TOWARDS AUTOMATIC ARGUMENT DIAGRAMMING OF MULTIPARTY MEETINGS

Dilek Hakkani-Tür

International Computer Science Institute (ICSI), Berkeley, CA
dilek@icsi.berkeley.edu

ABSTRACT

This paper focuses on a lesser studied multiparty meetings processing task of argument diagramming. Argument diagramming aims at tagging the utterances and their relationships to represent the flow and structure of reasoning in conversations, especially in discussions and arguments. In this work, we tackle the problem of automatically assigning node types to user utterances using several lexical and prosodic features. We performed experiments using the AMI Meeting Corpus annotated according to the the Twente Argumentation Schema. Our results indicate that while lexical and prosodic features both provide orthogonal information for this task, using a cascaded approach, eliminating backchannel utterances improves the performance. With this final approach, when all features are used, we achieve about 9% relatively better error rates than a simpler classifier based on only lexical features.

Index Terms— argument mapping, classification, lexical and prosodic features, multiparty meeting processing.

1. INTRODUCTION

Meetings form an essential part of information exchange and communication in organizations. In the recent years, the availability of multiparty meeting corpora with annotations, such as the ICSI [1] and AMI [2] meeting corpora, allowed speech and language processing research in several areas including automatic transcription [3], dialog act segmentation [4] and tagging [5], and summarization [6] of meeting content.

In this paper, we focus on a lesser studied meetings processing task of argument diagramming. Argument diagramming aims to display a visual representation of the flow and structure of reasoning in conversations, especially in discussions and arguments [7]. To this end, the utterances and their relationships are tagged with predefined classes representing the characteristics of the discussion and argumentation. For example, one utterance may open a new issue and another utterance replying to that one may elaborate on that. In the argument diagrams, typically, utterances are represented via typed nodes, and the relationship of utterances with other related utterances are shown via typed edges connecting two nodes, forming a tree structure for the topics discussed.

Argument diagrams extracted from meetings can be useful for meeting participants, to help them in following discussions and catch up with arguments, if the maps can be extracted during the meeting [8]. There is a wide body of work that focuses on visualization of argument maps, as entered by the conversation participants [8, 9]. Argument diagrams can also help users in browsing past meetings, tracking progress across several meetings and can be useful in meeting summarization. [10] have performed experiments with human subjects, and their results indicated that argumentation information from meetings can be useful in question answering. Argument diagrams can also help the related tasks of action item extraction [11, 12] and decision detection [13] in meetings. Note that argument diagramming is different than decision detection in several ways, the most important one is that not all discussions are required to include a decision.

For the multiparty meetings domain, two studies proposed argumentative models of meeting discussion. Combining their experience from two meeting processing projects, DARPA CALO and Swiss National Research project IM2, Pallotta *et al.* discussed four perspectives (persuasion, decision making, episodes, and conversations), and a theoretical model for each perspective [14]. Similarly, Rienks *et al.* proposed the Twente Argumentation Schema (TAS), and annotated the AMI meeting corpus according to TAS [7]. In this paper, we use the TAS and the corresponding argument diagram annotations on the AMI meeting corpus [2]. In this representation, there are six node types, and nine relation types, which are described in more detail in the next section. The relations apply to specific node type pairs. In this work, we only tackle the problem of assigning node types to user utterances, and study the use of several lexical and prosodic features for this task. Previously, Galley *et al.* used lexical, durational and structural features with Bayesian networks, to detect agreement and disagreements in conversations [15]. Murray *et al.* investigated the use of prosodic features to detect rhetorical relations, that aim to describe conversations in terms of coherence [16]. Rienks and Verbree used decision trees with features extracted from manual annotations, such as the presence of a question mark, utterance length, label of the preceding segment, and automatically computed features such as part of speech tags to investigate the learnability of argument diagram node types[10].

Section 2 describes TAS, along with examples from meetings from the AMI corpus. Section 3 describes our approach that is based on lexical and prosodic features. Section 4 presents results of preliminary experiments for node type detection, as well as an analysis of feature usage.

2. THE TWENTE ARGUMENT SCHEMA (TAS)

TAS was created at University of Twente, where argument diagrams for parts of meeting transcripts that contain discussions around a specific topic, were also formed [17]. In TAS, argument diagrams are tree-structured; the nodes of the tree contain speech act units (usually parts of or complete speaker turns) and the edges show the relations between the nodes, the edges emanate from parents and end at children nodes, where the children nodes follow parent nodes in time. At a high level, there are two types of nodes: *issues and statements*. The *issue* nodes mainly open up an issue and request a response and are further categorized into three depending on the form of the response they expect: *open issue* (OIS), *A/B issue* (AIS) and Yes/No issue (YIS). The *open issues* are utterances that allow for

TYPE	EXAMPLE
STA	And you keep losing them.
WST	We should probably just use conventional batteries.
OIS	What's the functionality of that?
AIS	So, double or triple?
YIS	Do we need an LCD display?
OTHER	Mm-hmm.

Table 1. Examples of utterances that belong to statement (STA), weak statement (WST), open issue (OIS), A/B issue (AIS), Yes/No issue (YIS), and OTHER node types.

various possible responses, that are not included in the utterances themselves. In contrast, *A/B issues* are utterances that request possible responses that are specified in the utterance. The *Yes/No issues* directly request the other participants' opinion as a "Yes" or "No". The *statements* are utterances that convey the position of the speaker on a subject/topic. To be able to represent the statements for which the speaker is not highly certain about what they say, the *statements* are split into two: *statements* (STA) and *weak statements* (WST). The *weak statements* represent the cases where the speaker is not very confident. The rest of the utterances that are not involved in reasoning or backchannelling utterances are represented with an additional (OTHER) category. Table 1 shows example utterances for each node type.

The relations between a pair of utterances are categorized into nine types: *Elaboration*, *Specialization*, *Request*, *Positive*, *Negative*, *Uncertain*, *Option*, *Option Exclusion*, and *Subject To*. As its name implies, *Elaboration* relation applies to the pair of utterances (both which can be statements or issues), where the child node utterance elaborates on the parent node utterance. Similarly, the *Specialization* relation applies to pairs (statements and statements or issues and issues), where the child node is a specialization of the parent node. The *Request* relation relates two utterances (statements to issues), where the child utterance asks for more information about the parent. The *Positive* and *Negative* relations apply to utterances, where the child utterance supports or refutes the parent utterance, respectively. The *Uncertain* relation applies to pairs, where it is not clear if the child supports or refutes the parent node. The *Option* relation relates pairs of utterances (statements to issues or other statements), where the where the child is a possible answer, option or solution to the parent utterance. The *Option Exclusion* relates pairs (statements or issues to issues), where the child node eliminates one or more of the possible answers, options or solutions to the parent utterance. The *Subject To* relation applies to pairs (statements and Yes/No or A/B issues or statements), where the child provides criteria that need to be fulfilled before the parent node can be supported or denied.

More information about the relation types and example utterance pairs and annotated tree structures can be found in [7] and in the annotation guidelines [17]. One important thing to note about relations is, they usually relate pairs of utterances of specific node types. Therefore, the detection of node types before determining the relations is intuitively the processing sequence that we follow in this work for extracting argument diagrams from conversations, while joint modeling techniques should also be investigated in the future.

3. AUTOMATIC DETECTION OF NODE TYPES

We consider the automatic annotation of node types as a multi-class utterance classification problem. More formally, given an utterance $x_i \in X$, the problem is to associate a class $c_i \in C$ with x_i where C is the finite set of argument diagram node types (C = STA, WST, OIS, AIS, YIS, OTHER). Given a collection of m labeled examples $S = \{(x_1, c_1), ..., (x_m, c_m)\}$, the learning task is achieved by using a Bayesian classifier:

$$\hat{c} = \arg\max_{c \in C} p(y = c|x)$$

In this study, similar to most other tasks such as dialog act tagging or decision detection, with the recent advances in machine learning, we rely on a discriminative state of the art classification approach, namely Boosting. Boosting is an iterative learning algorithm that aims to combine weak base classifiers to come up with a strong classifier. At each iteration, a weak classifier is learned so as to minimize the training error, and a different distribution or weighting over the training examples is used to give more emphasis to examples that are often misclassified by preceding weak classifiers. For many text and speech categorization tasks Boosting is shown to results in performance very close to the state-of-the-art, while providing models that are easily interpretable by humans.

As the features for classification we rely on lexical and prosodic information. The lexical information is nothing but the word n-grams ($n = 1, 2, 3$) as extracted from candidate utterances. The prosodic features are expected to be useful to distinguish *backchannels* (subset of OTHER) from the rest and *issues* (OIS, AIS, and YIS), which are mainly questions, from *statements* (STA and WST). Inspired by the previous work on question detection [18], we compute several local and contextual prosodic features. The local features include mean, median, minimum, maximum pitch and energy features over the complete utterance, as well as their speaker normalized versions and range features. The contextual features compare the complete utterances, as well as the final 200ms window of the utterances to the following and preceding utterances.

While the nature of the task suggests that a sequence classification method may be more appropriate, as the data set we use in our experiments does not include annotations for all user utterances, we choose to use a local classifier without considering preceding and/or following utterances, except when computing features.

Extending this basic classification schema, we also propose the use of a cascaded approach, in which first a classifier is employed to distinguish the OTHER category (which are mostly backchannels) from the rest of the utterances. Then a second classifier is used only to determine the type of the issue or the statement. This has at least two big advantages. The first use is in discriminating backchannels from agreement statements. Note that most agreement sentences are single word utterances, such as *yeah* or *okay* and they are also used as backchannels. Furthermore their usage in a backchannelling utterance is typically more frequent than their usage for agreement. Unless contextual information is considered, there is no way to understand that a particular word is uttered for agreement or backchannelling. In the earlier work [12], the prosodic features were also shown to be useful in distinguishing backchannels from issues and statements. The second use is in making the task of secondary classifier easier by providing training data with no ambiguously labeled utterances.

4. EXPERIMENTS AND RESULTS

4.1. Data Sets and Evaluation

In our experiments on detecting argument diagram node types, we used the AMI Meeting Corpus [2], a multi-modal meeting data collection from meetings of 4 participants, that are about 30 minutes on

Category	Rel. Frequency	Avg.Utt. Length
STA	58.9%	16.0
WST	2.8%	16.9
OIS	3.4%	18.3
AIS	1.0%	23.8
YIS	6.4%	16.3
OTHER	27.5%	6.2

Table 2. The relative frequency and average utterance length in terms of number of words for statement (STA), weak statement (WST), open issue (OIS), A/B issue (AIS), Yes/No issue (YIS), and OTHER node types.

average. The meetings are scenario-driven, where participants have been assigned roles in a loosely scripted collaborative design task. All meetings are hand-transcribed and annotated fully or partially for several categories, such as dialog acts, topic segments. We used a 95-meeting subset which is also annotated for arguments between participants [7]. In this data set, not all utterances are annotated, therefore we only use the annotated utterances for training and testing. In total, there are 6,920 utterances annotated with a node category of one of the 6 node types. The relative frequency and average utterance length for each category are summarized in Table 2.

We performed n-fold cross validation experiments (n = 95), where a single meeting is used as the test set, and 10 meetings are used as the held-out data set to optimize parameters in each iteration. To compare results from different experiments, we use test set error rate (TER). In all the experiments, we used the BoosTexter classifier [19], an implementation of the boosting family of classifiers. All features, except word sequences and utterance begin and end times, are automatically computed. For computing word n-grams, we use manual transcriptions of user turns, to investigate the use of words for this task. Note that, unlike the previous work investigating the computability of TAS node labels [10], we excluded the punctuation marks, such as question marks that can signal questions, from the transcriptions.

4.2. Results

In order to form a simple baseline, we assigned all utterances the type of the majority class, that is a statement. This resulted in an error rate of 41.1%. The average utterance lengths for each node type seemed to be a good way of distinguishing backchannels that form the majority of the OTHER class from the rest of the utterances. Therefore, the first experiment we performed used only utterance length as classification feature. However this resulted in an error rate even higher than the simple baseline. Although the statements have a large average utterance length, their variance is also high. A quarter of all statements are single word utterances, such as confirmations in response to questions. Similarly, while most of the OTHER utterances are short backchannelling utterances, these also include non-backchannelling utterances that are not related to discussions. For example, 22% of the utterances of OTHER type are longer than 6 words, and 9% of them are longer than 15 words.

In order to form another baseline, we used only the word n-grams ($n = 1, 2, 3$) extracted from every utterance as classification features. This resulted in an error rate of 29.2%, which is significantly better than the simple majority class baseline. This shows that word sequences by themselves are useful for detection of node types. The selected word n-grams by BoosTexter included word sequences such as "Mm-hmm" that signal backchannels, "maybe", and "possibly" that signal weak statements. We also used all features in a kitchen sink model, and obtained the best single classifier performance of 27.2%, which is about 7% better relatively than the word n-grams only model. Table 3 summarizes the results from these experiments.

Experiment	Error rate
Majority	41.1%
Utterance Length	41.5%
Word n-grams	29.2%
All features	27.2%

Table 3. Results of node type detection experiments using single-pass classification.

Experiment	Error rate
Majority	27.6%
Word n-grams	17.6%
All features	15.1%

Table 4. Results of experiments that detect OTHER type utterances.

When examining the confusion matrices from these experiments, we confirmed that the prosodic features seem to help the detection of backchannelling utterances that form the majority of the OTHER category. To further analyze this, we also performed cascaded experiments, where in the first stage of the cascade, we tried to detect of an utterance belongs to the OTHER category or not, and in the second stage, classified only utterances that do not belong to the OTHER class into the remaining 5 categories. The results of the first stage of the cascaded experiments are shown in Table 4. The assignment of the majority class (non-OTHER) to all utterances results in an error rate of 27.6%, which drops to 17.6% when word n-grams are used, and 15.1% when all features are used. In the boosting model files, the most widely used prosodic features for detecting OTHER class utterances are speaker normalized mean and median energy, as well as pitch and energy ranges.

The results of the cascaded approach are shown in Table 5. If only the lexical features are used in this approach, the error rate is 28.8%, which is only slightly better than the single step approach using lexical features. The cascaded approach using all features results in an error rate of 26.6%, which is the best result on the overall, and is 9% relatively better than a single step classifier using only lexical features. If all features are used in the first step, and only lexical features are used in the second step, the classification performance is very similar, suggesting that the prosodic features are not as useful in the second step as expected. This may be due to the fact that the nodes frequently include more than one dialog act unit, and the question is not always the final dialog act unit of the issue nodes. The investigation of prosodic features computed at the dialog act unit level can help this problem, and remains as an issue to be further investigated.

Experiment	Error rate
Word n-grams	28.8%
All features	26.6%

Table 5. Results of cascaded node type detection experiments.

A difficulty related to this task is that not all utterances involved in meeting discussions are annotated, this made the application of sequence classification methods not practical.

5. CONCLUSIONS

We have presented a first study towards automatic argument diagramming of multiparty meetings. While this is a new task, it is an important enabling task for many higher level meeting understanding tasks, such as decision detection or summarization.

We have employed a cascaded approach relying on two classifiers using lexical and prosodic features for tagging the argumentation types of the utterances. We see that prosodic information is very helpful in distinguishing the backchannels and questions raising issues.

Our future work involves automatic tagging of relationships between the utterances. This is a complementary task mutually dependent on the task of utterance argumentation type classification which is studied in this paper. We also plan to perform experiments using the automatic speech recognizer output instead of manual transcriptions.

Acknowledgments: This work is partly supported by the Defense Advanced Research Projects Agency (DARPA) under Contract number NBCHD030010 (CALO). 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. Thanks to Rutger Rienks *et al.* for creating these data sets, and helping with the format of the data sets. Thanks to Elizabeth Shriberg and Gokhan Tur for their suggestions about the paper text.

6. REFERENCES

[1] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI Meeting Corpus," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong, 2003.

[2] J.C. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraiij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, M. McCowan, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus: A preannouncement," in *Proc. of the 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI)*, Edinburgh, Scotland, July 2005.

[3] A. Stolcke, X. Anguera, K. Boakye, O. Cetin, F. Grezl, A. Janin, A. Mandal, B. Peskin, C. Wooters, and J. Zheng, "Further progress in meeting recognition: The icsi-sri spring 2005 speech-to-text evaluation system," in *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI*, Edinburgh, U.K., July 2005.

[4] M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke, "Toward joint segmentation and classification of dialog acts in multiparty meetings," in *Proceedings of the Machine Learning for Multimodal Interaction: Second International Workshop, MLMI*, Edinburgh, U.K., July 2005.

[5] M. Zimmermann, D. Hakkani-Tür, E. Shriberg, and A. Stolcke, "Text based dialog act classification for multiparty meetings," in *Proceedings of the MLMI*, Washington D.C., May 2006.

[6] K. Riedhammer, D. Gillick, B. Favre, and D. Hakkani-Tür, "Packing the meeting summarization knapsack," in *Proceedings of INTERSPEECH*, Brisbane, Australia, September 2008.

[7] R. Rienks, D. Heylen, and E. van der Weijden, "Argument diagramming of meeting conversations," in *Multimodal Multiparty Meeting Processing, Workshop at the 7th International Conference on Multimodal Interfaces (ICMI)*, Trento, Italy, October 2005.

[8] K. Fujita, K. Nishimoto, Y. Sumi, S. Kunifuji, and K. Mase, "Meeting support by visualizing discussion structure and semantics," in *Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems*, Adelaide, Australia, April 1998.

[9] M. S. Bachler, S. J. Buckingham Shum, D. C. De Roure, D. T. Michaelides, and K. R. Page, "Meeting support by visualizing discussion structure and semantics," in *Proceedings of the first International Workshop on Hypermedia and the Semantic Web (HTSW2003)*, Nottingham, UK, August 2003.

[10] R. Rienks and D. Verbree, "About the usefulness and learnability of argument-diagrams from real discussions," in *3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Washington D.C., USA, May 2006.

[11] M. Purver, J. Dowding, J. Niekrasz, P. Ehlen, S. Noorbaloochi, and S. Peters, "Detecting and summarizing action items in multi-party dialogue," in *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, September 2007.

[12] E. Shriberg F. Yang, G. Tur, "Exploiting dialog act tagging and prosodic information for action item identification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, NV, USA, April 2008.

[13] R. Fernandez, M. Frampton, P. Ehlen, M. Purver, and S. Peters, "Modelling and detecting decisions in multi-party dialogue," in *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, Columbus, OH, USA, June 2008.

[14] V. Pallotta, J. Niekrasz, and M. Purver, "Collaborative and argumentative models of meeting discussions," in *Proceedings of the 5th Workshop on Computational Models of Natural Argument (CMNA)*, Edinburgh, Scotland, July 2005.

[15] Michel Galley, K. McKeown, J. Hirschberg, and E. Shriberg, "Identifying agreement 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)*, Barcelona, Spain, July 2004.

[16] G. Murray, S. Renals, and M. Taboada, "Prosodic correlates of rhetorical relations," in *Proceedings of HLT-NAACL Workshop on Analyzing Conversations in Text and Speech (ACTS)*, New York City, NY, USA, June 2006.

[17] R. Rienks and D. Verbree, "Twente argument schema annotation manual v 0.99b," http://mmm.idiap.ch/private/ami/annotation/TAS-annotation-manual.pdf.

[18] J. Liscombe, J.J. Venditti, and J. Hirschberg, "Detecting question-bearing turns in spoken tutorial dialogues," in *Proceedings of Ninth International Conference on Spoken Language Processing (ICSLP)*, Pittsburgh, PA, USA, September 2006.

[19] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.