Leveraging Minimal User Input to Improve Targeted Extraction of Action Items

Matthew Frampton, Raquel Fernández, Patrick Ehlen, Anish Adukuzhiyil and **Stanley Peters**
Center for the Study of Language and Information
Stanford University
{frampton,raquel,ehlen,ajohna,peters}@stanford.edu

Abstract

In face-to-face meetings, assigning and agreeing to carry out future actions is a frequent subject of conversation. Work thus far on identifying these action item discussions has focused on extracting them from entire transcripts of meetings. Here we investigate a *human-initiative targeting* approach by simulating a scenario where meeting participants provide low-load input (pressing a button during the dialogue) to indicate that an action item is being discussed. We compare the performance of categorical and sequential machine learning methods and their robustness when the point of user input varies. We also consider automatic summarization of action items in cases where individual utterances contain more than one type of relevant information.

1 Introduction

Regrettably, people do not always pay attention to everything you say. In fact, research on lexical change blindness suggests they miss more than you might imagine (Sanford et al., 2006). But such attention-constraining strategies can prove adaptive in the face of so-called "information overload," and the myriad pressures on attention that arise from living in the modern era (or, perhaps, any era). For example, an effective attention strategy during a business meeting might be to pay close attention to e-mail on your laptop while processing the ongoing meeting dialogue in a shallow way that picks up on segments of interest to you, or in which it seems you are about to be assigned some task. When those patterns of dialogue arise, you then pay closer attention to the dialogue, or even participate yourself. Such strategies come second nature to us.

But this strategy of targeted listening can be employed in machine interpretation of meeting dialogue as well, using an approach to dialogue processing we call *targeted understanding*. While a machine's interpretation of semantics in multi-human dialogue faces different obstacles from those faced by a human—lacking the facility with context and intentionality that we take for granted—the general approach to interpretation can be similar: Only segments that contain certain patterns of dialogue are identified as deserving close attention, followed by a deeper semantic analysis of those segments for the most relevant bits of information.

In this paper we briefly discuss how we use targeted understanding to identify the tasks people agree to in meetings (their *action items*) from multiparty meeting dialogue. Work thus far on this endeavor has focused on extracting action items from entire records of meetings (Purver et al., 2007; Ehlen et al., 2008), relying on a machine-initiative approach that extracts all possible action item discussions and then asks meeting participants to cull them after the meeting is finished. Here we will steer a slightly different tack, investigating the potential of "human-initiative targeting" that allows participants in a meeting to give some indication of an area of interest—by, say, pressing a button when an action item is being discussed. We then use automatic methods to extract the semantic properties of utterances that are salient to that segment of dialogue, and generate a readable summary.

In the next Section we describe previous work on extracting action items. After that, in Section 3, we present our approach and methodology for this study. Sections 4 and 5 present our experiments and results: First with respect to the task of detecting those utterances that contain semantic information related to action items; and second with respect to extracting different kinds of properties from single utterances that contain more than one type of action item-related semantic information. We conclude with directions for future work in Section 6.

2 Targeted Understanding of Action Items

The process of assigning and agreeing to carry out future actions frequently arises through some channel of communication, such as e-mails or dialogue. They are often called *action items* or *next actions* and arise as *public commitments* to undertake a task. Several recent efforts have sought to utilize this communicative channel to extract them automatically, and to mine and summarize useful information from them.

2.1 Action Items in Dialogue

How do people in meetings discuss action items? Because the process of deciding what tasks will be done and who will do them is a common and significant interaction during meetings, their discussion approximates an *exemplary structure*, adhering to a recognizable pattern—even if that pattern comes spread over several persons and multiple utterances.

(1) A: We should have a rerun of the three
of us sitting together
B: Sure
A: Some time this week again
C: OK
A: And finish up the values of this
B: Yeah

In the first place, there is usually some discussion of the task that needs to be performed. In the example above the sub-utterances *"have a rerun of the three of us sitting together"* and *"finish up the values"* contribute to a *task description*. The first utterance also includes a second component that is commonly found in action items, which is discussion of who will be responsible for—or take *ownership* of—the task to be performed (in this case, all participants, or *"we"*). A third component is some designation of the *timeframe* in which the task should be completed, in this case *"some time this week"*. Finally, because this is a public, joint commitment and not a solitary one, one often hears some indication of agreement from the participants agreeing to the commitment (*"Sure", "OK", "Yeah"*) Because acknowledgments like these help to glue together verbal acts of coordination, *agreement* is an important fourth component in such discussions.

Thus, a dialogue that discusses an action item tends toward some approximation of this exemplary structure, and includes utterances that play one or more of these four roles at a time. Granted, the structure is exemplary, so sometimes one of these elements (such as the timeframe) may not be present. But in general, the closer a round of dialogue comes to representing these four types of dialogue moves—*task description*, *ownership*, *timeframe*, and *agreement*—the more likely we find that some future task or action item is being discussed.

2.2 Structural Extraction Approach

This structural insight was fleshed out in Purver et al. (2006; 2007). Others (Morgan et al., 2006; Hsueh and Moore, 2007) had attempted a *flat* approach to action item detection in dialogue where utterances were simply marked as either being relevant to an action item discussion or not. Purver et al. (2007) replaced this flat classification approach with a structured, hierarchical one. They trained four linear Support Vector Machine (SVM) classifiers to detect utterances that correspond to each of the four Action Item-related Dialogue Acts (AIDAs) in Table 1. Then they used a *super-classifier* trained with the hypothesized labels and confidence scores of the four independent classifiers to detect clusters of those sub-classes, which indicate probable discussions of action items. On the task of detecting action item discussions, this approach achieved an F-score of 0.45, (using a criterion of at least 50% overlap between hypothesized and oracle action item discussion), compared to 0.35 using a flat approach with the same feature and data sets.

The strategy of attending to and targeting a specific dialogic structure exhibits a clear benefit over a flat approach. But note that this approach to hierarchical classification does not presume any sequential dependencies in the utterances, since they are classified separately and aggregated by window, thus

D	*description*	discussion of the task to be performed
T	*timeframe*	discussion of the required timeframe
O	*owner*	assignment of responsibility (to self or other)
A	*agreement*	explicit agreement or commitment

Table 1: Action item dialogue act (AIDA) classes.

ignoring any temporal organization that might exist in the exemplary pattern of action item discussions. This is one possibility we intend to investigate here.

2.3 Exploiting User Feedback

Another way to improve detection of action item discussions and their associated AIDAs is to involve a person in the loop who can provide some feedback about whether or not the detected utterances really do correspond to discussion of an action item.

This possibility was explored by Ehlen et al. (2007; 2008), who used a post-meeting browser tool to present detected action items to meeting participants taken from the DARPA CALO 2007 CLP evaluation. After each meeting, participants could review their action items, changing the *task description* (D), *timeframe* (T), and *owner* (O) entries in ways that allowed feedback to three of the four corresponding AIDA sub-classifiers. When users added action items to their to-do lists or rejected them, feedback for the super-classifier was also harvested.

These data from human feedback were used to re-train each of the targeted classifiers, allowing an assessment of whether implicit user feedback could help improve the models. Indeed, this type of feedback yielded F-score error reductions between 20 and 40% for different meeting sequences, indicating that human feedback could be useful.

Results such as these bring up the question of whether some other types of human input might yield similar improvements. Instead of requiring meeting participants to review action items after a meeting is finished, perhaps they could "mark" relevant segments of a meeting as they happen, by, for example, pushing a button when something occurs that corresponds to information they wish to recall or have extracted. Our first experiment in Section 4 simulates just such a scenario.

2.4 Summarizing Action Items

There is a growing interest in dialogue summarization, with most approaches attempting to summarize the content of entire dialogues (Zechner, 2002; Murray et al., 2005; Murray and Renals, 2007). The most obvious application of identifying action item discussions and their corresponding dialogue acts is to produce a more structured and targeted meeting summary by providing a descriptive record of the tasks assigned, perhaps presented as an automatically generated to-do list.

Purver et al. (2007) made a preliminary attempt at generating extractive summaries of action items, focusing on utterances tagged as performing one of two AIDAs: either the *task description* (D) or the *timeframe* (T) during which the task is to be performed. Their approach involved parsing the word confusion network (WCN) for each relevant utterance using a general rule-based parser (Dowding et al., 1993), which produced multiple short fragments rather than one full utterance parse. An SVM classifier was then trained to learn a model which ranked these phrases according to their likelihood of appearing in a gold-standard extractive summary. Various features were used including WCN, parse, lexical and temporal expression tags.

This approach produced mixed results. While precision was higher than that of a baseline that used the entire 1-best utterance transcription, only the F-scores obtained for *timeframe* outperformed the baseline. Besides yielding mixed results, this prior work did not consider summarisation of action items where utterances are tagged with multiple AIDA classes. In such cases, it is necessary to determine which bits of information are related to which dialogue act, and as a result the summarization task becomes more complicated. Our second experiment in Section 5 addresses this issue.

3 Approach & Methodology

The work of Purver et al. (2007) has shown that automatically identifying AIDAs in transcripts of full meetings is a difficult task—achieving F-scores below 0.25 (see Table 2). One reason is that AIDAs are very sparse, making up only around 1.4% of utterances in a meeting transcript. In the first of two experiments, we want to investigate how on-line input given by meeting participants can reduce the sparseness problem and thus help in automatic identification. If participants could indicate where an action item is being discussed by, for instance, pressing a button during the ongoing dialogue, such "human-initiative targeting" could help the system to bypass large sections of dialogue in favor of specific, relevant regions.

We simulate participants' input by selecting sections of dialogue that include discussion of action items, and then use machine learning on the targeted sections to identify the AIDAs. In doing so we address a number of issues.

First, we investigate the degree to which human-initiative targeting can improve classifier performance by training only on windows of utterances instead of full meetings. The average length of an action item discussion is 7.8 utterances, and 92% of action items are at most 15 utterances long. Hence we allow the system to have access only to 15 utterance windows.

Secondly, we compare the performance of a Support Vector Machine (SVM) categorical classifier, as used by Purver et al. (2007), against a Hidden Markov model (HMM). The HMM is a sequential model, and so assuming that action item discussions exhibit regularities in sequences of utterance types, it may perform better.

Thirdly, we also investigate how robust classifier performance is with regard to when the human input is given. We first consider a case in which participants always press a button right at the end of an action item discussion, and then look at a presumably more realistic case in which participants may press the button at different times in relation to the end of an action item discussion. This allows us to investigate the extent to which performance degrades in less systematic and more realistic situations.

Our second experiment is concerned with extracting words from AIDAs that can be used to generate a useful descriptive summary of an action item discussion. As mentioned in the previous section, the fact that utterances can be tagged with multiple AIDAs complicates the task of extracting information for summarization purposes, since we need to distinguish between bits of information related to different AIDAs but contained within a single utterance. We address this issue in Section 5, focusing on those utterances that have been simultaneously tagged with classes D and O. Again, we compare performance of categorical (SVM) and sequential (HMM) classifiers.

For our two experiments, we used the ICSI Meeting Corpus (Janin et al., 2004), which contains recordings and manual transcriptions of naturally occurring research group meetings. In particular, we used the annotated sub-corpus of Purver et al. (2007), which consists of 18 ICSI meeting transcripts annotated using the AIDA classes shown in Table 1. The annotations also include a summary description for every instance of an AIDA class, created by manual selection of words and phrases from the gold-standard transcripts.

4 Experiment I: Targeted AIDA Detection

In this section we present our experiment on detecting AIDAs from targeted regions of meeting transcripts.

4.1 Data

The 18 ICSI meetings in our subcorpus have been annotated with 190 action item discussions in total (10.6 action items per meeting on average). To simulate user input, we generated two different data-sets from this corpus: a *systematic input* data-set and a *non-systematic input* data-set. The systematic input data-set was generated by extracting 190 sections of 15 utterances, and for each the last utterance corresponded to the last AIDA of an action item. This data-set simulates a scenario where participants always press a button right at the end of an action item discussion. The non-systematic data-set simulates a more realistic situation where user input is given at random points towards the end of an action item discussion. Here we allow the system to look 10 utterances backwards and 5 forward from the point when the input is given. The data-set was generated by extracting 190 sections of 15 utterances, where the input is assumed to be randomly given either immediately after the last AIDA of an action item discussion, or 1, 2, 3 or 4 utterances earlier.

Targeting sections of dialogue that contain action item discussions obviously reduces AIDA sparseness considerably. Averaging over the *systematic* and *non-systematic input* data-sets (which are very similar in this respect), 13.7% of utterances (around 2 on average per window) are tagged with class D, 4.4% (around 0.6 per window) are tagged with class T, 9.5% (around 1.4 per window) are tagged with class O, and 14.6% (around 2.2 per window) are tagged with class A.

4.2 Classifiers & Features

We use the linear-kernel support vector machine classifier SVM*light* (Joachims, 1999) and the structural support vector machine classifier SVM*hmm* (Altun et al., 2003), which trains models that are isomorphic to hidden Markov models.

We train four individual SVM classifiers—one for each AIDA class—and compare their performance to that of one single HMM classifier that uses six different labels for the model states: labels D, T, O, and A for each of the AIDA classes, plus a label X for utterances outside the action item discussion and an insertion-class label I for those utterances inside an action item discussion that do not belong to any AIDA class. In all cases, we evaluate performance using 18-fold cross-validation, with each fold containing those 15-utterance windows that belong to the same meeting.

To train the classifiers, we use similar features to those of Purver et al. (2007), derived from the properties of the utterances in context: lexical unigrams, durational features from the transcriptions, dialogue act tags from the ICSI-MRDA annotations (Shriberg et al., 2004), temporal expression tags using the MITRE TIMEX tool, as well as contextual features consisting of the same features for the immediately preceding and following 5 utterances.

4.3 Results

The results reported in Purver et al. (2007) for the task of identifying AIDAs from whole meetings are shown in Table 2. Using simulated participant input to target regions of dialogue that contain action item discussions, we are able to improve these baseline results by more than 30% (see Table 3).

Table 3 shows the scores we obtained when simulated participant input was provided, systematically at the end of an action item discussion and non-systematically at any point in the second half of the discussion. In this case the results for these two different data-sets are very similar. The non-systematic input data-set yields slightly lower F-scores, but the drop is only statistically significant for classes D and A ($p < 0.05$ on a paired t-test). The slightly lower results may be due to the fact that some AIDAs may not fall into the 15 utterance window the classifier is looking at (for instance, if the input is given at the end of the action item and the discussion is more than 10 utterance long, then since the classifier is only looking 10 utterances back, the AIDA(s) at the beginning of the action item discussion are not considered), which reduces the number of available positive examples.[1]

	D	T	O	A
Recall	.19	.15	.21	.18
Precision	.18	.46	.27	.16
F-score	.19	.22	.24	.17

Table 2: SVMs trained on whole meeting transcripts

	D	T	O	A
Recall	.66	.57	.66	.78
Precision	.51	.45	.51	.49
F-score	.57	.51	.57	.60
Recall	.56	.52	.62	.82
Precision	.45	.45	.50	.44
F-score	.50	.48	.55	.57

Table 3: SVMs trained on targeted regions; systematic input (top) vs. non-systematic input (bottom)

Table 4 shows the results we obtained when we used a single HMM instead of four independent SVM classifiers. While recall is significantly lower for all classes ($p < 0.05$) leading to a drop of F-scores, the sequential model is able to achieve good precision results. In contrast to the SVMs, however, using the non-systematic input data-set with the HMM classifier leads to a statistically significant drop in performance, especially for classes A and D, where both recall and precision decrease ($p < 0.01$). This is perhaps not surprising, since the variability of the non-systematic data-set disrupts the sequen-

[1] A possible way of compensating for this would be to increase the size of the window. This however is not an optimal solution since the bigger the window the sparser the AIDAs.

	D	T	O	A
Recall	.48	.33	.45	.54
Precision	.53	.52	.50	.53
F-score	.50	.40	.48	.53
Recall	.32	.22	.32	.38
Precision	.45	.41	.46	.40
F-score	.37	.29	.38	.39

Table 4: HMM trained on targeted regions; systematic input (top) vs. non-systematic input (bottom)

tial organization that drives this kind of model.

While lexical features were the most useful in all cases, we observed that using the MRDA dialogue act tags `commitment` and `suggestion` improved precision significantly, especially for classes O and D. TIMEX tags boost scores for class T, although using targeted regions does not improve precision for this class.

In summary, using online input to target regions of dialogue where an action item is being discussed can improve AIDA detection substantially when compared to a no-input approach, even if the input is given randomly towards the end of the action item discussion. Although the sequential model yielded good precision scores, its performance was less robust to non-systematic user input. A possible reason for its lower recall even with the systematic data-set is that HMMs may not be so well suited when target classes are sparse:[2] if the model fails to hypothesize one AIDA where it should, it may then fail to hypothesize subsequent AIDAs. SVMs do not have this problem because each utterance is assessed independently.

5 Experiment II: Summarization of Utterances Tagged with Multiple AIDAs

Having indentified the constituent utterances in an action item, the next task is to summarize their action item-related semantic content so that it can be presented in a to-do list for the user. Here, we use a different methodology from Purver et al. (2007) that does not require a parser, and concentrate on extracting summary-worthy words from utterances that have been tagged with multiple AIDAs. While in general there is a large degree of independence between class distributions (with most cosine distances below 0.3), classes D and O often overlap, yielding a between-class cosine distance of 0.55 (where 1 represents exact correlation and 0 total independence). Hence we concentrate on those utterances that have been tagged as both *ownership* (O) and *task description* (D).

5.1 Methodology

In our 18 meeting corpus there are 162 utterances that have been tagged as both D and O. These utterances contain a total of 2697 words, 409 of which have been annotated as summary-worthy for class O, and 1015 as summary-worthy for class D. Example (2) shows a D + O utterance with the gold-standard summary-worthy phrases indicated in square brackets.

(2) It would be great if [*you*]$_O$ could um not transcribe it all but uh [*pick out some stuff*]$_D$

We use gold-standard extractive summaries as targets and train a classifier to decide whether or not each word in the manual transcription of a D + O utterance is summary-worthy for classes O and D, respectively. This approach exploits the fact that critical phrases that contain summary-worthy information for different AIDAs display characteristic syntactic, semantic, and lexical features.

To train our classifiers we used lexical trigrams (including the current word, and the immediately preceding and following words) and Part-of-Speech (PoS) tags generated by the Stanford PoS tagger (Toutanova and Manning, 2000). In all cases, testing was performed using 10-fold cross-validation. We experimented with the following types of classifiers:

- SVM: Two independent classifiers each trained to distinguish O and D words, respectively, from other words.
- SVM (O/D): One classifier trained to distinguish between O, D, and other words.
- HMM (B/I): One classifier trained to distinguish between O words (beginning and inside of sequence), D words (beginning and inside of sequence), and other words.[3]

[2] As mentioned in Section 4.1, AIDA classes in targeted regions make up between 4.4% and 14/6% of utterances.

[3] The end of the sequence is labelled with the inside (I) tag.

5.2 Evaluation

We evaluated each classifier's performance against the manually-annotated summary descriptions. Recall was therefore the proportion of words in the gold-standard summaries which overlapped with the words extracted by the classifiers; precision was the proportion of words extracted by the classifiers which also appear in the gold-standard summaries.

The O and D classes are compared to different baselines. Since the role of the O class is to assign responsibilty for a task, a large number of utterances tagged with O contain names or pronouns identifying the responsible party. Hence it is reasonable to use a baseline which tags all instances of first and second person personal pronouns (*I, you, we*) as positive. For class D, there was no clear majority POS class, so we settled on a baseline that tagged half of all words in D utterances as positive, where this half was selected randomly.

5.3 Results

Table 5 shows results for the different classifiers. All of the classifiers achieved substantially higher F-scores than the baseline for both *ownership* (O) and *task description* (D).

	Ownership			Description		
Model	Re	Pr	F1	Re	Pr	F1
Baseline	.39	.59	.47	.53	.38	.44
SVM	.76	.56	.64	.80	.64	.71
SVM (O/D)	.61	.67	.64	.74	.68	.71
HMM (B/I)	.61	.69	.65	.74	.71	.73

Table 5: Extraction of summary-worthy O/D words

For O, all of the classifiers achieved very similar F-scores. However a t-test shows that the HMM's score is significantly higher than the SVM(O/D) ($p < 0.005$). For D, the HMM performed significantly better than both the SVM(O/D) and SVM(D) classifiers in terms of precision and F-score ($p < 0.01$). Its F-score of .73 is much higher than that achieved by the best model of Purver et al. (2007): .38, lower even than their baseline which was the entire 1-best utterance transcription (see Section 2.4). Although those results are not directly comparable to ours, (since we used gold-standard transcriptions rather than WCNs, and focused on utterances that had been tagged with 2 rather than 1 AIDA class), we believe they show that the general approach has promise, and that the sequential model is well-suited to this task.

6 Conclusions & Future work

We have simulated a "human-initiative targeting" approach to action item detection where participants provide input—e.g. by pressing a button—to indicate that an action item is being discussed, which allows a system to concentrate on relevant dialogue regions. As a result we were able to improve the detection of action item-related dialogue acts (AIDAs) very substantially, obtaining F-scores that are twice as high as when using whole meetings.

Categorical models (SVM) proved to be more useful than sequential ones (HMM) for this task. The HMM yielded good precision scores but significantly lower recall, and so the overall performance was lower for this type of classifier. When we compared systematic user input given at the end of an action item discussion with less systematic input given randomly at different points towards the end of the action item, we found that the SVMs were more robust than the sequential model. This is not surprising since such unsystematic behavior disrupts the sequential organization which the HMM relies on.

We also addressed the task of extracting summary-worthy information from utterances that had been tagged with two AIDAs—*ownership* and *task description*—and found sequential models to be useful for this task, achieving F-scores of .65 and .73, respectively.

In the future we plan to experiment with a two-stage classification approach. This would involve first using SVMs to make classifications and provide confidence scores independent of sequence, and then second, giving this information to a sequential model that makes the final classifcations. Combining the two different types of classifier in this way may produce better results for both AIDA classification and summarization.

Our findings with respect to targeted understanding are useful, but of course, real user behavior during actual meetings will differ in many respects, and will surely prove more variable than what we have simulated here. Bearing this in mind, future work will involve conducting an experiment in which we ask actual meeting participants to provide live button-pushing input during meetings when it occurs to them that an action item is being discussed. Only then can we know whether the approach described in this paper will be robust enough to handle the vagaries of real human behavior.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-01850004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA, or the Air Force Research Laboratory.

References

Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Hidden Markov support vector machines. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*.

John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran. 1993. GEMINI: a natural language system for spoken-language understanding. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Patrick Ehlen, Matthew Purver, and John Niekrasz. 2007. A meeting browser that learns. In *Proceedings of the AAAI Spring Symposium on Interaction Challenges for Intelligent Assistants*.

Patrick Ehlen, Matthew Purver, John Niekrasz, Kari Lee, and Stanley Peters. 2008. Meeting adjourned: Offline learning interfaces for automatic meeting understanding. In *Proceedings of the International Conference of Intelligent User Interfaces*, Canary Islands, Spain.

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.

William Morgan, Pi-Chuan Chang, Surabhi Gupta, and Jason M. Brenier. 2006. Automatically detecting action items in audio meeting recordings. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 96–103, Sydney, Australia.

Gabriel Murray and Steve Renals. 2007. Towards online speech summarization. In *Proceedings of INTERSPEECH 2007*, Antwerp, Belgium.

Gabriel Murray, Steve Renals, and Jean Carletta. 2005. Extractive summarization of meeting recordings. In *Proceedings of the 10th European Conference on Speech Communication and Technology (INTERSPEECH - EUROSPEECH)*.

Matthew Purver, Patrick Ehlen, and John Niekrasz. 2006. Detecting action items in multi-party meetings: Annotation and initial experiments. In *MLMI 2006, Revised Selected Papers*, Lecture Notes in Computer Science. Springer.

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.

Alison J. S. Sanford, Anthony J. Sanford, Jo Molle, and Catherine Emmott. 2006. Shallow processing and attention capture in written and spoken discourse. *Discourse Processes*, 42(2):109–130.

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100, Cambridge, Massachusetts.

Kristina Toutanova and Christopher Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*.

Klaus Zechner. 2002. Automatic summarization of open-domain multiparty dialogues in diverse genres. *Computational Linguistics*, 28(4):447–485.