V-ROOM: A Virtual Meeting System with Intelligent Structured Summarisation

Anne E. James*, Antonios G. Nanos, Philip Thompson
Distributed Systems and Modelling (DSM) Research Group
Faculty of Engineering and Computing
Coventry University, Priory Street,
Coventry, CV1 5FB, U.K.

Abstract—With the growth of virtual organisations and multinational companies, virtual collaboration tasks are becoming more important for employees. This paper describes the development of a virtual meeting system, called V-ROOM. An exploration of facilities required in such a system has been conducted. The findings highlighted that intelligent systems are needed, especially since information that individuals have to know and process, is vast. The survey results showed that meeting summarisation is one of the most important new features that should be added to virtual meeting systems for enterprises. This paper highlights the innovative methods employed in V-ROOM to produce relevant meeting summaries. V-ROOM's approach is compared to other methods from the literature and it is shown how the use of meta-data provided by parts of the V-ROOM system can improve the quality of summaries produced.

Keywords—*virtual meetings; virtual collaboration; artificial intelligence; summarisation; meeting minutes*

1 Introduction

A survey by Doodle in 2009 which involved 2500 administrative and management staff from across Europe and the US (Doodle 2009) has shown that on average staff attend 7.1 meetings per week and that each meeting lasts on average 2.75 hours and has 7 participants attending. The survey also shows how technology is transforming the way we hold meetings. According to the survey in 2009, 29% of our meetings were web-based, 26% of meetings were face-to-face, 24% were conference calls, private meetings accounted for 11% while other modes were used for 10%. It is likely that the use of virtual conferencing has increased since then and will increase further in line with the increased connectivity we see in and between enterprises.

So virtual meetings and virtual collaboration are becoming commonplace and it is likely that virtual collaboration will grow (Martins, Gilson and Maynard 2004). It is useful to consider how such collaboration can be further supported. Previous research indicated that meeting facilitation leads to more effective meetings and can produce better outcomes (Adkins, Burgoon and Nunamaker 2003) and that facilitation is an important factor especially for electronic meetings (Rangarajan and Rohrbaugh 2003). Macaualy and Alabdulkarim (2005) outline some factors that can improve the productivity of a meeting. These are: reducing the amount of time that the staff spend in virtual meetings; having more informed sessions in order to make the meetings more effective; and agreeing upon outcomes with full commitment from all of the participants. Considering that the facilitation of virtual teams can be extremely difficult, it is important to develop new systems that will address the needs of the facilitators (Romano et al 1999). Complexity arises through organisational, human and technological factors. These include the size of teams, disparate locations, time zones, lack of facial cues, differing cultures, commitment expectations, understanding and technology. As well as to improve the communication among participants, the role of the facilitator is to assist a group to make the right decision (Griffith, Fuller and Northcraft 1998). When a team of geographically dispersed members needs to communicate and collaborate, a system has to facilitate and assist those tasks. Moreover, successful meetings need to be well structured in order to ensure that the participants stay on subject and to avoid frustration (Hayne 1999). Wang, Haake and Rubart (2004) mentioned five support services that would lead to a successful meeting. These were: well-structured agendas; ensuring that participants attend the right session; a space for sharing information;

meeting moderation; and document sharing. At Coventry University we have developed a virtual meeting system called V-ROOM that provides those services. V-ROOM makes use of agendas of timed items that are presented during the meeting, only invited participants are allowed to access the meeting space, a facilitator is in charge of moderating the meeting and a document uploading feature can be used for sharing purposes. V-ROOM makes use of meta-data such as agendas, roles and meeting structure to facilitate the generation of meeting minutes. Furthermore, V-ROOM enables meeting minutes to be generated in the context of varying levels of user support.

This paper provides some background to V-ROOM and explains how summarisation is carried out to produce meeting minutes. The paper is organised as follows. Section 2 discusses the development of V-ROOM, outlining the motivation and the methodology being used. Section 3 provides an overview on related work on summarisation. Section 4 describes in detail the methods used in V-ROOM for summarising. Section 5 discusses our results and highlights the innovation compared with other systems. Finally section 6 concludes and describes our future work.

2 Development of V-ROOM

Our motivation is to improve the support provided in virtual meeting systems through the use of artificial intelligence techniques. Artificial intelligence systems often need to process masses of data and may have infinite performance possibilities. It is therefore generally impossible to test all scenarios in a particular application (Straub and Huber 2013). We take the view of using artificial intelligence as a support for facilitators rather than a replacement unless in situations where security and reliability can be assured. The system we have developed consists of three parts which are the pre-meeting tasks, the live meeting support and the post-meeting tasks. The target is to decrease the time that people spend in order to prepare and create a virtual meeting, and to minimise the effort when they work on the post-meeting tasks. During the live meeting, support is provided in the form of a running agenda with timed items, intelligent document retrieval and chat. The V-ROOM system provides three main additional features not provided by the commercial systems. These are: meeting roles; agenda setting and support; and minute generation. The agenda consists of items and timings. Further features can be added such as specifying the item leader. The use of timed items enables a strong structure to be imposed on the meeting. V-ROOM differs from other research systems in the provision of the structured timed agenda and generation of meeting minutes. An ideal system should be capable of providing the facilitator with information relating to an agenda item, proposing experts in order to resolve a problem and performing tasks after the end of the meeting such as producing the meeting minutes. With the advantage of finding experts for resolving a problem, the facilitator will be able to create the meeting in less time. When participants need information, the system should research on behalf of them because it is very common, for instance, for people to use Google while chatting. If the system could automatically search required items, loss of attention will be restricted. Another useful feature would be to integrate the system with other systems such as DropBox for file sharing, e-mail systems, or to allow social media integration - depending on the privacy concerns of a meeting.

V-ROOM has been developed and evaluated in stages. Initially an agent based system was developed to test novel architecture and communication concepts (Thompson, James and Stanciu 2010). Then a focus group was established to determine the features that a virtual meeting system should include (Thompson, James and Iqbal 2011). In order to develop the system, our investigation focused on understanding the role of the facilitator and evaluating the possible features that would facilitate a virtual meeting session. The focus group research resulted in the findings that the prototype functionality would include standard facilities seen in other virtual meetings and also the following novel features: a meeting ontology would be maintained for the execution of the meeting; synchronous and asynchronous mode of interaction would be supported; minutes from the meeting would be generated; and an agenda-based polling facility would be implemented. Having established the requirements, a prototype system was developed and evaluated (Thompson, James and Nanos 2013). The evaluation revealed that the following V-ROOM features were considered to be good: meeting minutes; uploading documents; polling on agenda items; meeting structure; remote collaboration; security; invitations and asynchronous mode. The evaluation revealed that other features considered important to have in virtual meeting systems are: audio/visual support; document sharing; desktop sharing; on-line whiteboard support; mobile phone interface; list of participants present; use of emoticons; instant messaging between participants; time agenda items; and personalisation of the environment. These features are considered part of the established standard V-ROOM. Then we entered the next phase of development which was to add more artificial intelligence to the prototype in order to provide more advanced support. In order to decide

what features to include we conducted a survey of professional virtual meeting system users to establish what features they thought would be most useful. We suggested in the survey that the following might be useful: meeting minutes summarisation; voice recognition; expert suggestion; sentiment analysis; opinion mining; gestures recognition; and text generation. We placed the respondents into two groups, senior professionals and junior professionals. Our survey results showed that meeting summarisation was regarded as the most needed future feature (see Figure 1). Full results of the survey can be found in our previous publication (James and Nanos 2013).

Following the results of our survey we have developed an advanced summarisation feature for V-ROOM which includes varying levels of support and the ability to select various types of summary. More detail is provided on our methods for summarisation in Section 4.

3 Related Work

3.1 Improved Support for Virtual Collaboration

Researchers have investigated what features can be introduced to make virtual meeting support software more effective and useful. Approaches that have been explored to provide improved support include the following: developing intelligent agents to help participants and facilitators; multimedia recording and storing of meetings with subsequent development of effective browsers and annotation so that meeting records can be easily found; effectiveness of virtual facilitation; building appropriate models; gathering data for feedback on meeting participation; extensible and integrated groupware; and summarisation of meetings. We can consider some examples of this related work.

In previous research from Coventry University (Thompson, James and Stanciu 2010) a software agent has been developed to assist the human facilitator of the meeting. A meeting ontology was achieved through the use of a graph database which, used of a triples datastore to help identify the terminology used in the meeting environment, the relationship between objects reflected in those words and the properties of the objects. Agents as personal assistants have also been used in other research (Huhns and Singh 1998; Campagnolo et al,2009; Barthès 2011; Rahman et al 2012; Sugawar, Manabe and Fujita 2012; Paraiso and Malucelli 2012). To make such applications succeed, participants have to use specific vocabulary that can be matched in an ontology. Results have been quite effective. The way our research differs from these others is that a detailed analysis has been carried out into the ontology and protocol representation needed   specifically to support meeting environments. The construct of an agenda is used as a foundation. Other agent work has not specifically addressed the meeting environment.

MeetingManager (Oh,Tuchina and Wu 2001) developed at the MIT Artificial Intelligence Laboratory, has been described as a multi-user, multi-modal collaboration tool for planning, facilitating, and browsing structured meetings. MeetingManager takes advantage of the natural structure of the meeting that many organizations practice for an efficient meeting process. MeetingManager worked in the context of an intelligent room were sensors were used to record many aspects of the meeting. Participants were using a meeting planner feature of the system to create the agenda and the date of the meeting. During the meeting the agenda is presented by the facilitator who was in charge of starting the meeting video recording. At the end of the meeting, the system produced a record of the meeting and sent it to the participants by email. The record was limited to items that could be triggered from the structure of the meeting and included links to recorded sections. Another form of facilitation is to develop software entities that can interact with the participants when they are asked particular queries. Our research has similarities to this work in that we exploit the structure of a meeting. However we use the structure for summarising and developing the meeting minutes, an area which MeetingManager did not address.

Researchers from the University of La Verne explored the role of virtual facilitation using a virtual facilitation simulation tool (Wiggens et al 2011; Colbert et al 2011). A virtual facilitator monitors a conversation and intervenes when it believes that a participant does not provide strong enough information to the conversation or when the participants use specific keywords. Keywords or keyword endings were used to trigger the virtual facilitator questions. The researchers were testing whether decision making and conflict resolution could be improved through the use of a virtual facilitator. They found that these aspects were not improved and concluded that more research was needed.

However other previous research has supported the facilitator role (Rangarajan and Rohrbaugh 2003). It is noted however that in the research of Wiggens et al (2001) and Colbert et al (2011), the facilitator was seen in the function of drawing ideas out of people and persuading them to reach conclusions. In other research the facilitator is seen to have a wider brief and not necessarily that of encourager and persuader (Romano et al 2009; Griffith, Fuller and Northcraft 1998). It should also be noted that it is likely that it was the method that was flawed rather than the overall concept, hence the need for further investigation. The lesson to note here is that the research findings will differ according to the emphasis placed on a particular role function as well as the method used and so must be seen in context. In V-ROOM we have represented a number of roles (see Section 4) and protocols of interaction around those roles. We have taken note of previous research by enabling V-ROOM to be extensible so that new roles and protocols can be added according to the culture of the enterprise. We do not solely concentrate on the role of the facilitator.

Other research uses similar methods to provide structure and hence further automation to support virtual meetings. In some earlier work, roles, resources and process were appropriately proposed as the three main categories from which to build a model for groupware (Atunes and Carrico 2005). Some systems additionally capture specific business and management methodologies. In Gruenbacher and Briggs (2001) the system uses a specific methodology, EasyWinWin, for facilitating the tasks between the distributed participants. Conflicts can occur between the interests of the organisation and that of a team member. Justifying and making vision and requirements explicit through collaboration can ameliorate such undesirable effects. The idea is that teams use EasyWinWin throughout the development cycle to develop an acceptable set of system requirements. Additionally, the research classified three tasks that would be used for group facilitation: assisting the users on how to use the system; motivating the users by keeping a smooth conversation or giving procedural instructions as well as guiding users in behavioural matters as long as participants can not see each other; and enhancing user participation when needed in meeting phases. Similarly, continuing from previous work, Roman, Mubin and Dillenbourg (2012) developed a client-server meeting system that can give immediate feedback to users about their participation. Users of this system can use various devices to get information about a past or current meeting, mostly in form of diagrams. Also, their system can extract measurable parameters in standard formats such as XML or CSV, ready for use in statistical processing packages such as SPSS. PowerMeeting (Wang 2010) is a system that also supports groupware settings. It supports decision making, brainstorming, some roles, agenda items and a voting tool with a multi-criteria decision support tool. In addition, the system's extensibility is granted by allowing users to develop and integrate tools that they wish to use for their purposes of collaboration. It also includes a Chat and Skype facility. Guerrro and Pino (2008) developed a system to support discussion of issues asynchronously. Contributors can mark their contributions in various ways which will help clarify decision influencers including *question*, *suggestion*, *answer*, *argument in favour* or *argument against*. In this way the system is performing a facilitating role and helping to prepare users better for final decision making. Our research differs from those others described here in that, as well as providing various facilities to improve support and building on previous work to do this, we have developed a system which offers summarisation and the generation of meeting minutes.

3.2 Summarisation

It is clear to see that the amount of text conversations generated and captured has grown enormously over the last decades as a result of email, social networking and collaborative software. Good summarisation techniques will enable such text to be used productively in a variety of contexts. An excellent overview of text conversation summarisation is given in a Carenini, Murray and Ng (2011). Our research focusses on meeting summarisation, or more specifically the assisted or automatic generation of meeting minutes. Meeting minutes are a well-recognised artefact for recording the main points of a meeting for others to see, to enable appropriate business decisions to be made and to ensure business procedures are followed. However recording minutes can be burdensome and it has been recognised that automatic summarisation of meetings would be very desirable in the context of a virtual meeting system (Nanos and James 2011). A virtual meeting system can quite easily capture a transcript of the meeting either in the form of captured chat or recorded speech. If the meeting is captured as recorded speech, an Automatic Speech Recognition (ASR) system can be used to generate the text transcript. Such systems however have high word error rates so this can hamper the quality of the transcript. Similarly chat transcripts can have many word errors. It has nevertheless been stated that summarisation systems can still operate efficiently in spite of word errors (Carenini, Murray and Ng 2011). However it is recommended that raw transcripts are pre-processed before entering a summarisation system to remove as many errors as possible.

There are various types of summarisation techniques. Furthermore, summarisation systems may contain combinations of techniques. Popular techniques include topic modelling, opinion modelling and extracting conversational structure. Other characterisations of summarisation include single vs. multiple object summarisation, extractive versus abstractive summary, generic vs. query based summaries and indicative vs. informative summarisation (Carenini, Murray and Ng 2011). We provide an explanation on these in the next paragraphs and illustrate some of them in the description of the V-ROOM summarisation method in section 4.

A very important technique is topic modelling where distinct topics are recognised within the transcript. Topics are recognised by picking out keywords and their synonyms that occur frequently within a text cluster. To detect a change of topic, systems compare one sentence with the next and generate a probability on whether a topic has changed or not depending on commonality of word or word phrasing between the two sentences. Other lexical or structural clues can also be used to determine whether a topic has changed.

Opinion modelling has gained much publicity lately in the guise of sentiment analysis, a technique very popular in marketing and politics for gauging public opinion. Various techniques have been proposed for performing opinion modelling (Pang and Lee 2008). Lexical and syntactic features of the phrases are used. For instance recognising positive versus negative words or word phrases, like "I like", I love", "I hate", "that's good", "that's horrible" in the context of some object of interest can enable an opinion to be classified as positive or negative. Then by counting negatives versus positives across messages in a sample one obtain an opinion ratio. Various subtleties of language construction and vocabulary make this task challenging and have to be taken into account.

Extracting the conversational structure is also an important technique for summarisation. Knowing how contributions can be grouped in conversational structure such as question-answer pairs or what sort of role a contribution is playing in a conversation can be very useful. For instance a contribution could be a question, an answer, a description, a request, an acceptance of a request, a proposal, a decision, an agreement or other performative. Some researchers have used structure to help in a supervised approach to summarisation (Cohen, Carvolho and Mitchell 2004; Carvalho and Cohen 2005; Shresta and McKeown 2004). However there is no commonly accepted standard classification for dialogue acts. Jeong et al in 2009 used the following classifications in a semi-supervised approach: *statement*; *polite mechanism*; *yes-no question*; *action motivator*; *Wh- question*; *accept response*; *open-ended question*; *acknowledge* and *appreciate*; *or/or clause question*; *reject response*; *uncertain response*; *rhetorical question*. Many approaches to dialogue act modelling have been supervised approaches. Ritter, Cherry and Dolan (2010) however used an unsupervised or minimally supervised approach to dialogue act modelling. Instead of labelling each contribution, they cluster contributions according to conversational function. They used Twitter conversations in their experiment. They identified an underlying model for Twitter conversations, namely that a Twitter conversation starts in one of the three following ways: *Status*; *Reference Broadcast*; or *Question to Followers*. *Status* just describes what the subject is doing; *Reference Broadcast* provides a link to something else and *Question to Followers* just asks the followers a question. Each of these acts may then be followed which other dialogue acts. For instance, a *Reference Broadcast* may be followed by one or more *Reactions.* A *Status* may be followed by one or more Comments or a Question. The interesting part of this work is that it is an attempt to capture typical dialogue patterns. In a meeting scenario, the capture of protocols in the form of dialogue patterns is helpful in capturing meaning.

Other researchers have investigated the construction of summaries based on specific dialogue acts such as decisions and actions. Hsueh and Moore (2007) adopted a supervised approach to detecting decisions using lexical, prosodic, topical and contextual features. They predicted features at utterance and topic segment level. Fernadez et al (2008) worked at a finer grained level identifying different types of decision component under the labels of *issue*, *resolution* and *agreement*. Under *resolution* there are two components: *proposal* and *restatement*. A dialogue act of *agreemen*t was also recognised. Murray and Renals (2008) trained a classifier using prosodic, lexical, structural, length and speaker features. Purver , Ehlen and Niekrasz (2006) adopted a finer-grained approach and identified subclasses of action items, namely *description*, *time-frame*, *owner* and *agreement*. In our research we develop a summarisation method based around the structure of a meeting (see section 4) which classifies dialogue acts in a meeting context. Such detailed tailoring to the context of virtual meetings has not been addressed by other researchers.

Other characteristics of summarisation such as single versus multiple object summarisation, extractive versus abstractive summary, generic versus query-based summaries and indicative versus informative summarisation also need to be taken into consideration by researchers (Carenini, Murray and Ng 2011). Single versus multiple object refers to whether a summarisation is done on the basis of a single document, utterance, message of contribution or whether the summarisation is carried out in the context of multiple objects. Extractive versus abstractive summarisation refers to whether whole sentences or phrases are extracted unaltered from the transcripts to form the summary or whether some processing is carried out on these to present the meaning in a different and perhaps more digestible way. Indicative versus informative summarisation refers to whether an attempt is made to convey the actual content in the summary or whether just a high level outline which would provide an indication to the reader on whether it would be interesting to look at the full record to find out more. Other characteristics include: domain-specific versus general-purpose; and textual versus multi-media output. The latter refers to whether the output summary is in text form or in some other media (e.g. audio or video) or whether it constitutes a combination of media. With regard to our research we work with multiple objects in that messages are analysed in the context of surrounding messages. We adopt a hybrid approach of extractive and abstractive. Raw relevant sentences are selected but an attempt is made to improve them to the contextual style of the summary and to provide more information. Our summaries consist of a combination of indicative and informative content depending on confidence on the identified meaning. Our system is domain specific in that we stick to the context of meetings and transcriptions thereof. The output of our system is textual.

Most of the research in summarisation has involved sentence or partial sentence extraction. Statistical techniques are also used, for instance comparing the frequency of terms. However results are far from perfect and there is room for more effective approaches. Lloret and Palomar (2012) state that the rapid development of emerging technologies poses new challenges in this research field. Saggion and Poibeau (2013) consider that although research into automatic text summarisation is over 50 years old, more effort is needed given the new types of requirements. The researchers discuss "domain-specific summaries", "opinion-oriented summaries" and "update summaries". Spärck Jones (2007) observed that we do not know how to automate summarisation without heavy application-specific guidance. She concluded that automatic summarisation has made valuable progress in the decade to 2005, with useful applications, better evaluation and more task understanding but systems still need to better exploit the purpose of the summary and context of use. The system we have developed in V-ROOM allows for different types of summary to be generated.

The research work described above is based mostly on domain-specific, structured, underlying models of the collaboration and ontologies or keywords or key parts of words. It is very difficult to develop intelligent agents that can understand in depth the human language and perform actions on behalf of humans without the use of such structures. We can conclude that to achieve effective intelligent virtual assistance, the virtual assistant has to have foundation intelligence in the form of knowledge of the application, context and structure, just as a human facilitator or minutes writer is more effective through understanding of the business and the culture. Meetings can be of various type, each with its own characteristics, including structure, vocabulary and interaction patterns. Thus capturing the intelligence to support every type of meeting is a huge task which continues as different types of virtual collaboration emerge. An efficient methodology for capturing the foundation intelligence needed for a particular class of meetings together with some generic patterns would be a useful contribution to the field, leaving scope for many research and development projects. This way new domain databases could be grown in a scalable fashion.

4 The V-ROOM Approach to Summarisation

4.1 High-level Approach

Artificial intelligence cannot fully replace the intelligent insight of humans but can certainly assist in many human tasks. In V-ROOM, as explained in section 2, we are exploring the use of various methods of artificial intelligence with the aim of improving the functionality of virtual meeting systems. One area we have addressed is summarisation. Bearing in mind the limits of complete automation, we developed a two level approach to summarisation, assisted and automatic. The system works as follows. A meeting takes place according to a pre-set agenda. A transcript record of the meeting is generated which shows the agenda and provides a record of the

meeting with timings. This transcript is input to the summariser to produce the meeting minutes. The user has the choice to use assisted summarisation or automatic summarisation. If the user selects assisted, the transcript appears in a browser window and the user can select various tools to use for annotation. After annotation, the meeting minutes can be generated. If the user selects automatic generation, the minutes are generated and then appear in a browser window alongside the original transcript. The user can then choose to correct the minutes or accept them as automatically generated. Figure 2 shows the overall architecture of the summarisation component of V-ROOM. Note that there are options with the system for completely assisted, completely automatic or hybrid modes of operation.

Both assisted and automatic summarisation relies heavily on a recognised structure of a meeting. This approach is similar to that of the use of conversation structure recognised by other researchers and discussed earlier. However, as far as we are aware, no other research has used meeting structure as holistically as we have to produce a relevant summary. We have developed a model of a typical meeting and use the constructs identified to guide the production of the summary. In this section we first describe our meeting model. We then explain how this is used to guide summarisation in both assisted and automatic modes.

4.2 Meeting Model

Our meeting model centres around the constructs of *role*, *agenda* and *meeting type*. The main meeting roles are *chair*, *participant*, *item leader*, and *facilitator*. The role of facilitator is to set up the meeting, making sure everything is ready such as agenda set, documents uploaded and invitations made. It is not necessary for all meetings to have all roles represented and a person could have more than one role at the same time, for instance the chair could also be the facilitator or a participant may be an item leader on one or items. The idea of an item leader is to recognise that a participant who acts as an item leader might introduce and speak about an item on the agenda. This participant would typically be someone who has been deeply involved with the item in question and so is the most knowledgeable about it. The meeting, once it has taken place, will have a transcript, namely a record of the exact dialogue produced either through recording the chat or recording speech and translating to a transcript through ASR.

An agenda is made up of the items that need to be discussed at the meeting, the participant leading on each item and estimate timings for each item. Thus an *agenda* has *items*. When the agenda is set up each item can be given a *start time*, *end time* and *item leader*. This will help structure the meeting. Items have other characteristics. They have a *transcript*. This is the part of the meeting transcript which relates to the item. The item transcripts are contained within the transcript for the complete meeting but it is not an easy task to automatically extract item transcripts because item boundaries first have to be identified within the transcript of the complete meeting. Natural language processing and text mining techniques are used to achieve this. The defined agenda and associated items also help with this task. Items also have a *descriptive content* which is an overall description of what the item is about. To add more meaning other characteristics have been identified. An *item* will usually require an *outcome*. An outcome could be an *acknowledgement* or *acceptance*. For instance at a meeting a report might be presented and there will be a requirement that the participants accept the report (or dismiss it with reasons). Another type of outcome might be an *action* or set of *actions*. An action would be something that someone has to do and thus an action would have a *description*, an *owner* and a timeframe, which can be represented as a *start-time* and an *end-time*. Sometimes it can be useful to recognise that an item also may contain one or more issues for discussion. An *issue* in turn may require discussion of alternative solutions and then a decision or action plan. The way we model this is for an *issue* to have a *descriptive content* and one or more *propositions*. A *proposition* has a *proposal* and an *outcome*. It might also have a *poll*. The outcome can be an *agreement* or a *non-agreement* and/or an *action* or set of *actions*. An *issue* may also have an *outcome* which summarises what decision was made after the proposals were considered. It may also have one or more actions. In fact actions and outcomes can be represented at the granularity of *item*, *issue* or *proposal*. Figures 3 and 4 show our model for meetings in diagramatic form. Figure 3 shows the overall structure and Figure 4 shows some more detail on the *outcome* construct. It is this model that drives the innovation in both assisted and automatic mode for summarisation in V-ROOM.

Meetings can be of various type each with its own structure and protocol. We have initially considered committee meetings and project meetings in the higher education context. Project meetings cover various conferences from

large consortia meetings to small supervisor-student project meetings. Capturing distinguishing characteristics of meeting types can aid in summarisation.

4.3 Use of the Model in Assisted Mode

Assisted mode provides the user with a browser window and annotation tools. The meeting transcript is presented in the browser window. Then a user can quickly mark which parts of the transcript relate to which agenda items. Within each item, the user will also mark the issues.   Similarly proposals can be marked up as well as agreements, dismissals and actions. In fact all constructs provided in the model can be used to mark up the transcript. When the transcript mark-up is complete, the user can choose to write the summarisation or minutes her/himself or can choose to submit for automatic generation in the knowledge that the correct annotation has been done and thus making a better automatic generation more likely. Figure 5 shows part of the user interface for assisted mode.

4.4 Use of the Model in Automatic Mode

4.4.1 Identifying Items

In automatic mode the system uses information provided in the agenda to make the best job of summarisation. An agenda which will have been set before the meeting consists of items and estimated timings for the items and the lead participant for each item. These characteristics can be very helpful for intelligent generation of the minutes. An example agenda is shown below in Extract 1. AOB stands for "Any Other Business" and is a standard item on most agendas. The agenda would have been set up by whoever called the meeting and thus acts as facilitator. Figure 6 shows three screens for setting up a meeting in V-ROOM.

Meeting Title: Update Meeting
(Date: 4-07-2012, Start: 15:00, Finish:15:30)

Item for discussion: Progress update (15 minutes)
Item for discussion: Expenses update (10 minutes)
Item for discussion: AOB (5 minutes)

Attending: JJ; PP

Extract 1. A V-ROOM agenda

The agenda shown in Extract 1 happened to be for a meeting between a research student and supervisor and thus in this case lead participants were not specified. Notice that estimate timings are given for items. At a first level, the timing structure can be used to generate minutes for items. The stated duration for each item in the agenda can be checked the timings in the transcript. This leads to the automatic suggestion of specific item boundaries which in turn leads to an annotated transcript as shown in Figure 7. The full transcript of the meeting is given in Appendix A. It must be noted that basing the minute generation around timings is not failsafe as, in spite of support being given for timings in V-ROOM, meeting attendees might overshoot allocated item time. V-ROOM reminds participants but does not force them to stick to agenda timings. Improvements can be made, during minutes generation, by adjusting item time boundaries proportionately according to whether a meeting has taken longer or less time than expected. Other intelligence that is used to identify item boundaries is the use of keywords. The agenda can also support this since it includes item names. The item names are used as keywords together with a related item ontology that identifies words which might commonly occur in conversations about the specific items. The keyword technique is used in conjunction with the timing approach. Table 1 shows part of an item keyword relationship ontology which includes items of the agenda. This ontology shows words and phrases which might be used when discussing the item keyword. V-ROOM provides an item keyword relationship ontology for common meeting item terms and also provides the facility for users to extend ontologies, thus the culture and internal dialect of the organisation can gradually be built up. Keywords are counted to determine item boundaries. For instance if a keyword "progress" or its related words occur frequently in a sequence of dialogue then the system can infer that that sequence of dialogue relates to

the *Progress Update* item of the meeting. Sometimes there are lexical clues that relate directly to the agenda structure. For instance the chair might say something like "Now Item X" or "Moving on to Item X" (see line 37 in Appendix A). A structural ontology provides typical lexical constructs that indicate identification of structural aspects of a meeting. Table 2 illustrates part of the V-ROOM structural ontology. The final agenda characteristic is the specification of item leaders. Item leaders are participants that lead on a particular item of the agenda. In the case of the example there are no lead participants identified. However in a larger meeting where lead participants are identified, vital clues can be provided as to the topic boundaries. The transcript tells us who made which contributions and when. The conversation clustered around a lead participant is likely to be about the item that the lead participant is leading. This is especially true if the timings also correspond, that is if the real-life timings match the estimates provided by the agenda. Thus the use of timings, keywords and participant roles gives vital intelligence on item boundaries and using these together can give probabilities for boundary suggestions. Once the items are identified outline minutes can be produced as shown in Extract 2.

Table 1. Item keyword relationship ontology

Item Keyword	Related Words and Phrases
…	*…*
Expenses	Cost, Spend, Claim, How much? Receipts, Form, *Actual Money Figures*
…	*…*
Progress	Project, Deadline, How is it going ? Work, *Actual Dates*
…	*…*

Table 2. Structural Ontology

Role	Utterance	Structure
Chair	Moving on to …	Change of Item
Chair	Next…	Change of Item
Chair	Item …	Change of Item
Any	Anyway…	Slight change of direction
Any	OK	Agreement
Chair	OK…..	Indication to stop discussion and summarise
Chair	Thank you.	Indication to stop discussion and summarise

Minutes
Update Meeting

The meeting took place on 2012-07-11 15:00:00

Attending: JJ, PP

Agenda
Meeting Title: update meeting
Item for discussion: Progress update
Item for discussion: Expenses update
Item for discussion: AOB

Content

The following discussions occurred:

1. Progress update
   ……Extracts from relevant part of transcript.……

2. Expenses update
   ……Extracts from relevant part of transcript.……

3. AoB
   ……Extracts from relevant part of transcript.……

Extract 2. Output structure based on items

Appendix A shows, for the case of our sample meeting, how the contributions have been categorized into items. Lexical clues such as "How is the Project going" (lines 5) and "Ok then. Let's move onto teh Expense item." (line 37) were important lexical clues as to categorising the Items. The agenda with item keywords and timings was also used. Once the relevant transcript segments have been associated to the various agenda items, the next stage is to abstract relevant sentences for each item and also to identify characteristics such as *issue* and *outcome*.

4.4.2 Extracting Sentences

To select which sentences to extract we use roles, protocol of the meeting, and frequency of key words. The use of roles gives us the intelligence to add weight to a participant's contribution. Usually the chair of a meeting will keep a meeting on track and summarise decisions and actions. Thus we can add extra weight to the chair's contribution when deciding how to summarise. Similarly the lead participant for an item is likely to make a good summary of the issues involved and perhaps make proposals. We recognize that relevant content can also come from other participants so the use of roles is just one of the inputs into our formula. Word frequency is also used to aid identification of issues and potential sentences for extraction. Our summaristion process is shown in Figure 8.

4.4.2.1 Word Frequency and Topics

After ignoring stop words like "the", "a", "and", "for" and others, our summarization system makes a count of words used in the transcript for a meeting. Frequency clusters of words identify topics and give a basis for making a summary. For instance consider the part of the transcript in Appendix A which relates to the *Progress Update* item of our meeting (lines 5-36). In lines 10 -15 "GUI" occurs three times and in lines 17- 20 "ROSJava" occurs three times. This gives us a clue that GUI and ROSJava are issues and surrounding sentences can be clustered accordingly. If the system had the intelligence that ROS might be linked to ROSJava then lines 17-25 might have been clustered. Our summarisation system comes up with the clusters shown in Table 3.

Table 3. Topic clusters

lines	Issue/Topic	Occurrences
11-15	GUI	3
12-17	Compiling/Compile	4
14-20	ROSJAVA	4
19-22	packages	3
29-35	Android	3

4.4.2.2 Identify Descriptions, Propositions and Outcomes

Next the summarization system determines which parts of the transcript are descriptions, which are propositions and which are outcomes. It does this by first identifying proposals. Keywords that indicate proposals are "could", "should", "would", "perhaps", "maybe". The system comes up with the results shown in Table 4.

Table 4. Identifying proposals

lines	Proposal
29	in the mean time, *I could* look at android and the cloud or something mobile computing related
32	perhaps then *you should* write a brief report on your findings regarding ROS and ROS Java and the simulation. Then think of an Android type application that you could develop
34	If you can link it somehow to Clem it *would* be good. Maybe make some sort of quiz infrastructure where teachers can input multichoice questions and tehre quizzes can be accessed ia Android.

Next the system looks for outcome sentences. To do this, the systems concentrates towards the end of the item and looks for keywords like "will" or "OK" as agreements to suggestions. It also looks for a protocol of *proposal* by Supervisor and *agreement* by Student or vice versa as these are a typical protocol exchanges in a Supervisor–Student project meeting. The summarization system find that lines 29-32 and 32-36 of the transcript constitute proposal-agreement pairs. Parts of the transcript not identified as *proposal* or *outcome* are identified as *description.*

4.4.2.3 Identify Issues

Next the system identifies issues by taking or grouping topics from the *description* part of the transcript rather than the *proposal* part. This is because proposals are intended to resolve issues already identified. From the list shown in Table 3, the system recognizes the overlaps in the line ranges for keywords and clusters the potential issues together as "GUI/Compiling/ROSJava/packages". Thus the *Progress Update* item becomes a one-issue item. "Android" is not included as this is mentioned in the proposal part of the dialogue.

Following the identification of the various features, the minutes for the items can be generated.   Given that issues have been identified, an indicative summary could be generated such as

"The following issues were discussed: issue1; issue2; … and issueN."

or if only one issue

"The following issue was discussed: issue1; issue2; … and issueN."

In the case of our example

"The following issue was discussed: GUI/Compiling/ROSJava/packages."

If we wish to have a more informative summary we need to extract relevant sentences. Various meeting types have their own policy sets. The overall policy currently adopted by our summarization system is:

- take a subset of sentences from *descriptions*
- take all sentences from the *proposals*
- take agreed *proposals* and *actions* from *outcomes*

For a Supervisor-Student meeting, the addition protocol policy is also adopted

- take all *proposal-agreement* pairs

4.4.2.4 Extract or Compose Sentences

So considering the first part of the policy "take a subset of sentences from *descriptions",* the question arises as how to select sentences from the *description* part. Various policies could be adopted. Our system takes the first and last sentences of *description* part of a transcript if these sentences contain an *issue* keyword. The rationale behind this is that the first sentence usually sets the issue out and the last might attempt to sum up. In the case of issues with multiple keywords like "GUI/Compiling/ROSJAVA", the first sentence is extracted for each of the keywords. This is so that a reasonably full sense of the issue is conveyed. Our summarization system then wraps the extracted sentences to provide more context and to enable better comprehension for the reader, for instance instead of directly stating,

"PP- 3:13 PM Jul, 04 2012: So I've started the programming aspect in eclipse" (line 10)

the system generates,

"PP said that he has started the programming aspect in eclipse".

The V-ROOM summarization system also drops some sub-clauses of sentences to enhance conciseness.

4.4.2.4 Generate Minutes

Having completed the preceding steps, for the *Progress Update* part of the meeting the V-ROOM summarization system generates the minutes shown in Extract 3 .

---

1. Progress Update

JJ asked how things are going with the project. PP said that he has started the programming aspect in eclipse.

Issue: GUI/Compiling/ROSJava/packages

PP said the GUI for the app has already been built but he is having trouble compiling the code since it seems that the android library is not being referenced properly. JJ asked if the GUI controls the Robot simulation yet. PP answered, No, the GUI at the moment is simply a screen with the seek bars for adjustment and text labels. PP said, even the ROSjava tutorials are not compiling properly. JJ stated that it does seem quite a problem if the ROSjava tutorials are not compiling and asked PP if he thinks it is worth pursuing? PP answered that in his opinion at the moment it is not worth pursuing as a viable option but in the future it will definitely be an option. JJ asked if PP could think of something else he would rather do in his time here.

Proposal

PP answered that he could look at android and the cloud or something mobile computing related. JJ agreed

Proposal

* JJ suggested PP should write a brief report on his findings regarding ROS and ROSJava and the simulation. Then think of an Android type application that he could develop.*

Action
PP said he can definitely write up some documentation, and will try to come up with a new proposal by this weekend

Proposal
JJ suggested he can link it somehow to Clem. PP agreed. He will look into it

Outcome

PP will write up some documentation, and will try to come up with a new proposal by this weekend. PP will look at android and the cloud or something mobile computing related.PP will look into linking it with CLEM

Extract 3: Part of the V-ROOM generated meeting minutes

5 Discussion

When we look at the meeting minutes generated (Extract 3), we may consider that perhaps they are a little verbose. In fact some might say that just the outcome statement is needed. This observation highlights a point made previously by Spärck Jones (2007) that the purpose of the summarization is extremely important in deciding how and what to summarise. The summary our system produces is the direct result of the policy applied in the system, namely how to recognize transcript features, how to make use of protocols and which sentences to extract and manipulate when the dialogue features are identified. Alternative policies could be implemented. We are in fact exploring alternative policies and are extending the system so that different policies for summarization can be selected depending on the needs of the user. We might also say that the minutes are a little clumsy and have a repetitive style, for instance "PP said", "JJ suggested", "PP answered" etc. An experienced minute writer could certainly summarise better. It is worth remembering however that the generated minutes need not be the final minutes. The V-ROOM system incorporates an option for the user to improve them after generation. Furthermore a post-processor could be developed to improve style by varying sentence construction.

A problem with transcripts either produced through chat or through ASR systems is that there are word errors, i.e. mistyped words or words generated when speech was not recognized properly. The summarization system can still work (Carenini, Murray and Ng 2011) but mistakes in transcripts can be taken through to final output and quality will not be so good. That is why it is useful to have a pre-processor or post-processor to remove as many mistakes as possible. In our example we, can see some mistypes, for instance, line 17, "teh" instead of "the" and "pursueing" instead of "pursuing". Simple errors like this can be smoothed out via a pre-processing or post-processing stage. In the Extract 3, these simple problems have been removed during minutes generation. The development of a more comprehensive post-processor would be a useful addition to the system, although we recognise that not all errors would be captured and corrected.

Our system uses a number of different inputs for determining item boundaries. One of these involves matching the estimated time for an item with the times of the transcript. For instance if the agenda says the first item will take 15 minutes, then the first 15 minutes of dialogue can be taken as representing that item in the meeting. However events and circumstances can invalidate such an assumption. The time assigned to an item in the agenda is just an estimate. The actual time spent on the item might vary significantly. We can see in our example that an estimated time of 10 minutes was given in the agenda for the item *Expenses Update*. In reality the item only took 3 minutes (lines 39-41) because, we see if we read the text, that the student had already received the necessary advice from another professor before the meeting.

In terms of summarisation characteristics, we have seen that our system works on multi-objects in that dialogue contributions are not considered in isolation. Contributions are considered in clusters and in terms of protocols and context, such as the Supervisor-Student protocol of proposal-agreement. We have used a hybrid approach to extraction versus abstraction. We extract relevant sentences but then manipulate them to make them more readable. Our current work is looking at how to improve this further. Our summarisation includes indicative and informative elements in that issues or topics are listed but also an attempt is made to record something more about the conversation around those topics. The summarization system produces textual outputs and is domain-specific in that we have used the context of meetings and within that we try to use protocols of specific meetings were possible.

6 Conclusion and future work

We have shown how the use of an *agenda* with *items* and *roles* can greatly aid the automatic generation of meeting minutes. We have produced a meeting model which aids the identification of a meeting structure and dialogue parts within a meeting transcript. Identification of topics and selection of relevant sentences for extraction can be greatly enhanced by use of the meeting structure. Our system allows for assisted meeting minutes generation or automatic. The automatic mode allows for human intervention later to improve the minutes if needed. Our research has shown that artificial intelligence techniques can be used to support human intellectual work, in this case producing minutes. However it is important to use artificial intelligence for supporting people rather than with an aim of replacing people because artificial intelligence, whilst helpful, cannot be relied on in the same way as a human brain when faced with infinite potential performance scenarios. Furthermore, a wide range of social research is needed regarding the use of artificial intelligence and the perceptions that people have when a software entity interacts on behalf of them. Our future work will involve capturing protocols for different types of meeting, further examination of meeting structure, experimenting with different policies for summarising meetings and improving quality of generated prose.

References

1. Adkins, M., Burgoon, M. and Nunamaker, J.F. Jr. 2003. Using group support systems for strategic planning with the United States Air Force. *Decision Support Systems*, 34: 315-337.
2. Antunes, P. and L. Carrico, L. 2005. Modeling the information structures of groupware. *Sistemas de Informacao*, 17: 21-37.
3. Barthès, P. A. 2011. OMAS - a flexible multi-agent environment for CSCWD, *Future Generation Computer Systems*, 27(1):78-87.
4. Campagnolo, B., Tacla, C. A., Paraiso, E.C. , Sato, G. Y. and Ramos, M. P. An architecture for supporting small collocated teams in cooperative software development, *Proceedings of the 13th Int'l Conf. on Computer Supported*

Cooperative Work in Design (CSCWD 2009) Santiago, Chile. pp. 264-269. Washington, DC, USA: IEEE Computer Society.

5. Carenini, G.,Murray G. and Ng, R. 2011. *Methods for Mining and Summarising Text Coverations*. Morgan and Claypool

6. Carvalho, V. and Cohen, W.,On the collective classification of email "speech acts". *Proceedings of the 31st Annual Conference on Research and Development in Information Retrieval* (ACM SIGIR 2005). Singapore. Pp. 345-352. New York, NY:ACM.

7. Cohen, W. Carvalho, V. and Mitchell, T. Learning to classify email into "speech acts". *Proceedings of the 2004 Conf. Empirical Methods in Natural Language Processing (EMNLP 2004),* Barcelona, Spain. pp 309-316.

8. Colbert, A., Guzman, A, McQueen, M.and Luechtefeld, R. Decision-Making and Conflict Resolutions Skills Enhanced by Virtual Facilitation: A Qualitative Study, *Proceedings of the Frontiers in Education Conference 2011 (FIE 2011)*. pp. F2C-1 - F2C-4.

9. Doodle 2009. http://www.doodle.com/about/mediareleases/survey.html

10. R Fernandez, R., Frampton, M., Ehlen,P., Purver. M. and Peters, S. Modelling and detecting decisions in multi-party dialogue, Proceedings of the 9th Annual Meeting on Discourse and Dialogue (SIGDial 2008). Pp. 156-163. New York, NY: ACM.

11. Griffith, T.L., Fuller, M. A. and Northcraft, G. B. 1998. Facilitator influence in Group Support Systems: Intended and Unintended Effects. Information Systems Research, 9 (1):20-36.

12. Gruenbacher, P..and Briggs, R. O. Surfacing Tacit Knowledge in Requirements Negotiation: Experiences using EasyWinWin.. *Proceedings of the 34th Hawaii Int'l Conf. on System Sciences* (HICSS 34 2001). 8 pp.

13. Guerrero, L. A., and Pino, J. A. (2009). Supporting discussions for decision meetings. *Group Decision and Negotiation*, 18(6): 589-601.

14. Hayne, S. C., 1999. The facilitators perspective on meetings and implications for group support systems design, *ACM SIGMIS Database*, 30(3–4): 72–91.

15. Hsueh P. Y. and Moore, J. What decisions have you made: Automatic decision detection in conversational speech. *Proceedings of the 8th Annual Meeting North American Association for Computational Lingusitic (ACL 2007*). pp 25-32.

16. Huhns, M. N. and Singh, M.P. 1998. Personal assistants, Internet Computing, 2(5): 90-92.

17. Jeong, M., Lin C-Y., and Lee, G., Semi-supervised speech act recognition in emails and forums, *Proceedings of the 2009 conference on Empirical Methods in Natural Language Processing (EMNLP 2009).* pp. 1250-1259.

18*.* Lloret, E., and Palomar,M.   2012. Text summarisation in progress: a literature review, Artificial Intelligence Review, 37(1):21-41.

19. Purver, M., Ehlen, P., and JNiekrasz, J. Detecting action items in multi-part meetings: Annotation and initial experiments. *Proceedings of the 3rd joint workshop on Machine Learning and Multimodal Interaction (MLMI 2006).* pp. 200-211.

20. Macaualy, L.A. and A. Alabdulkarim, A. Facilitation of e-meetings: state-of-the-art review, *Proceedings of the international conference on e-Technology, e-Commerce and e-Service (EEE 2005).* pp 728 – 735.

21. Martins, L.L., Gilson, L. and Maynard, M. 2004. Virtual teams: What do we know and where do we go from here? *Journal of Management*, 30 (6): 805–835.

22. Murray, G. and Renals, S. Detecting action items in meetings, *Proceedings of the 5th joint workshop on Machine Learning and Multimodal Interaction (MLMI 2008).* pp. 208-213.

23. Nanos, A. G. and James, A. E. A Virtual Meeting System for the New Age, *Proceedings of the 10th IEEE International Conference on E-Business Engineering (ICEBE 2013*), Coventry, UK. pp 98-105. Washington, DC, USA: IEEE Computer Society.

24. Oh, A., Tuchinda, R. and Wu, L. MeetingManager: A collaborative tool in the Intelligent Room, *Proceedings of the Student Oxygen Workshop*, 2001.

25. Pang, P. and Lee, L. 2008. Opinion mining and sentiment analysis, *Foundations & Trends in Information Retrieval*, 2 (1-2):1-135.

26. Paraiso, E. C. and Malucelli, A*.* 2012. Ontologies Supporting Intelligent Agent-Based Assistance*, Computing and Informatics,* 30(4), pp. *829-855.*

27. Rahman, S. A., Yadav, D., Agerwal, P., and Bisth, P.S. 2012. Multiagent Knowledge Management Architecture, *Journal of Software Engineering and Application*, 5:33-40.

28. Rangarajan, N. and J. Rohrbaugh, J. 2003 Multiple roles of online facilitation: An example in any-time, any-place meetings. *Group Facilitation*, 5 (1): 26–36.

29. Roman, F., Mubin, O., and Dillenbourg, P. ReflectWorld: a distributed architecture for meetings and groups evolution Analysis, Proceedings of the 10th international conference on Collaboration Technologies and Systems (CTS 2012), pp. 389 –396.

30. Romano, N. C., Nunamaker, J. F. Jr. , Briggs, R. O. and Mittleman, D. D. Distributed GSS facilitation and participation: field action research, Proceedings of the 32nd Annual Hawaii international. Conference on. Systems Science (HICSS 32), 1999, pp. 5 – 8.

31. Saggion, H. and Poibeau, T. Automatic text summarization: Past, present and future, *Multi-source, Multilingual Information Extraction and Summarization,* Springer Berlin Heidelberg, 2013, pp.3-12.

32. Ritter, A., C Cherry, C., and B Dolan, B., Unsupervised modelling of twitter conversations, *Proceedings of the 11th Annual Meeting North America Association for Computational Linguistics (ACL 2010),* pp. 172-180.

33. Shrestha, L. and K McKeown, K. Detection of question-answer pairs in email conversations. *Proceedings of the 20th international conference on Computational Linguistics (COLING 2004),* pp. 889-895. New York, NY: ACM.

34. Spärck Jones, K. 2007. Automatic summarising: the state of the art, *Information Processing & Management*, 43(6):1449-1481.

35. Straub, J. and Huber, J. 2013. A characterization of the utility of using   artificial   intelligence   to   test   two artificial intelligence systems,*Computers*, 2 (2): 67-87.

36. Sugawara, K., Manabe, Y., and Fujita, S. Mobile symbiotic interaction between a user and a personal assistant agent, *Proceedings of the 11th international conference on Cognitive Informatics & Cognitive Computing (ICCI* CC 2012)*, pp. 341-345. Washington, DC, USA: IEEE Computer Society.

37. Thompson, P., James, A., and E. Stanciu, E., Agent based Ontology Driven Virtual Meeting Assistant, *Future Generation Information Technology*, LNCS, 6485, 2010, pp 51-62.

38. Thompson, P, James, A. and Nanos, A. G. V-ROOM: Virtual Meeting System Trial, *Proceedings of the 17th international conference on Computer Supported Cooperative Work in Design (CSCWD 2013),* Whistler, Canada, pp. 563-569. Washington, DC, USA: IEEE Computer Society.

39. Thompson, P., James, A., and Iqbal, R. Agent-based facilitator assistant for virtual meetings, *Proceedings of 15th international conference on Computer Supported Cooperative Work in Design* (CSCWD 2011), Laussane, Switzerland, pp. 335 – 341. Washington, DC, USA: IEEE Computer Society.

40. Wang, W., Haake, J. M. and Rubart, J. 2004. Supporting virtual meetings in the overall business context, *International Journal of Computer Applications in Technology,* 19: 195 –208.

41. Wang, W. Meeting in Browsers: The PowerMeeting Real-Time Collaboration Service, Proceedings Case Studies in Service Innovation, Centre for Service Research, Manchester, 2010, pp. 87-89.

42. Wiggins, T., Swift, D., Mai, U. and Luechtefeld, R. Acquiring professional skills: virtual facilitator as model for team communication, *Proceedings of Frontiers in Education (FIE 2011),* Rapid City, South Dakota, USA, pp. S1C-1 - S1C-5.

43. Yankelovich, N., Walker, W., Roberts, P., Wessler, M., Kaplan, J. and Provino, J. Meeting central: making distributed meetings more effective, *Proceedings of the conference on Computer Supported Cooperative Work (CSCW 2004)*, pp. 419 – 428.

44. Zhao, J. L., Nunamaker, J. F. Jr. Briggs, R. O. Intelligent workflow techniques for distributed group facilitation, *Proceedings of the 35th Hawaii international conference. on System Sciences (HICSS 35)*, 2002, pp. 597 – 605.

Appendix A Transcript of a V-ROOM supervisor-student meeting

| Line No	Speaker-Time		Item	Structure	Topic
1	JJ- 3:01 PM Jul, 04 2012	Hello Philip, are you in teh meeting room	Other		
2	PP- 3:02 PM Jul, 04 2012	I can see the discussion space	Other		
3	PP- 3:04 PM Jul, 04 2012	But I've navigated the site and can't find the 'Meeting Space'	Other		
4	JJ- 3:11 PM Jul, 04 2012	Ok - this is actually the meeting space - it is just chat	Other		
5	JJ- 3:11 PM Jul, 04 2012	Anyway how are things going with the project?	Progress Update	Project Update start; Question	
6	PP- 3:12 PM Jul, 04 2012	Hi	Progress Update		
7	JJ- 3:13 PM Jul, 04 2012	Hello	Progress Update		
8	PP- 3:13 PM Jul, 04 2012	Sorry about that, it looks like you have to refresh it every 10 minutes	Progress Update		
9	JJ- 3:13 PM Jul, 04 2012	Can you sum up how things are going with the project?	Progress Update	Content Summary; Recognised Duplicate (5)	
10	PP- 3:13 PM Jul, 04 2012	So I've started the programming aspect in eclipse	Project Update	Answer (5)	
11	JJ- 3:14 PM Jul, 04 2012	The GUI for the app has already been built	Progress Update	Content Summary	GUI
12	PP- 3:14 PM Jul, 04 2012	But I'm having trouble compiling the code since it seems that the android library is not being referenced properly.	Progress Update	Content Summary; continuation (11);	GUI; Compiling; Android
13	JJ- 3:14 PM Jul, 04 2012	Does the GUI control the Robot simulation yet?	Progress Update	Content Summary;	GUI
14	PP- 3:14 PM Jul, 04 2012	Even the rosjava tutorials are not compiling properly	Progress Update	Content Summary Continuation(12)	ROSJava; compiling
15	JJ- 3:15 PM Jul, 04 2012	No, the GUI at the moment is simply a screen with the seek bars for adjustment and text labels	Progress Update	Content Summary; Answer (13)	GUI
16	PP- 3:15 PM Jul, 04 2012	I need to get the code to compile properly or else it won't install and run	Progress Update	Content Summary	compile
17	JJ- 3:15 PM Jul, 04 2012	It does seem quite a problem if teh ROSjava tutorials are not compiling. Do you think it is worth pursueing?	Progress Update	Content Summary	ROSJava; compiling
18	PP- 3:15 PM Jul, 04 2012	I have been looking online	Progress Update	Content Summary	
19	PP- 3:16 PM Jul, 04 2012	and have not found that there is more discrepancies in the packages due to the rosjava updates	Progress Update	Content Summary; continuation(18)	ROSJava; packages
20	PP- 3:16 PM Jul, 04 2012	I mean I have found that there is discrepancies in the packages due to the rosjava updates	Progress Update	Content Summary	ROSJava; packages
21	PP- 3:17 PM Jul, 04 2012	as a result when I try to look online for help or documentation	Progress Update	Content Summary; continuation(20)	documentation
22	PP- 3:17 PM Jul, 04 2012	they are either outdated or using irrelevant packages	Progress Update	Content Summary; continuation(21)	packages
23	PP- 3:17 PM Jul, 04 2012	It seems that unless they prepare proper documentation	Progress Update	Content Summary; continuation(22)	documentation
24	PP- 3:18 PM Jul, 04 2012	it will be difficult to create a fully working system without bugs	Progress Update	Content Summary; continuation(23)	
25	PP- 3:18 PM Jul, 04 2012	due to the fact that everything related to ros is unorganized and fragmeneted	Progress Update	Content Summary;	

		due to version updates which are not backwards compatible		continuation(24)	
26	JJ- 3:19 PM Jul, 04 2012	Could you perhaps then think of something else you would rather do in your time here?	Progress Update	Content Summary; Question	
27	PP- 3:19 PM Jul, 04 2012	So in my opinion at the moment it is not worth pursuing as a viable option but in the future it will definitely be an option	Progress Update	Content Summary; Answer(17)	
28	PP- 3:20 PM Jul, 04 2012	it probably could be created now but you would need an ROS developer's help	Progress Update	Content Summary	ROS
29	PP- 3:20 PM Jul, 04 2012	in the mean time, I could look at android and the cloud or something mobile computing related	Progress Update	Proposal; Answer(26)	Cloud Android
30	PP- 3:21 PM Jul, 04 2012	since I have already spent considerable time on the android aspect and the cloud computing aspect when I first came	Progress Update	continuation(29)	Cloud Android
31	PP- 3:22 PM Jul, 04 2012	and robot simulation really requires some high level mechatronics classes which I would later take if I got a mechanical engineering minor but I have not yet	Progress Update	continuation(30)	
32	JJ- 3:22 PM Jul, 04 2012	Ok - perhaps then you should write a brief report on your findings regarding ROS and ROS Java and teh simulation. Then think of an Android type application that you could develop	Progress Update	Agreement; Proposal	ROS; ROSJava; Android
33	PP- 3:22 PM Jul, 04 2012	I can definitely write up some documentation, and will try to come up with a new proposal by this weekend	Progress Update	Proposal; Action	
34	JJ- 3:24 PM Jul, 04 2012	If you can link it somehow to Clem it would be good. Maybe make some sort of quiz infrastructure where teachers can input multichoice questions and tehre quizzes can be accessed ia Android.	Progress Update	Proposal	CLEM
35	JJ- 3:24 PM Jul, 04 2012	Or any application could be okay but better if related somehow to CLEM.	Progress Update	Proposal; continuation(34)	CLEM Android
36	PP- 3:24 PM Jul, 04 2012	Ok, I'll look into it	Progress Update	Agreement; Outcome	
37	JJ- 3:25 PM Jul, 04 2012	Ok then. Let's move onto teh Expense item. Are you okay with this? Do you understand what to do?	Expenses Update	Question x2	Move
38	PP- 3:26 PM Jul, 04 2012	Professor Roberts actually stopped by today and clarified the process	Expenses Update	Answer(37)	
39	PP- 3:26 PM Jul, 04 2012	I'm going to give him the completed form and receipts this friday.	Expenses Update	Continuation(38) Action	
40	PP- 3:27 PM Jul, 04 2012	sorry about that	Expenses Update		
41	PP- 3:28 PM Jul, 04 2012	it logged me out for some reason	Expenses Update		
42	PP- 3:28 PM Jul, 04 2012	Is there anything else?	AOB	Question	
43	JJ- 3:28 PM Jul, 04 2012	Yea - me too	AOB	Answer(42)	
44	JJ- 3:29 PM Jul, 04 2012	About the move. Have you discussed this with Prof Roberts	AOB	Question	Move
45	PP- 3:29 PM Jul, 04 2012	Not yet, since I wasnt sure if I should continue debugging the java files when we met	AOB	Answer(44)	
46	JJ- 3:30 PM Jul, 04 2012	I meant teh physical move out of the building	AOB		Move
47	PP- 3:30 PM Jul, 04 2012	Oh.	AOB		
48	PP- 3:30 PM Jul, 04 2012	No, not yet.	AOB		
49	PP- 3:30 PM Jul, 04 2012	Is there something specific I need to do?	AOB	Question	
50	PP- 3:32 PM Jul, 04 2012	Not really except that you will need to work from home apart from when you have prearranged meetings. Anyway I was wondering if we could meeet on Monday - are you in or will you be taking time off	AOB	Answer(49) Question	

51	PP- 3:32 PM Jul, 04 2012	I'll definitely help pack and move everything out.	AOB	Action	Move
52	JJ- 3:32 PM Jul, 04 2012	shall we meet Monday at 15.00?	AOB	Question	Meet
53	PP- 3:33 PM Jul, 04 2012	Where?	AOB	Question	
54	JJ- 3:33 PM Jul, 04 2012	I think your current room wll still be available that day. So we can meet there	AOB	Proposal	Meet
55	PP- 3:34 PM Jul, 04 2012	Ok, sounds good.	AOB	Agreement	
56	JJ- 3:34 PM Jul, 04 2012	ok but if you want time off for your parents' visit just let me know	AOB	Proposal	
57	PP- 3:35 PM Jul, 04 2012	Well Im visiting them on the weekend	AOB		
58	JJ- 3:35 PM Jul, 04 2012	We can finish the meeting now. Bye.	AOB	Proposal	
59	PP- 3:35 PM Jul, 04 2012	Ok. Bye.	AOB	Agreement	

Figures
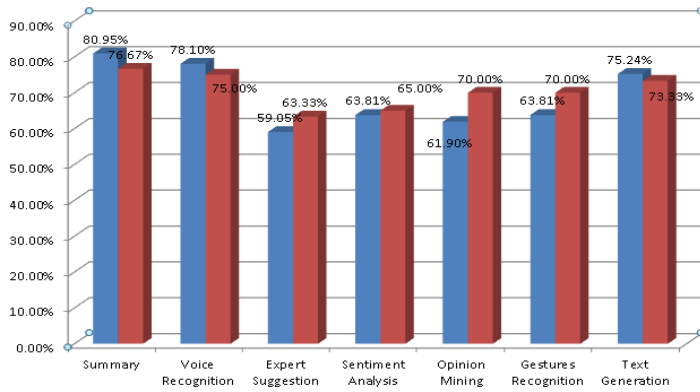
The left side represents the juniors and the right the seniors.

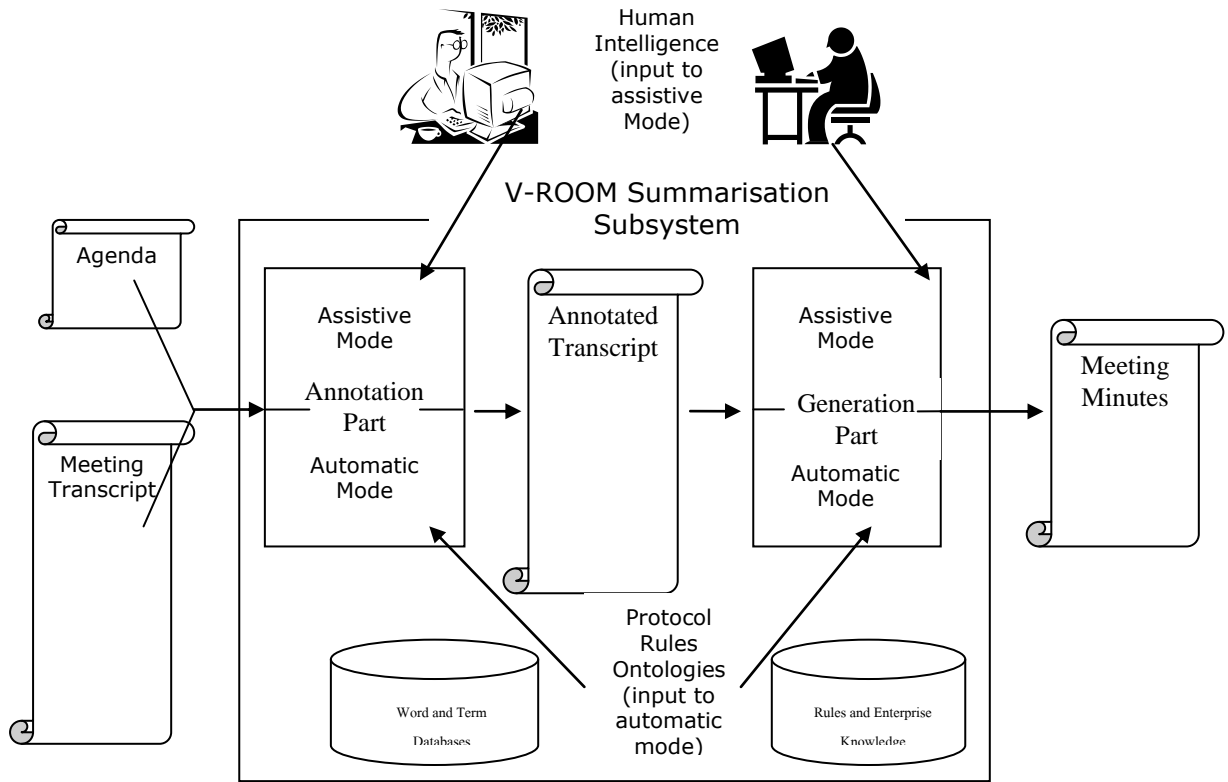Figure 1. The choices by users for future features of virtual meeting systems

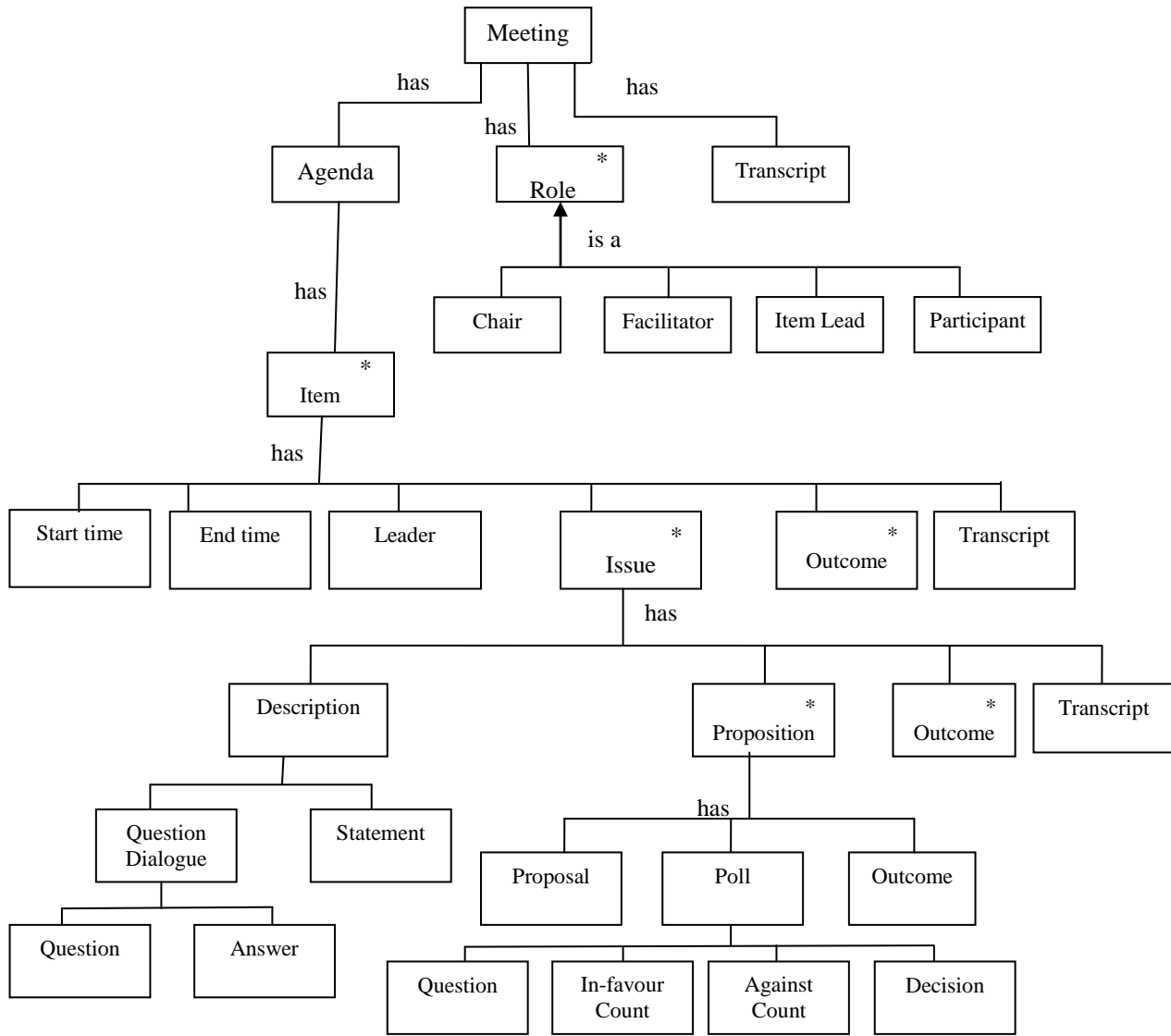Figure 2. High-level structure of V-ROOM summarisation sub-system

Meeting

has

has

has

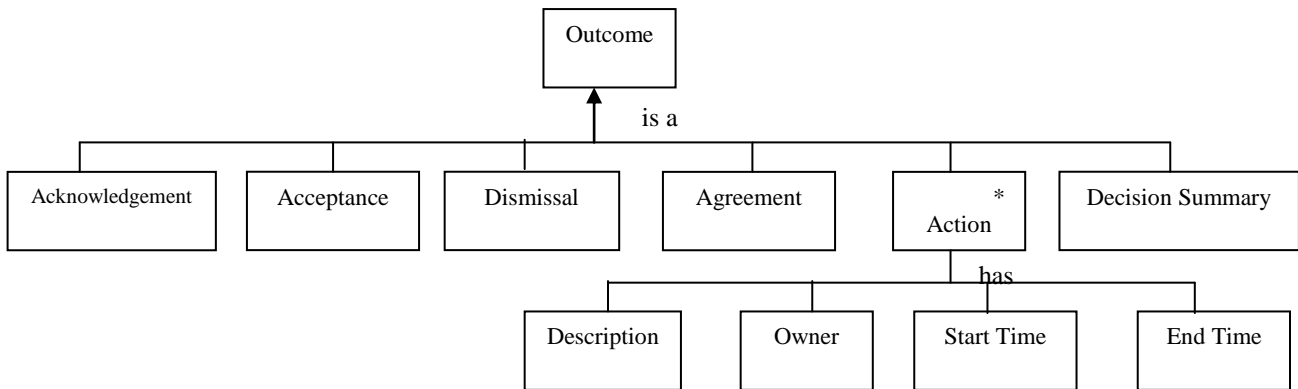Agenda

Role *

Transcript

is a

Chair

Facilitator

Item Lead

Participant

has

Item *

has

Start time

End time

Leader

Issue *

Outcome *

Transcript

has

Description

Proposition *

Outcome *

Transcript

Question Dialogue

Statement

has

Proposal

Poll

Outcome

Question

Answer

Question

In-favour Count

Against Count

Decision

Figure 3. V-ROOM meeting model

Outcome

is a

Acknowledgement

Acceptance

Dismissal

Agreement

Action *

Decision Summary

has

Description

Owner

Start Time

End Time

Figure 4. Structure of an outcome
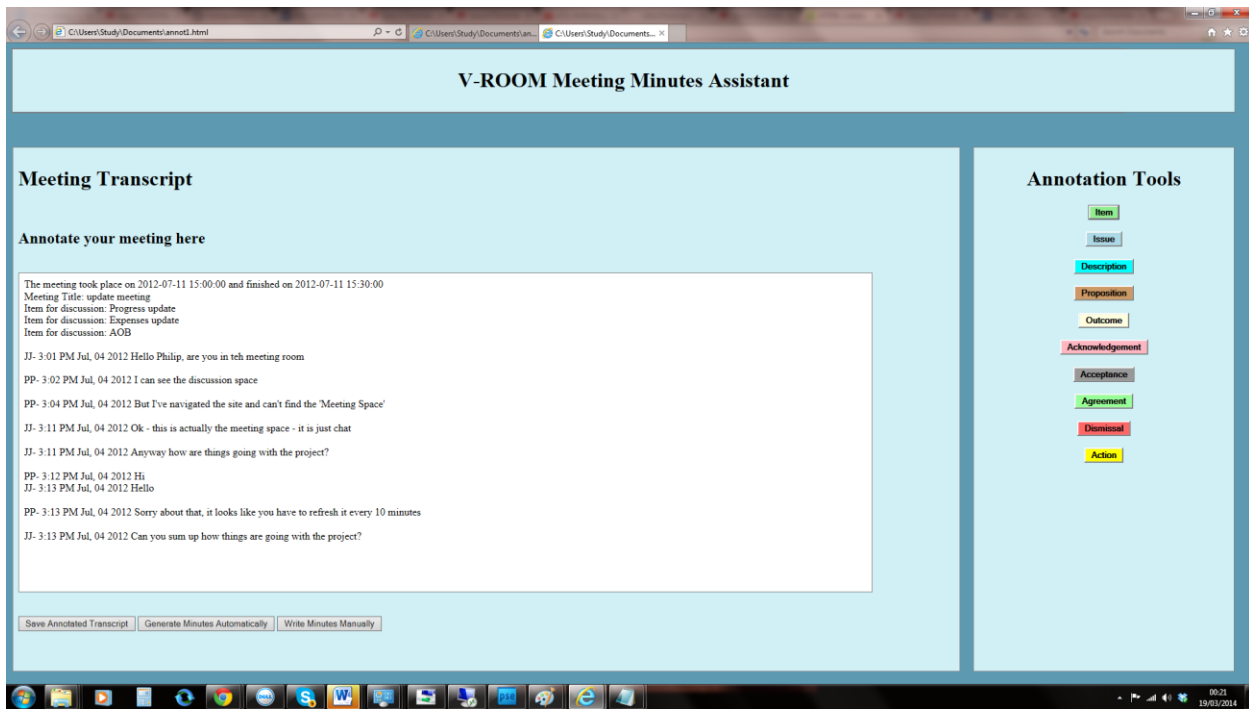
Figure 5. Assisted mode interface

(a)   Select date

(b)   Add items or invite

(c)   Adding an item

Figure 6. Setting up an agenda in V-ROOM

Meeting Title: update meeting
(Date: 4-07-2012, Start: 15:00, Finish:15:30)

Item for discussion: Progress update (15 minutes)
Item for discussion: Expenses update (10 minutes)
Item for discussion: AOB (5 minutes)

Attending: JJ (student); PP (supervisor)

JJ- 3:01 PM Jul, 04 2012

Hello Philip, are you in teh meeting room

…..

Progress update

PP- 3:15 PM Jul, 04 2012

I have been looking online

PP- 3:16 PM Jul, 04 2012

and have not found that there is more .

…...

PP- 3:26 PM Jul, 04 2012

Expenses update

I'm going to give him the completed form and receipts   this Friday

PP- 3:27 PM Jul, 04 2012

sorry about that

AOB

PP- 3:30 PM Jul, 04 2012

Is there something specific I need to do?

Figure 7. Using timings to suggest item boundaries

```
                    ┌─────────────────────────┐
                    │  Count frequency of words│
                    │    excluding stop words  │
                    └─────────────────────────┘
                                │
                                ▼
                    ┌─────────────────────────┐
                    │     Cluster into Topics  │
                    └─────────────────────────┘
                                │
                                ▼
                    ┌─────────────────────────┐
                    │  Identify Description,   │
                    │  Proposal and Outcome    │
                    │        Dialogue          │
                    └─────────────────────────┘
                                │
                                ▼
                    ┌─────────────────────────┐
                    │      Select Issues       │
                    └─────────────────────────┘
                                │
                                ▼
                    ┌─────────────────────────┐
                    │    Extract or Compose    │
                    │        Sentences         │
                    └─────────────────────────┘
                                │
                                ▼
                    ┌─────────────────────────┐
                    │  Produce summary in the  │
                    │  form of Meeting Minutes │
                    └─────────────────────────┘
```
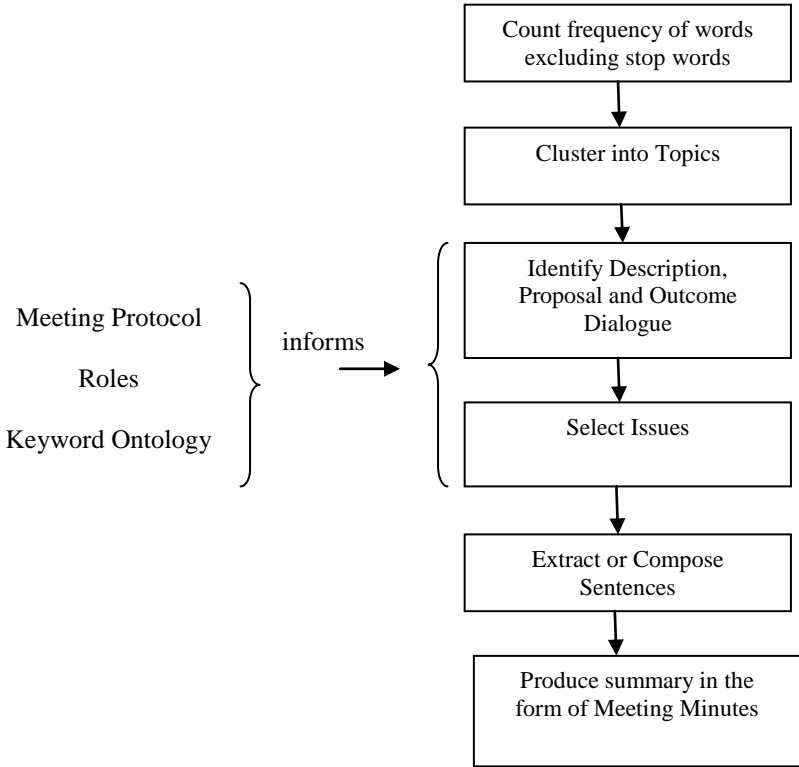
Meeting Protocol

Roles

Keyword Ontology

} informs →

Figure 8. Stages in generating minutes from item transcripts