Real-time decision detection in multi-party dialogue

Matthew Frampton, Jia Huang, Trung Huu Bui and **Stanley Peters**
Center for the Study of Language and Information
Stanford University
Stanford, CA, 94305, USA
{frampton|jiahuang|thbui|peters}@stanford.edu

Abstract

We describe a process for automatically detecting decision-making sub-dialogues in multi-party, human-human meetings in real-time. Our basic approach to decision detection involves distinguishing between different utterance types based on the roles that they play in the formulation of a decision. In this paper, we describe how this approach can be implemented in real-time, and show that the resulting system's performance compares well with other detectors, including an off-line version.

1 Introduction

In collaborative and organized work environments, people share information and make decisions through multi-party conversations, commonly referred to as meetings. The demand for automatic methods that process, understand and summarize information contained in audio and video recordings of meetings is growing rapidly, as evidenced by on-going projects which are focused on this goal, (Waibel et al., 2003; Janin et al., 2004). Our research is part of a general effort to develop a system that can automatically extract and summarize information such as conversational topics, action items, and decisions.

This paper concerns the development of a real-time decision detector — a system which can detect and summarize decisions as they are made during a meeting. Such a system could provide a summary of all of the decisions which have been made up until the current point in the meeting, and this is something which we expect will help users to enjoy more productive meetings. Certainly, good decision-making relies on access to relevant information, and decisions made earlier in a meeting often have a bearing on the current topic of discussion, and so form part of this relevant information. However, in a long and winding meeting, participants might not have these earlier decisions at the forefront of their minds, and so an accurate and succinct reminder, as provided by a real-time decision detector, could potentially be very useful. A record of earlier decisions could also help users to identify outstanding issues for discussion, and to therefore make better use of the remainder of the meeting.

Our approach to decision detection uses an annotation scheme which distinguishes between different types of utterance based on the roles which they play in the decision-making process. Such a scheme facilitates the detection of decision discussions (Fernández et al., 2008), and by indicating which utterances contain particular types of information, it also aids their summarization. To automatically detect decision discussions, we use what we refer to as *hierarchical classification*. Here, independent binary *sub-classifiers* detect the different decision dialogue acts, and then based on the sub-classifier hypotheses, a *super-classifier* determines which dialogue regions are decision discussions. In this paper then, we address the challenges for applying this approach in real-time, and produce a system which is able to detect decisions soon after they are made, (for example within a minute). We conduct tests and compare this system's performance with other detectors, including an off-line equivalent.

The remainder of the paper proceeds as follows. Section 2 describes related work, and Section 3 describes our annotation scheme for decision discussions, and our experimental data. Next, Section 4 explains the hierarchical classification approach in more detail, and Section 5 considers how it can be applied in real-time. Section 6 describes the experiments in which we test the real-time detector, and finally, Section 7 presents conclusions and ideas for future work.

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1133–1141,
Singapore, 6-7 August 2009.

2 Related Work

Decisions are one of the most important meeting outputs. User studies (Lisowska et al., 2004; Banerjee et al., 2005) have confirmed that meeting participants consider this to be the case, and Whittaker et al. (2006) found that the development of an automatic decision detection component is critical to the re-use of meeting archives. As a result, with the new availability of substantial meeting corpora such as the ISL (Burger et al., 2002), ICSI (Janin et al., 2004) and AMI (McCowan et al., 2005) Meeting Corpora, recent years have seen an increasing amount of research on decision-making dialogue.

This recent research has tackled issues such as the automatic detection of agreement and disagreement (Hillard et al., 2003; Galley et al., 2004), and of the level of involvement of conversational participants (Wrede and Shriberg, 2003; Gatica-Perez et al., 2005). In addition, Verbree et al. (2006) created an argumentation scheme intended to support automatic production of argument structure diagrams from decision-oriented meeting transcripts. Only very recent research has specifically investigated the automatic detection of decisions, namely (Hsueh and Moore, 2007) and (Fernández et al., 2008).

Hsueh and Moore (2007) used the AMI Meeting Corpus, and attempted to automatically identify dialogue acts (DAs) in meeting transcripts which are "decision-related". Within any meeting, the authors decided which DAs were decision-related based on two different kinds of manually created summary: the first was an extractive summary of the whole meeting, and the second, an abstractive summary of the decisions which were made. Those DAs in the extractive summary which support any of the decisions in the abstractive summary were manually tagged as decision-related. Hsueh and Moore (2007) then trained a Maximum Entropy classifier to recognize this single DA class, using a variety of lexical, prosodic, dialogue act and conversational topic features. They achieved an F-score of 0.35, which gives an indication of the difficulty of this task.

Unlike Hsueh and Moore (2007), Fernández et al. (2008) made an attempt at modelling the structure of decision-making dialogue. They designed an annotation scheme that takes account of the different roles which different utterances play in the decision-making process — for example, their scheme distinguishes between decision DAs (DDAs) which initiate a discussion by raising a topic/issue, those which propose a resolution, and those which express agreement for a proposed resolution and cause it to be accepted as a decision. The authors applied the annotation scheme to a portion of the AMI corpus, and then took what they refer to as a *hierarchical classification* approach in order to automatically identify decision discussions and their component DAs. Here, one binary *Support Vector Machine (SVM)* per DDA class hypothesized occurrences of that DDA class, and then based on the hypotheses of these so-called *sub-classifiers*, a *super-classifier*, (a further SVM), determined which regions of dialogue represented decision discussions. This approach produced better results than the kind of "flat classification" approach pursued by Hsueh and Moore (2007) where a single classifier looks for examples of a single decision-related DA class. Using manual transcripts, and a variety of lexical, utterance, speaker, DA and prosodic features for the sub-classifiers, the super-classifier's F1-score was 0.58 according to a lenient match metric. Note that (Purver et al., 2007) had previously pursued the same basic approach as Fernández et al. (2008) in order to detect action items.

While both Hsueh and Moore (2007), and Fernández et al. (2008) attempted off-line decision detection, in this paper, we attempt real-time decision detection. We take the same basic approach as Fernández et al. (2008), and make changes to its implementation so that it can work effectively in real-time.

3 Data

The AMI corpus (McCowan et al., 2005), is a freely available corpus of multi-party meetings containing both audio and video recordings, as well as a wide range of annotated information including dialogue acts and topic segmentation. Conversations are all in English, but participants can include non-native English speakers. All of the meetings in our sub-corpus last around 30 minutes, and are scenario-driven, wherein four participants play different roles in a company's design team: *project manager*, *marketing expert*, *interface designer* and *industrial designer*. The discussions concern how to design a remote control.

We used the off-line version of the Decipher speech recognition engine (Stolcke et al., 2008) in

order to obtain off-line ASR transcripts for these 17 meetings, and the real-time version, to obtain real-time ASR transcripts. Decipher generates the transcripts by first producing Word Confusion Networks (WCNs) and then extracting their best paths. The real-time recognizer generates "live" transcripts with 5 to 15 seconds of latency for immediate display. In processing completed meetings, the off-line system makes seven recognition passes, including acoustic adaptation and language model rescoring, in about 4.2 times real-time (on a 4-score 2.6 GHz Opteron server). In general usage with multi-party dialogue, the word error rate (WER) for the off-line version of Decipher is approximately 23%, and for the real-time version, approximately 35%[1]. Stolcke et al. (2008) report a WER of 26.9% for the off-line version on AMI meetings.

The real-time ASR transcripts for the 17 meetings contain a total of 8440 utterances/dialogue acts, (around 496 per meeting), and the off-line ASR transcripts, 7495 utterances/dialogue acts, (around 441 per meeting).

3.1 Modelling Decision Discussions

We use the same annotation scheme as (Fernández et al., 2008) in order to model decision-making dialogue. As stated in Section 2, this scheme distinguishes between a small number of dialogue act types based on the role which they perform in the formulation of a decision. Recall that using this scheme in conjunction with *hierarchical classification* produced better decision detection than a "flat classification" approach with a single "decision-related" DA class. Since it indicates which utterances contain particular types of information, such a scheme also aids the summarization of decision discussions.

The annotation scheme (see Table 1 for a summary) is based on the observation that a decision discussion contains the following main structural components: (a) a topic or issue requiring resolution is raised, (b) one or more possible resolutions are considered, (c) a particular resolution is agreed upon, that is, it becomes the decision. Hence the scheme distinguishes between three corresponding decision dialogue act (DDA) classes: *Issue* (*I*), *Resolution* (*R*), and *Agreement* (*A*). Class *R* is further subdivided into *Resolution Proposal* (*RP*) and *Resolution Restatement* (*RR*). Note that an utterance can be assigned to more than one of these DDA classes, and that within a decision discussion, more than one utterance may correspond to a particular DDA class.

Here we use the sample decision discussion below in 1 in order to provide examples of the different DDA types. *I* utterances introduce the topic of the decision discussion, examples being *"Are we going to have a backup?"* and *"But would a backup really be necessary?"* On the other hand, *R* utterances specify the resolution which is ultimately adopted as the decision. *RP* utterances propose this resolution (e.g. *"I think maybe we could just go for the kinetic energy..."*), while *RR* utterances close the discussion by confirming/summarizing the decision (e.g. *"Okay, fully kinetic energy"*). Finally, *A* utterances agree with the proposed resolution, so causing it to be adopted as the decision, (e.g. *"Yeah"*, *"Good"* and *"Okay"*.

(1) A: Are we going to have a backup?
Or we do just–
B: But would a backup really be necessary?
A: I think maybe we could just go for the kinetic energy and be bold and innovative.
C: Yeah.
B: I think– yeah.
A: It could even be one of our selling points.
C: Yeah –*laugh*–.
D: Environmentally conscious or something.
A: Yeah.
B: Okay, fully kinetic energy.
D: Good.[2]

3.2 Experimental data for real-time decision detection

Originally, two individuals used the annotation scheme described above in order to annotate the manual transcripts of 9 and 10 meetings respectively. The annotators overlapped on two meetings, and their *kappa* inter-annotator agreement ranged from 0.63 to 0.73 for the four DDA classes. The highest agreement was obtained for class *RP*, and the lowest for class *A*. Although these kappa values are not extremely high, if we used a single, less homogeneous "decision-related" DA class like Hsueh and Moore (2007), then its kappa score

[1] This information was obtained through personal communication.

[2] This example was extracted from the AMI dialogue ES2015c and has been modified slightly for presentation purposes.

key	DDA class	description
I	*issue*	utterances introducing the issue or topic under discussion
R	*resolution*	utterances containing the resolution adopted as the decision
RP	*– proposal*	– utterances where the decision is originally proposed
RR	*– restatement*	– utterances where the decision is confirmed or restated
A	*agreement*	utterances explicitly signalling agreement with the decision

Table 1: Set of decision dialogue act (DDA) classes

would probably be significantly lower. The decision discussion annotations used by Hsueh and Moore (2007) are part of the AMI corpus, and are for the manual transcriptions. The reader can find a comparison between these annotations and our own manual transcript annotations in (Fernández et al., 2008).

After obtaining the new off-line and real-time ASR transcripts, we transferred the DDA annotations from the manual transcripts. In both sets of ASR transcripts, each meeting contains on average around 26 DAs tagged with one or more of the DDA sub-classes in Table 1. DDAs are thus very sparse, corresponding to only 5.3% of utterances in the real-time transcripts, and 6.0% in the off-line. In the real-time transcripts, *Issue* utterances make up less than 1.2% of the total number of utterances in a meeting, while *Resolution* utterances are around 1.6%: 1.2% are *RP* and less than 0.4% are *RR* on average. Almost half of DDA utterances (slightly over 2.6% of all utterances on average) are tagged as belonging to class *Agreement*. In the off-line transcripts, the percentages are fairly similar: 1.6% of utterances are *Issue* DDAs, 2.0% are *RP*, 0.5% are *RR*, and 2.4% are *A*.

We now move on to describe the *hierarchical classification* approach which we use to try to automatically detect decision sub-dialogues and their component DDAs.

4 Hierarchical Classification

Hierarchical classification is designed to exploit the fact that within decision discussions, our DDAs can be expected to co-occur in particular types of patterns. It involves two different types of classifier:

1. **Sub-classifier:** One independent binary *sub-classifier* per DDA class classifies each utterance.

2. **Super-classifier:** A sliding window shifts through the meeting one utterance at a time, and following each shift, a binary *super-classifier* determines whether the region of dialogue within the window is part of a decision discussion.

In our decision detectors, the sub-classifiers run in parallel in order to reduce processing time. For each utterance, the sub-classifiers use features which are derived from the properties of that utterance in context. On the other hand, the super-classifier's features are the hypothesized class labels and confidence scores for the utterances within the window. In various experiments, we have found that a suitable size for the window, is the average length of a decision discussion in our data in utterances. The super-classifier also "corrects" the sub-classifiers. This means that if a DA is classified as positive by a sub-classifier, but does not fall within a region classified as part of a decision discussion by the super-classifier, then the sub-classifier's hypothesis is changed to negative.

We now move on to consider how this basic approach to decision detection can be implemented in a real-time system.

5 Design considerations for our real-time system

A real-time decision detector should detect decisions as soon after they are made as possible. It is for this reason that we have set our real-time detector to automatically run at frequent and regular intervals during a meeting. An alternative would be to give the user (a meeting participant) responsibility for instructing the detector when to run. However, a user may sometimes leave substantial gaps between giving run commands. When this happens, the detector will have to process a large number of utterances in a single run, and so the user may wait some time before being presented with any results. In addition, giving the user responsibility for instructing the detector when to

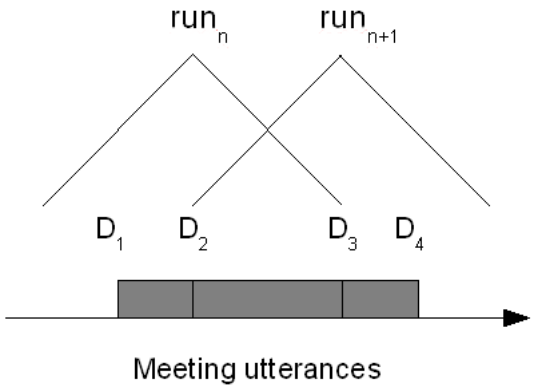

Figure 1: Decision discussion regions hypothesized by consecutive runs overlap (D_1 to D_3 and D_2 to D_4) and so are merged.

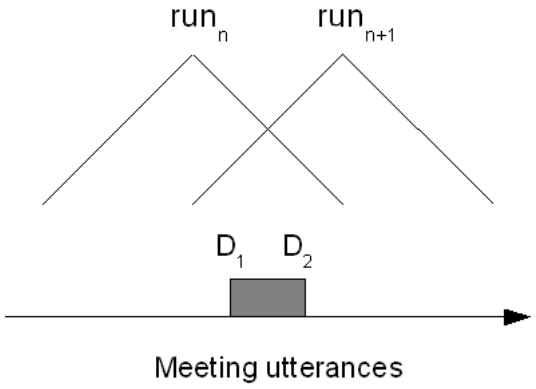

Figure 2: Consecutive runs hypothesize the same decision discussion region D_1 to D_2, and so one of the duplicates is discarded.

run means burdening the user with an extra task to perform during the meeting, and this goes against the general philosophy behind the system's development. The system is intended to be as unobtrusive as possible during the meeting, and to relieve users of tasks which distract their attention away from the current discussion (e.g. note-taking), not to create new tasks, however small.

Obviously, on the first occasion that the detector runs during a meeting, it can only process "new" (previously unprocessed) utterances, but on subsequent runs, it has the option to reprocess some number of "old" utterances (utterances which it has already processed in a previous run). Certainly, the detector should reprocess some of the most recent old utterances because it is possible that a decision discussion straddles these utterances and new utterances. However, the number of old utterances that are reprocessed should be limited. If the meeting has lasted a while already, then the processing of a large portion of the earlier old utterances is likely to be redundant — it will simply produce the same results for these utterances as the previous run.

The fact that the real-time detector processes recent old utterances means that consecutive runs can produce hypotheses for decision discussion regions which overlap, or which are duplicates. Figure 1 gives an example of the former. We deal with overlapping hypotheses by merging them into one, so forming a larger single decision discussion region. Figure 2 gives an example of duplicate hypotheses. Here, on run n, the detector hypothesizes decision discussion D_1 to D_2, and then on run $n+1$, since the bounds of this original hypothesis are now wholly contained within the region of old reprocessed utterances, the detector hypothesizes a duplicate. We deal with such cases by discarding the duplicate.

6 Experiments

We conducted various experiments related to real-time decision detection, our goal being to produce a system which:

- relative to alternative versions, detects decision discussions accurately,
- generates results for any portion of dialogue very soon after that portion of dialogue has ended.

The current version of our real-time detector is set to process the same number of old and new utterances on each run. Here, we refer to this value as i, and hence on each run the system processes a total of $2i$ utterances (i old and i new). Another of the system's characteristics is that runs take place every i utterances, meaning that as we decrease i, the system provides new results more frequently and is hence "more real-time". One of the things we investigate here then, is what to set i to in order to best satisfy the two design goals given above. Having found this value, we compare the hierarchical real-time detector's performance with alternative detectors, these being:

- an off-line detector applied to off-line ASR transcripts,
- a flat real-time detector,
- an off-line detector applied to the real-time ASR transcripts.

Lexical	unigrams after text normalization
Utterance	length in words, duration in word rate
Speaker	speaker ID & AMI speaker role
Context	features as above for utterances u +/- 1...u +/- 5

Table 2: Features for decision DA detection

Note that the off-line detectors use hierarchical classification, and that the flat real-time detector uses a single binary classifier which treats all DDAs as members of a single merged DDA class.

6.1 Classifiers and features

All classifiers (sub and super-classifiers) in all detectors are linear-kernel *Support Vector Machines (SVMs)*, produced using *SVMlight* (Joachims, 1999). For the sub-classifiers, we are obviously restricted to using features which can be computed in a very short period of time, and in the experiments here, we use *lexical*, *utterance* and *speaker* features. These are summarized in Table 2. An utterance's lexical features are the words in its transcription, its utterance features are its duration, number of words, and word rate (number of words divided by duration), and its speaker features are the speaker's role (see Section 3) and ID. We also use lexical features for the previous and where available, next utterances: the *I*, *RP* and *RR* sub-classifiers use the lexical features for the previous/next utterance and the *A* sub-classifier, those from the previous/next 5 utterances. These settings produced the best results in preliminary experiments. We do not use DA features because we lack an automatic DA tagger, nor do we use prosodic features because (Fernández et al., 2008) was unable to derive any value from them with SVMs.

6.2 Evaluation

We evaluate each of our decision detectors in 17-fold cross validations, where in each fold, the detector trains on 16 meetings and then tests on the remaining one. Evaluation can be made at three levels:

1. The sub-classifiers' detection of each of the DDA classes.

2. The sub-classifiers' detection of each of the DDA classes after correction by the super-classifier.

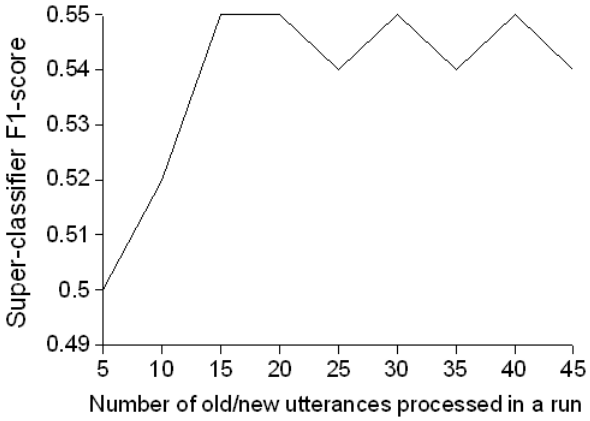

Figure 3: The relationship between the number of old/new utterances processed in a single run, and the super-classifier's F1-score. Here the sub-classifiers use only lexical features.

3. The super-classifier's detection of decision discussion regions.

For 1 and 2, we use the same lenient-match metric as (Fernández et al., 2008; Hsueh and Moore, 2007), which allows a margin of 20 seconds preceding and following a hypothesized DDA. Note that here we only give credit for hypotheses based on a 1-1 mapping with the gold-standard labels. For 3, we follow (Fernández et al., 2008; Purver et al., 2007) and use a windowed metric that divides the dialogue into 30-second windows and evaluates on a per window basis.

6.3 Results and analysis

Here, Section 6.3.1 will present results for different values of i, the number of old/new utterances processed in a single run. Section 6.3.2 then compares the performance of the real-time and off-line systems, (and also real-time systems which use hierarchical *vs.* flat classification), and Section 6.3.3 presents some feature analysis.

6.3.1 Varying the number of old/new utterances processed in a run

Figure 3 shows the relationship between i, the setting for the number of old/new utterances processed in a single run, and the super-classifier's F1-score. Here, the sub-classifiers are using only lexical features. We can see from the graph that as i increases to 15, the super-classifier's F1-score also increases, but thereafter, it plateaus. Hence 15 is apparently the value which best satisfies the two design goals given at the start of Section 6. It should also be noted that 15 is the mean length of a decision discussion in our data, and so per-

	sub-classifiers				super
	I	RP	RR	A	classifier
Re	.73	.73	.84	.71	.82
Pr	.08	.09	.03	.15	.40
F1	.15	.16	.06	.25	.54

Table 3: Results for the hierarchical real-time decision detector, using lexical, utterance and speaker features.

	sub-classifiers				super
	I	RP	RR	A	classifier
Re	.51	.51	.10	.63	.83
Pr	.12	.11	.04	.15	.41
F1	.19	.19	.05	.24	.55

Table 4: Results for the hierarchical off-line decision detector on off-line ASR transcripts, using lexical, utterance and speaker features.

haps this is a transferable finding. The mean duration of a run when $i = 15$ is approximately 4 seconds, while the mean duration of 15 utterances in our data-set is approximately 60 seconds, meaning that for the average case, the detector returns the results for the current run, long before it is due to make the next. Significant lee-way is perhaps necessary here, because the final version of the real-time detector will include a summarization component which extracts key phrases from *Issue/Resolution* utterances, and its processing can last some time, even for a single decision.

We should say then, that the system is not strictly real-time because in general, it detects decisions soon after they are made (for example within a minute), rather than immediately after. In the future we intend to modify the system so that it can run more frequently than once every i utterances. However it is important that runs do not occur too frequently — for example, if $i = 15$ and the system runs after every utterance, then the extra processing will cause it to gradually fall further and further behind the meeting.

6.3.2 Real-time *vs.* off-line results

Table 3 shows the results achieved by a hierarchical real-time decision detector whose run settings are as described above, and whose sub-classifiers[3] use lexical, utterance and speaker features. These results compare well with those of an equivalent off-line detector, which are shown in Table 4. The F1-scores for the real-time and off-line decision super-classifiers are $.54$ and $.55$ respectively, and the difference is not statistically significant. This may indicate that the hierarchical classification approach is fairly robust to increasing ASR Word Error Rates (WERs). Combining the output from each of the independent sub-classifiers might compensate somewhat for any decreases in their individual accuracy, as there was here for the *I* and *RP* sub-classifiers.

[3] In Tables 3 to 6, sub-classifier results are post-correction (see Section 6.2).

	sub-classifiers				super
	I	RP	RR	A	classifier
Re	.50	.51	.09	.63	.83
Pr	.11	.11	.03	.14	.41
F1	.19	.18	.05	.23	.55

Table 5: Results for the hierarchical off-line detector on real-time ASR transcripts, using lexical, utterance and speaker features.

	sub-classifiers				super
	I	RP	RR	A	classifier
Re	.67	.74	.84	.66	.85
Pr	.07	.08	.03	.14	.41
F1	.13	.15	.05	.24	.55

Table 6: Results for the hierarchical real-time decision detector, using lexical features only.

The hierarchical real-time detector's F1-score is also 10 points higher than a flat classifier ($.54$ *vs.* $.44$). Hence, while Fernández et al. (2008) demonstrated that the hierarchical classification approach could improve off-line decision detection, we have demonstrated here that it can also improve real-time decision detection.

Table 5 shows the results when an off-line detector is applied to real-time ASR transcripts. Here, the super-classifier obtains an F1-score of $.55$, one point higher than the real-time detector, but again, the difference is not statistically significant.

6.3.3 Feature analysis

We also investigated the contribution of the utterance and speaker features. Table 6 shows the results for the hierarchical real-time decision detector when its sub-classifiers use only lexical features. The sub-classifier F1-scores are all slightly lower than when utterance and speaker features are used (see Table 3), and the super-classifier

score is only 1 point different. None of these differences are statistically significant.

Since lexical features are important, we used *information gain* in order to investigate which words are predictive of each DDA type. Due to differences in the transcripts, the predictive words for the off-line and real-time systems are not the same, but we can find commonalities, and these commonalities make sense given the DDA definitions. Firstly in *Resolution* and particularly *Issue* DAs, some of the most predictive words could be used to define discussion topics, and so we might expect to find them in the meeting agenda. Examples are "energy", and "color". Predictive words for *Resolutions* also include semantically-related words which are key in defining the decision ("kinetic","green"). Additional predictive words for *RPs* are the personal pronouns "I" and "we", and the verbs, "think" and "like", and for *RRs*, words which we would associate with summing up ("consensus", "definitely", and "okay"). Unsurprisingly, for *Agreements*, "yeah" and "okay" both score very highly.

7 Conclusion

(Fernández et al., 2008) described an approach to decision detection in multi-party meetings and demonstrated how it could work relatively well in an off-line system. The approach has two defining characteristics. The first is its use of an annotation scheme which distinguishes between different utterance types based on the roles which they play in the decision-making process. The second is its use of hierarchical classification, whereby binary *sub-classifiers* detect instances of each of the decision DAs (DDAs), and then based on the sub-classifier hypotheses, a *super-classifier* determines which regions of dialogue are decision discussions.

In this paper then, we have taken the same basic approach to decision detection as Fernández et al. (2008), but changed the way in which it is implemented so that it can work effectively in real-time. Our implementation changes include running the detector at regular and frequent intervals during the meeting, and reprocessing recent utterances in case a decision discussion straddles these and brand new utterances. The fact that the detector reprocesses utterances means that on consecutive runs, overlapping and duplicate hypothesized decision discussions are possible. We have therefore added facilities to merge overlapping hypotheses and to remove duplicates.

In general, the resulting system is able to detect decisions soon after they are made (for example within a minute), rather than immediately after. It has performed well in testing, achieving an F1-score of .54, which is only one point lower than an equivalent off-line system, and in any case, the difference was not statistically significant. A flat real-time detector achieved .44.

In future work, we plan to extend the decision discussion annotation scheme and try to extract supporting arguments for decisions. We will also experiment with using sequential models in order to try to exploit any sequential ordering patterns in the occurrence of the DDAs.

Acknowledgements This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004, and by the Department of the Navy Office of Naval Research (ONR) under Grants No. N00014-05-1-0187 and N00014-09-1-0106. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or ONR. We are grateful to the three anonymous EMNLP reviewers for their helpful comments and suggestions, and to our partners at SRI International who provided us with off-line and real-time transcripts for our meeting data.

References

Satanjeev Banerjee, Carolyn Rosé, and Alex Rudnicky. 2005. The necessity of a meeting recording and playback system, and the benefit of topic-level annotations to meeting browsing. In *Proceedings of the 10th International Conference on Human-Computer Interaction*.

Susanne Burger, Victoria MacLaren, and Hua Yu. 2002. The ISL Meeting Corpus: The impact of meeting type on speech style. In *Proceedings of the 7th International Conference on Spoken Language Processing (INTERSPEECH - ICSLP)*, Denver, Colorado.

Raquel Fernández, Matthew Frampton, Patrick Ehlen, Matthew Purver, and Stanley Peters. 2008. Modelling and detecting decisions in multi-party dialogue. In *Proc. of the 9th SIGdial Workshop on Discourse and Dialogue*.

Michel Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. 2004. Identifying agree-

ment and disagreement in conversational speech: Use of Bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Daniel Gatica-Perez, Ian McCowan, Dong Zhang, and Samy Bengio. 2005. Detecting group interest level in meetings. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Dustin Hillard, Mari Ostendorf, and Elisabeth Shriberg. 2003. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Companion Volume of the Proceedings of HLT-NAACL 2003 - Short Papers*, Edmonton, Alberta, May.

Pey-Yun Hsueh and Johanna Moore. 2007. Automatic decision detection in meeting speech. In *Proceedings of MLMI 2007*, Lecture Notes in Computer Science. Springer-Verlag.

Adam Janin, Jeremy Ang, Sonali Bhagat, Rajdip Dhillon, Jane Edwards, Javier Marcías-Guarasa, Nelson Morgan, Barbara Peskin, Elizabeth Shriberg, Andreas Stolcke, Chuck Wooters, and Britta Wrede. 2004. The ICSI meeting project: Resources and research. In *Proceedings of the 2004 ICASSP NIST Meeting Recognition Workshop*.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*. MIT Press.

Agnes Lisowska, Andrei Popescu-Belis, and Susan Armstrong. 2004. User query analysis for the specification and evaluation of a dialogue processing and retrieval system. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*.

Iain McCowan, Jean Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. 2005. The AMI Meeting Corpus. In *Proceedings of Measuring Behavior, the 5th International Conference on Methods and Techniques in Behavioral Research*, Wageningen, Netherlands.

Matthew Purver, John Dowding, John Niekrasz, Patrick Ehlen, Sharareh Noorbaloochi, and Stanley Peters. 2007. Detecting and summarizing action items in multi-party dialogue. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium.

Andreas Stolcke, Xavier Anguera, Kofi Boakye, Özgür Çetin, Adam Janin, Matthew Magimai-Doss, Chuck Wooters, and Jing Zheng. 2008. The ICSI-SRI Spring 2007 meeting and lecture recognition system. In *Proc. of CLEAR 2007 and RT2007*.

Daan Verbree, Rutger Rienks, and Dirk Heylen. 2006. First steps towards the automatic construction of argument-diagrams from real discussions. In *Proceedings of the 1st International Conference on Computational Models of Argument*, volume 144, pages 183–194. IOS press.

A. Waibel, T. Schultz, M. Bett, M. Denecke, R. Malkin, I. Rogina, R. Stiefelhagen, and J. Yang. 2003. SMaRT: The smart meeting room task at ISL. In *ICASSP*.

Steve Whittaker, Rachel Laban, and Simon Tucker. 2006. Analysing meeting records: An ethnographic study and technological implications. In S. Renals and S. Bengio, editors, *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005, Revised Selected Papers*, volume 3869 of *Lecture Notes in Computer Science*, pages 101–113. Springer.

Britta Wrede and Elizabeth Shriberg. 2003. Spotting "hot spots" in meetings: Human judgements and prosodic cues. In *Proceedings of the 9th European Conference on Speech Communication and Technology*, Geneva, Switzerland.