spontaneity – produced by highly practiced professionals (though some work has also been done on more spontaneous monologues, see (Passonneau and Litman, 1997)). Topic boundaries in news broadcasts are designed to be obvious, with unambiguous shifts from one story to the next. In both domains, automatic segmentation algorithms tend to rely primarily on lexical co-occurrence statistics to calculate a measure of *lexical cohesion* between chunks of text (Hearst, 1994; Hearst, 1997). In the case of monologue, prosodic cues are often utilized as well (Tür et al., 2001; Hirschberg and Nakatani, 1998).

Discourse When turning to spontaneous discourse, most previous work has followed this text/monologue approach: for example, when (Galley et al., 2003) annotated 25 meetings in the ICSI Meeting corpus for topics, the discourse was represented *linearly* as a series of non-overlapping utterances, topics were represented as a linear sequence of segments, and topic boundaries were allowed only at *speaker changes*. Although we are aware of one project in which *hierarchical* topic annotations are being used (on the ICSI corpus using the NITE XML toolkit (Carletta and Kilgour, 2004)), no annotations are yet publicly available.

Rather than adapting the task of discourse segmentation to make it look more like a text segmentation task, we took an *application-driven* approach to segmenting discourse. Our motivation for topic segmentation was to enable broad understanding of a discourse, providing a coarse summary segmentation for broad-perspective user browsing capabilities, and allowing for selective "drill-down" and replay; for more detailed discussion of the utility of high-level segmentations, see (Banerjee et al., 2005). We therefore wanted to collect annotations which can be leveraged specifically to provide such capabilities for a digital personal office assistant. Specifically, we instructed the annotators to look at the problem of providing a topic segmentation from the perspective of utility: if they were reviewing a meeting they might not have attended, what segmentation would help them quickly "drill down" to portions they might be particularly interested in reviewing. While a bit vague, this description of the task avoids biasing the annotators toward relying on particular discourse phenomena or restricting them to particular boundary locations; (Ries, 2001) argues that such an application-driven approach, with linguistically naive coders, may help best represent end-users of meeting browser systems.

This application-driven approach proved difficult at first, resulting in low inter-annotator agreement among the two undergraduate annotators in the first five meetings that were annotated. However, through discussions

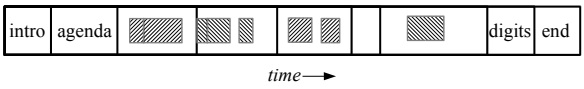


Figure 1: A sample hierarchical meeting segmentation

of the annotations (often using the comparison tool, discussed in section 5.2) – although banning discussion that would result in annotators taking away shared concrete heuristics – an acceptable level of inter-annotator agreement was reached for the majority of meetings (see section 3). Agreement results eventually reached a plateau, at which point further discussion of the annotation guidelines was terminated. At this point, guidelines were then drawn summarizing the result of these discussion: see (Gruenstein et al., 2004).

2.1.1 Topic Segmentation Schema

In this section, we describe the schema that resulted. Meetings were segmented according to a two-level hierarchical segmentation schema. In the top (*major*) level of the hierarchy, the entire meeting is wholly and contiguously segmented, where segment boundaries symbolize highly salient breaks in discourse structure and/or distinguish parts of the discourse between which there is an obvious difference in subject matter. In the second (*minor*) level of the schema, major segments are optionally subsegmented without a requirement for contiguity, but with overlapping segments forbidden. Minor segments signify either a temporary digression or a more focused discussion of the subject matter, while still remaining directly relevant to the encompassing major segment. Our pilot annotation work indicated that restricting topic breaks to speaker changes was an unnatural restriction. Instead, our schema allows topics to start and end at any point in the discourse, even in the middle of a single speaker's utterance. Some ramifications of this choice are discussed in section 3. Figure 1 depicts a meeting segmented according to the schema, with vertical lines separating major topics, and shaded areas representing minor topics.

Annotators also gave brief descriptive names to topics, though no standards were set as to the format or content of the assigned names, with the exception of the following *reserved* topic names:

- *AGENDA*: the portion of the meeting in which the agenda is presented and discussed
- *INTRO*: speech before the meeting "officially" begins (appears in every meeting, though may have zero length)
- *END*: speech after the meeting "officially" ends (appears in every meeting, though may have zero length)
- *TECHNICAL DIFFICULTIES*: a period in which there are technical difficulties with recording equipment
- *DIGITS*: the digits task in the ICSI meeting corpus [see (Janin et al., 2003)]

Except for *AGENDA*, the reserved names simply serve the purpose of highlighting portions of the recording which might not be considered part of the meeting proper; in section 3.3 we discuss how they play a role in defining a reference segmentation. In addition, if a new topic is a continuation of a discussion of a previous topic left off earlier, the convention is used that the same descriptive text is given for both topics – implicitly linking them.

2.2 Action Items

Though the focus of the annotation work was hierarchical topic segmentation, annotators also marked *action items*. Previously, we have shown how simple task-assignment charts can be inferred from highly scripted, multimodal meetings (Kaiser et al., 2004). In moving to free-form meetings, identifying *decision points* like action items follows as a natural first step in extending this work.

For the purposes of annotation, we define an action item loosely as a task which is discussed in the meeting and then assigned to a participant (or participants) to complete at some point after the completion of the meeting. In our schema, action items are defined as sets of *utterances*, rather than start and end times: this is possible because action items are usually discussed only briefly, so it is feasible for an annotator to pinpoint particular utterances in which the discussion occurred. Moreover, it is useful to identify as specifically as possible the utterances in which action items were discussed, as not all speech within a time window may be relevant due to the high levels of speech overlap in multi-party conversations.

It may be worthwhile in future annotation passes to further classify each utterance into categories such as *proposal*, *discussion*, and *assignment*. Furthermore, it may be useful to mark information particular to the task, such as: the person it has been assigned to, its deadline, and its relation to other tasks.

3 Analysis of Collected Annotations

We collected annotations for 49 meetings of the ICSI corpus and 16 of the ISL corpus. In this section, we provide a statistical analysis of our annotations, along with some more qualitative observations. We describe multiple algorithms which have been applied to the data to make our analysis possible. We also provide an analysis of inter-annotator agreement using multiple metrics. Last, we compare our annotations to other similar datasets.

3.1 Pre-processing

Some of the annotated meetings contain portions that should not be included in a proper analysis of topic structure, including the ICSI digit readings and times when the recording mechanism was not working properly. Also, every meeting recording has a beginning and end which do not actually contain meeting dialogue. Before analysis, we therefore perform pre-processing of our annotations to produce a segmentation that is free from these

idiosyncrasies. Because our annotators were asked to annotate these special cases, our pre-processing algorithm simply takes the union of the set of *INTRO*, *END*, *DIGITS*, and *TECHNICAL DIFFICULTIES* segments from both annotators and removes those portions of the discourse from both annotations, shifting the remainder into a contiguous discourse. All the analyses presented below were done after pre-processing.

3.2 Segment and Break Classification

While most text segmentation methods constrain the number of possible segmentations by specifying a finite set of discrete locations where segment boundaries may occur (most often at sentence boundaries), our annotators were free to assign boundaries at any time during the discourse. Unfortunately, this complicates our use of standard evaluation metrics, and it doesn't suit iterative automatic discourse segmentation algorithms which operate at discrete intervals of time.

To overcome these obstacles we transform our annotations into a set of classifications in two ways, arriving at what we call a *segment classification* and a *break classification*. For each of the two, the first step is to divide the discourse into temporal units based on a set of possible break locations, e.g. a set of evenly-spaced temporal values, utterance start times, or speaker changes. We use evenly-spaced intervals of 20 seconds in our analysis.

In the case of evenly-spaced windows, a discourse d is evenly divided into $i = |d|/n$ non-overlapping contiguous temporal intervals of length n, with the last window realizing any remainder and possibly being cut short. For the *segment classification*, each temporal unit is classified as to which topic segment it belongs. Temporal units which contain segment boundaries are classified simply by determining in which half of the unit the annotated boundary lies. If it lies in the later half, the unit is classified as belonging to the previous topic segment. For the earlier half, it is classified with the following topic segment. This produces segment boundaries which are between windows.

For *break classification*, each unit is classified as to whether or not it contains a topic boundary. This latter interpretation is essential for making use of the Kappa agreement statistic when the number of topic segments is unconstrained, as it is here. This may be transformed back into a set of segment boundaries by placing boundaries at the center of windows which have been classified as containing a topic break.

3.3 Reference Segmentation

Another essential processing step is to produce a reference segmentation from our individual annotations. This is important to providing a comparison to other annotations such as those used in (Galley et al., 2003), and for training automatic segmentation algorithms. Galley, et al. create a reference segmentation by establishing sets of topic boundaries based on co-occurrence between annotations within 20 seconds. They then choose those sets which have been annotated by a majority and establish a boundary at each set's median time value.

In our current method, we employ the same strategy of discarding the minor segments. However, we believe benefit can be derived using our second tier of segmentations as there are many cases where topic boundaries are annotated as a major shift by one annotator and as a minor shift by the other, suggesting some level of agreement that should be used. Also a second tier of segmentation in an automatic segmentation application would likely be useful for more localized "drill-down". Therefore, we do not believe this strategy should be a hard and fast rule: we provide our segmentations as individual annotations without establishing a defined reference. We will likely employ different strategies in the future for establishing a reference segmentation which incorporates minor boundaries.

3.4 Evaluating inter-annotator agreement

In this section we present the results of evaluating agreement between our two annotators and compare multiple agreement metrics. The results show variance among meetings, suggesting that the topic segmentation task may be ill-formed for certain classes of meetings.

The current standard metric for measuring inter-annotator agreement in classification tasks is the kappa statistic (K) (Carletta, 1996). While K is a good measure of how well annotators can agree on pinpointing topic breaks at time points, it does not accomodate near-miss break assignments in which annotators label different nearby time points as topic breaks. For the evaluation of segmentation algorithms specifically, two metrics are most commonly used: P_k (Beeferman et al., 1999) and *WindowDiff* (WD) (Pevzner and Hearst, 2002). These were designed principally to evaluate text segmentation algorithms that operate at sentence boundaries, but can be applied to continuous-time segmentations through the use of windowing. P_k accommodates near-miss labelings by considering how likely two time points are to be assigned to the same topic, while WD further refines this notion by measuring the number of intervening topic breaks between a time point assigned by annotators to distinct topics. Each metric provides a reasonable, though different, evaluation of inter-annotator agreement. Results given below show a high degree of correlation among them.

Our measurement of K follows that suggested in (Carletta, 1996) and described fully in (Siegel and N. J. Castellan, 1988):

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

This measures pairwise agreement on classification tasks, correcting for chance, where $P(A)$ is the probability of agreement and $P(E)$ is the probability of chance agreement between two annotators. Increasing values of K indicate better agreement. We use the break classification form of our annotations when calculating this metric.

Our second measurement is a variation on P_k, which is computed as follows:

$$P_k(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^{N-k}(\delta_{\mathbf{a}}(i, i+k) \,\overline{\oplus}\, \delta_{\mathbf{b}}(i, i+k))}{N-k}$$

P_k estimates the probability that two randomly drawn temporal values occurring during the discourse are classified as being in *different* segments by the two segmentations $\mathbf{a}$ and $\mathbf{b}$ – thus, decreasing P_k indicates better agreement. Here, $\delta_{\mathbf{x}}(t_1, t_2)$ is an indicator function which evaluates to 1 if the segmentation $\mathbf{x}$ places the times t_1 and t_2 in the same segment. The $\overline{\oplus}$ operator represents the XNOR function. As mentioned in (Beeferman et al., 1999), if the value k is set to half the mean topic segment length, the metric provides appropriate results for all degraded forms of segmentation, including random segmentation. We impose a slight variation on the calculation of k by not treating one annotation as a reference and the other as a hypothesis, but rather by incorporating both annotations when calculating the average segment length.

The third and final metric, WD, is the most recently proposed and is a variation on P_k intended to improve its tolerance of near-misses and varying segment size distributions:

$$WD(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^{N-k}(|b_{\mathbf{a}}(i, i+k) - b_{\mathbf{b}}(i, i+k)| > 0)}{N-k}$$

Here, $b_{\mathbf{x}}(t_1, t_2)$ replaces $\delta_{\mathbf{x}}(t_1, t_2)$ from P_k and is the number of segment boundaries occurring between times t_1 and t_2 in the segmentation $\mathbf{x}$. This metric is different from P_k in that a penalty is assessed at each evaluation point if the number of segment breaks in the interval is not equal between the annotations. In P_k, the number of breaks is not counted and a penalty is only assessed if one totals 0 and the other does not. For WD, we impose the same change to the calculation of k as we do in our calculation of P_k.

Because our annotations have continuous-time boundaries, we must establish a stepping method for i. Following (Galley et al., 2003), we use 20-second stepping intervals. An investigation of inter-annotator agreement for varying step sizes from 5 to 60 seconds showed no significant change in P_k or WD. An evaluation of K with varying break classification window widths showed a maximum at near 20 seconds. For the purposes of transparency and descriptiveness, we include measurements of all three of the above metrics in our evaluation, using a 20-second window width and/or step size.

	Major topics	Major and minor topics
WD	33.8%/32.2%	34.8%/34.0%
P_k	27.9%/25.0%	27.1%/26.1%
K	40.9%/44.0%	44.6%/46.5%

Table 1: Mean/median agreement on segmentations

3.5 Results

Multiple graphs showing results for inter-annotator agreement may be found in Figure 2. The top three plots show agreement based only on major topic boundaries. The bottom three include minor topic boundaries in the evaluation. Each of the columns rows shows a pair-wise comparison of two of the three metrics. Means and medians are provided in Table 1.

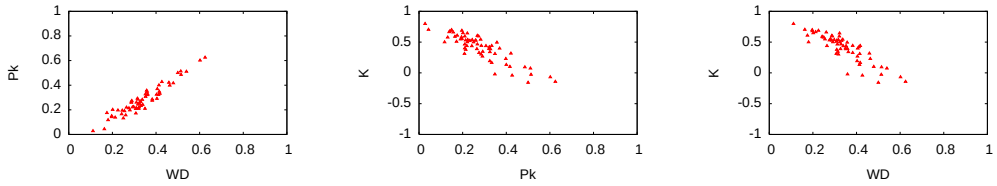
As expected, the metrics show a high level of correlation (correlation coefficients are given in the figure captions). It is difficult to say what values for our metrics signify a "good" level of reliability in the annotations. In computational linguistics, a value of $K = .67$ is generally used as a cutoff for reliable analysis, though it has been suggested on multiple occasions that this is not appropriate for all tasks (see (Eugenio and Glass, 2004) for a discussion). Undeniably low scores do occur in our annotations. This is often found for meetings which involved presentations of visual information, which made the audio-only annotation task difficult. Some of this information may be gleaned from the available annotator notes. Poor agreement and self-evaluation by the annotators on some meetings suggest that some of the annotations should not be used. It should be noted that there are more numerous outliers in the evaluation of major segments only, which is a result of there being some meetings which were only annotated as having as few as two major boundaries.

In addition, the two annotators marked 765 and 1076 utterances respectively as belonging to discussion about action items. We have yet to do significant analysis of these annotations and wish to produce further annotations of decision-making processes before using the data.

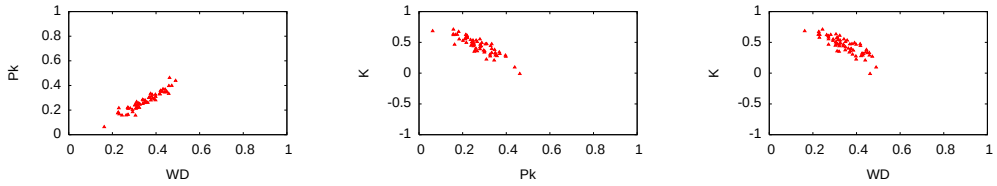
3.6 Comparison with similar corpora

In (Galley et al., 2003), 25 of the meetings in the ICSI Meeting corpus were hand annotated for topic breaks. A minimum of three annotators per meeting were given the task of deciding if each *speaker change* in a linearly represented meeting constituted a topic break.

Due to their process of establishing a reference segmentation, topic boundary frequency is significantly different between their annotations and our individual annotations. Our annotators produced major segments with



(a) WD vs. P_k. Corr = .95 (b) P_k vs. K. Corr = .87 (c) WD vs. K. Corr = .87



(d) WD vs. P_k. Corr = .94 (e) P_k vs. K. Corr = .88 (f) WD vs. K. Corr = .83

Figure 2: Segmentation inter-annotator agreement. (a)-(c) include major topics only; (d)-(f) include major & minor

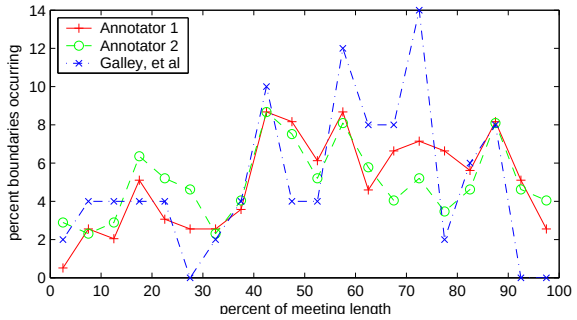


Figure 3: Distribution of boundaries over meeting duration.

an average length of 225 and 212 seconds respectively, while Galley, et al.'s average 684 seconds. Their annotations total 12.6 hours, while ours total 52.7, though after removing meetings with poor agreement results, these figures will be more similar.

One noteworthy statistic is the distribution of topic boundaries over meeting duration, depicted in figure 3. The distribution is shown for each of our annotators and from Galley, et al. While the total number of meetings is different between the two sets, there are significantly more topic changes in the latter half of the meetings for each. It will be interesting to take note of this statistic in other corpora to see if the trend is universal. It is unclear if this is a by-product of the annotation process or of the meeting itself.

4 Architecture for Meeting Annotation, Research, and Browsing

We now turn to describing the architecture we have developed over the course of working with multi-party discourse. Our architecture has grown out of three major threads of research: (1) performing and viewing annotations of discourse, (2) working toward automatic discourse segmentation, and (3) integrating our work with other components comprising a digital office assistant – including components responsible for vision, gesture, and high-level reasoning. In this section, we discuss a *multimodal discourse ontology* (MMDO) which has resulted from these efforts, as well as an *audiovisual toolkit* for manipulating multi-party discourse and annotations of that discourse. In section 5 we give examples of tools built thus far which make use of this architecture.

4.1 MMDO and Ontology Programming Interface

In order to generically represent both corpora and annotations of those corpora, we have devised a *multimodal discourse ontology* (MMDO). The MMDO is fully described in (Niekrasz and Purver, 2005; Niekrasz et al., 2005); here, we give a brief overview focusing on how the ontological framework allows us to unify several research threads. In accordance with our principles of *application-driven* annotations, the MMDO is a suitable representation on top of which to build agents capable of integrating with others into a digital personal assistant.

The MMDO follows recent trends in information technology which put *semantics* in the limelight of data-driven research, the most significant being the Semantic Web (Berners-Lee et al., 2001) which brings ontology and knowledge engineering in contact with the World Wide Web. Following this trend, research in annotation of both linguistic and multimedia resources has begun to shift away from the paradigm of *markup* toward that of *semantic annotation* (Farrar, forthcoming; Geurts et al., 2003). While the former are commonly schematized in a manner similar to an XML DTD, the latter is grounded in a formal ontology, providing an expressive semantics to the annotation and allowing inference.

The MMDO can be found as part of the software architecture in figure 4. At the core is a general upper ontol-

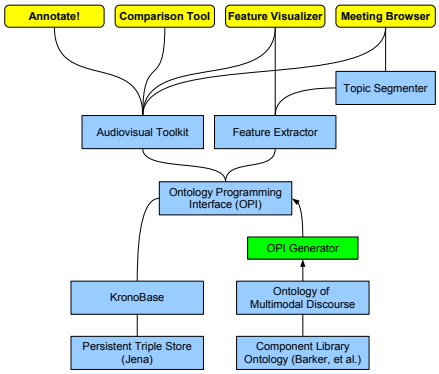


Figure 4: Architecture for annotating, browsing, and automatically segmenting multi-party discourse

ogy called the Component Library (Barker et al., 2001), the core ontology used in the CALO project. This provides the most abstract level of semantics to the annotation schema such as events, entities, and roles. Building from these general concepts, we have designed an ontology of multimodal discourse. This layer encodes the concepts important to understanding discourse, such as utterances, words, speaking events, writing events, linguistic constituents, gesturing, etc. In its design, we place an emphasis on unifying our multiple research threads (e.g. human-computer dialogue, open-domain parsing, meeting modeling, and lexical semantics) both theoretically and pragmatically where possible, as well as on capturing as many of the commonly-held concepts in natural language research as possible.

Using this ontology, we create a custom-made Java API, which we call an *ontology programming interface* (OPI), using an algorithm which encodes the hypernymic relations in the ontology as Java class inheritance and encodes the class relations (attributes) as Java methods. The OPI is written to interface with a triple-store database back-end, which supports persistent access to annotations, currently implemented using the Jena Semantic Framework. *Kronobase* is a layer we have developed for meta-annotation, which allows the recording of important aspects of annotation, including who performed it, when it was performed, and on which resources (other annotations) it is dependent.

4.2 Audiovisual Toolkit for Meeting Annotation, Research, and Browsing

Leveraging the OPI is a generic *audiovisual toolkit* for working with discourses and their associated annotations. The toolkit provides functionality for graphically displaying information stored in the ontology, thus creating a generic platform in which any discourse can be loaded so long as it can be converted to the appropriate format. Moreover, since the underlying ontology used to represent annotations is the same as that used internally in the CALO agent, the toolkit can be used to build applications which can be integrated directly into end-user applications.

The audiovisual toolkit has been the primary ingredient in building several annotation-related software tools discussed in section 5: *NOMOS* (an annotation tool), the *Comparison Tool*, and the *Feature Visualizer*. The first two were designed for the use of annotators, while the *Feature Visualizer* is for researchers working on conversational understanding systems. In addition, the audiovisual toolkit serves as the basis for a *Meeting Browser* tool currently under development, with which end-users will be able to browse through an automatically annotated meeting. Figure 4 shows the architectural hierarchy contributing to each piece of software. The audiovisual toolkit is implemented entirely in Java, as are the tools built on top of it. Each has been used extensively under several platforms, including Windows 2000/XP, OS X, and Linux. The toolkit provides an intuitive interactive interface for viewing and listening to a multiparty conversation, potentially with annotations overlayed. Screenshots of *NOMOS* and the *Feature Visualizer* can be found in figure 5 (see the appendix). Both tools consist of fairly minor additions to the generic framework, as the common interface of both demonstrates.

Transcription In the GUI, each conversational participant is assigned a *track*, in which the transcribed (or recognized) utterances of that participant are displayed – moving from left to right moves along the time axis. In the screenshots shown in figure 5, each of the top seven horizontal tracks are assigned to a particular conversational participant. Each small box on a track shows the transcription of a single utterance, where the left- and right-hand sides of each box are time-aligned with the start and stop time of the utterance. The vertical slider on the left-hand side can be used to *zoom* in and out, allowing the user to adjust how much of the transcription is viewed at one time; this makes it easy to move from a microscopic view of the discourse to a global one, and back. While figure 5(a) displays about a minute of discourse, figure 5(b) shows about half an hour.

Annotations *Major* topics are signaled graphically on the tracks by alternating the background color between blue and cyan. The *minor* breaks are indicated by the narrower bands of alternating light and dark gray centered vertically in the track. For instance, in figure 5(a) there are 2 major topics visible in the time slice shown; in addition, the first major topic is a parent to two child minor topics. Brief descriptions assigned to each major and minor topic are displayed in each track. Finally, the entire hierarchy of topics is shown in the upper-left-hand corner – clicking on any topic will shift the track display

below to the start of that topic.

An example of annotations for *action items* is also displayed in figure 5(a). Two utterances by the speaker in the second track have been shaded green to indicate that they are both related to the same action item. Moreover, the upper-right-hand corner shows that this discourse has been annotated with two action items. A brief description of each appears, followed below by a summary of information about each utterance comprising that action item: the speaker id of the speaker who uttered it, its start and stop time, the annotator's id, and the transcription of the utterance itself. Clicking on an utterance will scroll the track display to show that utterance. Each action item is assigned a color, shown both in the summary in the upper right and in the highlighted utterances in the display.

Finally, the *hide* button along the bottom toggles whether the transcription is currently visible or hidden. Zooming out and hiding the transcription is an extremely useful way to quickly get a feel for the structure of the meeting as a whole, as only the topic break annotations are visible without the clutter of the transcription.

Audio and Video The red vertical line on the right-hand-side of figure 5(a) is the audio and/or video cursor. It indicates the current position of playback: as playback proceeds, it moves from left to right and the track display is automatically scrolled. Buttons along the bottom can be used to pause playback, or skip forward and back a few seconds – allowing users to quickly replay a bit of the conversation, or quickly fast forward through parts of it. The *focus* button is used to center the display around the current media location; conversely clicking in a particular location in a track will move the cursor to that location. An arbitrary number of audio and video streams can be mixed (for instance: video plus audio for each participant).

Search A basic search capability is provided in the toolkit. Currently, a regular expression can be provided which will be matched against all of the topic names annotated in all corpora available. Clicking on the results, will load the identified conversation into the tool, and the track window will be shifted to show the particular topic in question. This sort of capability will be a core feature of the *meeting browser*, but it is included in the toolkit as it was useful for the annotators as well – especially during the early iterative phase in which they spent a lot of time discussing their annotations.

5 Tools

The multimodal discourse ontology, associated ontology programming interface, and audiovisual toolkit provide the basis for several tools. In sections 5.1 and 5.2 we describe tools built using this framework for performing and comparing annotations. Section 5.3 discusses a tool for visual feature analysis which we have used in preliminary automatic segmentation work. Finally, in section 5.4 we discuss preliminary work in creating a *meeting browser* – an end-user component of the CALO digital personal office assistant. Taken together, these tools demonstrate the flexibility of the architecture we have developed, showing how it can play a cross-cutting role across the tasks of meeting annotation, browsing, and research.

5.1 Annotation Tool: *NOMOS*

The annotation tool, *NOMOS*, is shown in figure 5(a). It leverages the full features of the audiovisual toolkit, complementing them with additional features designed to allow for actually annotating a discourse. The tool as described here is the result of a process of rapid iterative refinement which was coordinated with the period spent refining the annotation schema. We briefly note here features developed in the tool (as well as in the audiovisual toolkit) which particularly decrease the high cognitive load demanded by the annotation task. Notably, key capabilities revolve around simultaneously providing global and local insight into the meeting and annotations, as well as the capability to easily revise draft annotations.

Topic and action items are annotated by using the mouse to bring up context menus on the discourse, or by clicking buttons along the bottom of the display. A topic hierarchy (shown in the upper left of figure 5(a)) and a list of annotation items (shown in the upper right) shows the annotations at a global level. During the pilot period of annotation, it became clear how important it was to be able to modify annotations after making an initial rough pass through a discourse. As a result, capabilities for *renaming* and *deleting* both topics and action items exist, as well as the ability to *promote*, *demote*, or *merge* major and minor topics as appropriate. In addition, "reminders" can be inserted at particular time points, allowing annotators to make notes to refer back to in a subsequent pass.

The annotators found when working with the transcribed spoken corpora that there are both situations in which the transcriptions are critical and ones in which the audio itself is critical. For instance, sometimes detecting a topic shift seems to have a lot to do with the tone of voice which could only be detected through listening to the audio. At other times, for instance during a lengthy monologue on a single topic, it might suffice to skip quickly through the audio portion while skimming the transcriptions and looking for obvious pauses, speaker changes, disfluencies, or other cues. It was in response to this that the zooming capabilities of the audiovisual toolkit described above were developed, as well as the functionality described for efficiently skipping forward and back through the audio.

5.2 Annotation *Comparison Tool*

During our phase of iteratively refining the schema, it was quite important to be able to see each annotator's annotations of a single meeting side-by-side. This capability is implemented in a *Comparison Tool*. This tool does little above and beyond the basic capabilities provided by the audiovisual toolkit; it merely leverages these capabilities to graphically display several annotations for the same discourse stacked one above the other. Zooming out allowed the annotators to get a rough idea of where areas of disagreement and agreement were; these areas were then zoomed in on and discussed.

The comparison tool has also proved useful in comparing the annotations we've automatically generated using different machine learning techniques. Visually comparing similarities and differences lends powerful (though perhaps anecdotal) insight into differences among algorithms.

5.3 Feature Visualizer

We have developed a generic *Feature extractor* and *Feature Visualizer* using the ontology programming interface and audiovisual toolkit, as the architecture digram in figure 4 shows. We mean *feature* here in the sense of features which can be computed from discourse as input to machine learning algorithms for *e.g.* topic segmentation. The *Feature Extractor* is simply a set of Java classes which provide core functionality for processing discourse, as represented by the OPI. Functionalities include: extracting sets of utterances in a given time window, turning these utterances into bags of words per speaker, smoothing feature values, and calculating their derivatives. Moreover, generic tools are provided for iterating over discourses, processing them, and extracting sets of feature values at regular intervals which can then be piped directly into learners like decision trees or neural nets.

The *Feature Visualizer* is built on top of the extraction architecture and the audiovisual toolkit. It displays calculated feature values alongside an annotated discourse, as shown in figure 5(b). Moreover, as the popup window in figure 5(b) shows, it allows the user to dynamically modify each feature's parameters (for example: window size, smoothing, or other feature-specific parameters) and immediately observe the results. We have found the visualizer to be invaluable in debugging algorithms for feature extractors, tweaking parameter values, and hypothesizing new, interesting features.

5.4 Meeting Browser

We are currently developing a *Meeting Browser* tool, which will sit on top of both the audiovisual toolkit and the feature extractor. The eventual development of this tool is the motivation that has driven our annotations and associated schema. The browser is meant to allow users to "drill down" through the structure of the meeting, easily pinpointing segments of interest.

6 Current and Future Work

The work described in this paper represents our first steps toward automatic meeting understanding for a personal office assistant. While coarse-level meeting segmentation is a useful first step, we are tackling the problem from multiple angles including robust natural language chunk parsing, dialogue act detection, argumentation structure analysis, and decision detection. Our first steps in these areas will likely be similar to those we have taken in topic segmentation: establishing modular additions to the annotation ontology, supporting this in our audio-visual toolkit, coding annotation, research, and application tools for them, and then collecting annotations. Annotation of these richer structures will require use of the inference capabilities the ontology provides. For example, a tool designed for the annotation of argumentative structure will need to employ the constraints imposed by the ontology on that structure through the use of reasoning engines to constrain the annotations a human can make.

In parallel, we are currently developing an automatic topic segmenter, by training a classifier on the annotations presented above while using the presented software framework for feature extraction and visualization. Initial investigation following a roughly similar approach to (Galley et al., 2003) (using a decision tree trained on both lexical cohesion values and some discourse-based features – speaker activity, speaker overlap, amount of silence – and cross-validating over 25 ICSI meetings) has given average P_k error levels of around 0.35 for major topics. This is higher than Galley, et al. achieved on their segmentation, but this would be expected with our finer-grained and less restricted notion of topic, and is at least comparable to our mean human annotator agreement or 0.28. Future development will add prosodic features and chunk parser output. We also plan to expand our investigation into multimodal corpora currently being collected by our CALO partners. This will allow incorporation of features extracted from video and whiteboard interaction. We will also begin to use speech recognition hypotheses rather than transcriptions.

Lastly, we expect to use our audio-visual toolkit as a part of the CALO office assistant itself. This will involve the integration of our architecture with the CALO Desktop environment, allowing for pervasive feedback to our algorithms and online supervised learning.

Acknowledgments

Thanks to our two annotators Michael Deeringer and Claire Gilbert, to our CALO associates Satanjeev Banerjee and Bill

Jarrold, and to three anonymous reviewers. This work was supported by DARPA grant NBCH-D-03-0010. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study: Final report. In *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*.

Satanjeev Banerjee, Carolyn Rose, and Alex Rudnicky. 2005. The necessity of a meeting recording and playback system, and the benefit of topic–level annotations to meeting browsing. Submitted to INTERACT 2005.

Ken Barker, Bruce Porter, and Peter Clark. 2001. A library of generic concepts for composing knowledge bases. In *Proceedings of the First International Conference on Knowledge Capture*, October 21-23.

Doug Beeferman, Adam Berger, and John D. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American*, May.

Susanne Burger, Victoria MacLaren, and Hua Yu. 2002. The ISL meeting corpus: The impact of meeting type on speech style. In *Proceedings of the ICSLP 2002*, Denver, September.

Jean Carletta and Jonathan Kilgour. 2004. The NITE XML toolkit meets the ICSI meeting corpus: Import, annotation, browsing. In *Proceedings of the Joint AMI/PASCAL/IM2/M4 Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Martigny, Switzerland, June.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–255.

Freddy Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of NAACL-00*.

Barbara Di Eugenio and Michael Glass. 2004. The kappa statistic: A second look. *Computational Linguistics*, 30(1):95–101.

Scott Farrar. forthcoming. Using 'ontolinguistics' for language description. In *Ontolinguistics*. Mouton de Gruyter, Berlin.

Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, Sapporo, Japan, July.

Joost Geurts, Stefano Bocconi, Jacco van Ossenbruggen, and Lynda Hardman. 2003. Towards ontology-driven discourse: From semantic graphs to multimedia presentations. In *Second International Semantic Web Conference (ISWC2003)*, pages 597–612, Sanibel Island, Florida, Oct.

Alexander Gruenstein, John Niekrasz, Michael Deeringer, and Claire Gilbert, 2004. *Discourse annotation using Annotate!* http://cujo.stanford.edu/twiki/bin/view/Corpora/.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Meeting of the Association for Computational Linguistics*, Los Cruces, NM, June.

Marti Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Julia Hirschberg and Christine Nakatani. 1998. Acoustic indicators of topic segmentation. In *Proc. of ICSLP*.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The ICSI meeting corpus. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-03)*, pages 364–367.

Ed Kaiser, David Demirdjian, Alexander Gruenstein, Xiaoguang Li, John Niekrasz, Matt Wesson, and Sanjeev Kumar. 2004. A multimodal learning interface for sketch, speak and point creation of a schedule chart. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 329–330. ACM Press.

John Niekrasz and Matthew Purver. 2005. An ontology of multimodal discourse. In submission to MLMI '05, Edinburgh Scotland.

John Niekrasz, Matthew Purver, John Dowding, and Stanley Peters. 2005. Ontology-based discourse understanding for a persistent meeting assistant. In *Proceedings of the 2005 AAAI spring symposium on persistent assistants*, Stanford, March.

Rebecca J. Passonneau and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139.

Lev Pevzner and Marti Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

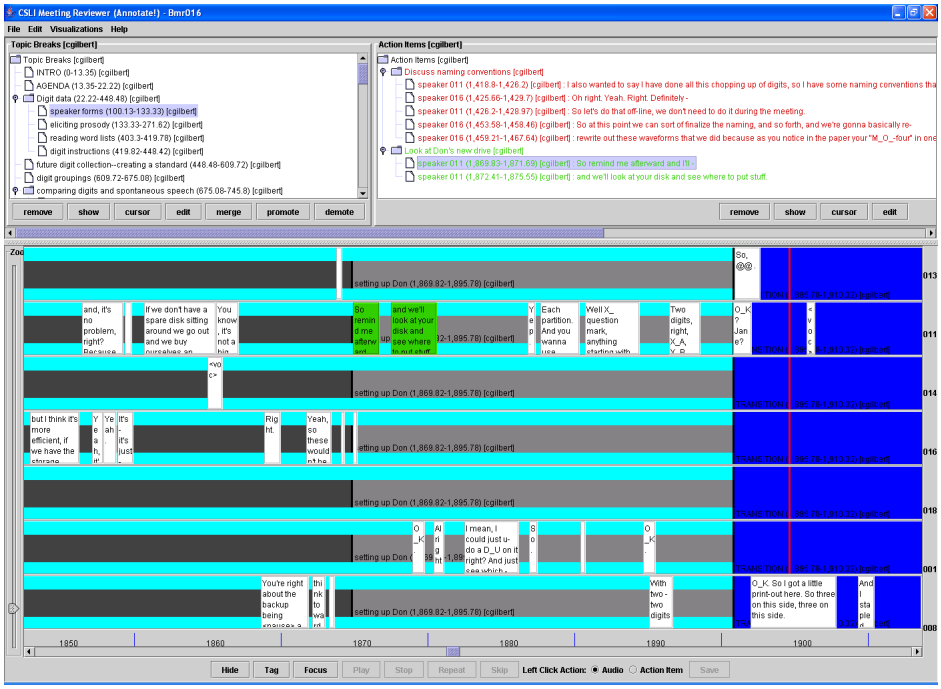
Klaus Ries. 2001. Segmenting conversations by topic, initiative and style. In *Proc. of ACM SIGIR Workshop on Information Retrieval Techniques for Speech Applications*.

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The ICSI meeting recorder dialog act (mrda) corpus. In *Proceedings of HLT-NAACL SIGDIAL Workshop*.

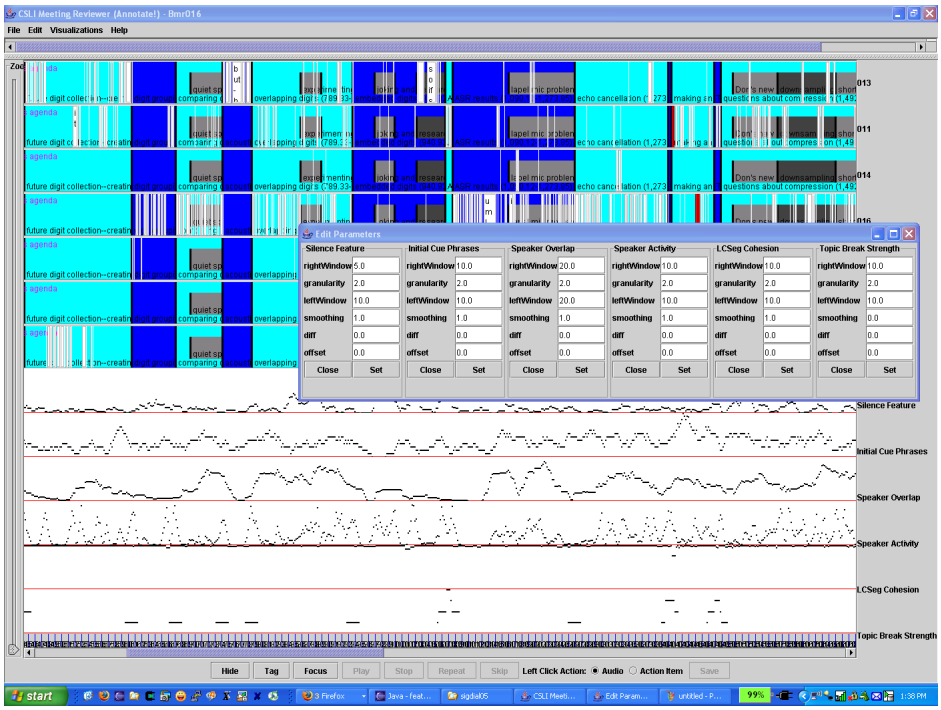
Sidney Siegel and Jr N. J. Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, 2nd edition.

Gökhan Tür, Dilek Hakkani-Tür, Andreas Stolcke, and Elizabeth Shriberg. 2001. Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics*, 27(1):31–57.

Britta Wrede and Elizabeth Shriberg. 2003. Spotting "hot spots" in meetings: Human judgements and prosodic cues. In *EUROSPEECH 2003*, Geneva, September.



(a) *NOMOS* – described in section 5.1



(b) Screenshot of the Feature Visualization Tool described in section 5.3

Figure 5: Tool screenshots. These are high resolution images; zooming in yields finer detail.