EXTRACTING QUESTION/ANSWER PAIRS IN MULTI-PARTY MEETINGS

Andreas Kathol   *Gokhan Tur*

SRI International,
Speech Technology and Research Lab
Menlo Park, CA, 94025
{kathol,gokhan}@speech.sri.com

ABSTRACT

Understanding multi-party meetings involves tasks such as dialog act segmentation and tagging, action item extraction, and summarization. In this paper we introduce a new task for multi-party meetings: extracting question/answer pairs. This is a practical application for further processing such as summarization. We propose a method based on discriminative classification of individual sentences as questions and answers via lexical, speaker, and dialog act tag information, followed by a contextual optimization via Markov models. Our results indicate that it is possible to outperform a nontrivial baseline using dialog act tag information. More specifically, our method achieves a 13% relative improvement over the baseline for the task of detecting answers in meetings.

***Index Terms*—** multi-party meetings, question/answer pair extraction

1. INTRODUCTION

With the advances in human/human automatic speech recognition (ASR) and understanding, it becomes feasible to automatically handle multi-party meetings. This process typically includes recognition, retrieval, and extraction of certain information such as action items, topics, and so on. Large projects such as DARPA-funded CALO (Cognitive Agent that Learns and Organizes) [1], EU-funded AMI (Augmented Multi-party Interaction) [2], and CHIL (Computers in the Human Interaction Loop) [3] have provided important research infrastructure and benchmark evaluations for multi-party meeting processing.

Figure 1 presents an example meeting where the dialog act boundaries are marked by // tokens and dialog act tags (DATs) are shown in parentheses. This is an agenda-driven meeting, where each agenda item can be considered a separate topic, and may contain action items with due dates and assignees. The meeting contains only one agenda item, that is, *arrangements for Joe Browning*. Two action items, one about his office, another about his account, are owned by *Kathy Brown* and *Cindy Green*, respectively.

For meeting recognition research, the annual NIST meeting recognition evaluations have become a driving force with substantial performance improvements in recent years [4]. For meeting understanding, basic tasks such as dialog act segmentation and tagging [5, 6] as well as higher level tasks such as topic segmentation and detection [7, 8], action item extraction [9, 10], summarization [11, 12], and decision detection [13] have been studied in the literature.

This paper introduces a new task for meeting understanding: extracting question/answer pairs. In this task, the goal is to detect the questions and their corresponding answers. Question detection has been studied in the framework of dialog act tagging [14]. Similarly, there has been recent work on addressee detection or extracting adjacency pairs [15, 16]. In these tasks the goal is to detect who is talking with whom. In that sense the task on which we focus in this study addresses a special case of adjacency pair extraction, namely, detecting questions and their answers if there are any.

- *John Smith*: so we need to arrange an office for joe browning (statement)
- *Kathy Brown*: are there special requirements (question)
- *Cindy Green*: when is he co- (disruption)
- *John Smith*: yes (affirmation) // there are (statement)
- *John Smith*: we want him to be close to the image processing guys
- *Kathy Brown*: okay (agreement) // I'll talk to the secretary (commitment)
- *Cindy Green*: hold on (floor grabber) // wh- when is he coming (question)
- *John Smith*: next monday (statement) // he will start on sixteenth (statement)
- *Cindy Green*: okay (backchannel)
- *John Smith*: can you make sure he has an account by then (question)
- *Cindy Green*: sure (agreement) // no problem (statement)
- *John Smith*: let's see what is next in the agenda (suggestion)

Fig. 1. Excerpt from a meeting with dialog act tags.

Extracted question/answer pairs may be used in a number of practical ways. For example, they can be exploited for a better offline meeting browser. Alternatively, they can be exploited in a more sophisticated application such as meeting summarization. For example, if the question is included in the summary, it must also include the answer. Similarly, it can be used to detect action items as in the last question/answer pair in the example meeting. Furthermore, sometimes the full meaning is understood only when the question is combined with the answer. The second question/answer pair is an example of this, indicating that *Joe Browning will come next Monday*. It can also help the task of addressee detection.

Note that this is not a very clearly defined task. In some cases it is hard even for humans to distinguish whether an utterance is a backchannel or an answer, especially for single-word sentences such as *yeah* or *okay*. Furthermore some answers may span multiple sentences and sometimes it is not clear what the final sentence in the answer is. This is especially true for open-ended questions such as *what did you do last week?*, for which the answer might be a long

1-4244-1484-9/08/$25.00 ©2008 IEEE

report.

We treat this type of task in two steps. In the first step, the sentences are given scores for being questions and answers. This is framed as a classification problem and discriminative classification methods are used. For classification, we used lexical features (word ngrams) and dialog act tags of the sentences provided by a tagger trained using a different corpus. Then in the second step, the task is treated as a sequence classification problem and the question/answer pairs are aligned using a Markov model with the posterior probabilities obtained from the first step.

In the next section we present this two step approach in detail. Section 3 presents the experiments and results obtained using the CALO corpus. We conclude with a discussion and plans for future research.

2. APPROACH

We first describe a nontrivial baseline system attacking the problem of question/answer pair extraction. Then we describe a two step approach for question/answer extraction.

2.1. Baseline System

In this study we first built a baseline system relying on the dialog act tags of the sentences.[1] A dialog act is an approximate representation of the illocutionary force of an utterance, such as question or backchannel [17]. Dialog act tagging is generally framed as an utterance classification problem [17, 18].

In this study we trained a discriminative classification model for detecting 5 high level dialog act tags, namely, *question*, *statement*, *disruption*, *backchannel*, and *floor grabber/holder*, using only lexical features, i.e., word ngrams without any contextual features [19]. This classifier is trained using the ICSI MRDA corpus [14].

The baseline system assumes that a sentence is a question if it is tagged as a question with a posterior probability greater than some threshold (as optimized on a small held-out set). Then the next sentence (which is not a backchannel or floor grabber) uttered by a different speaker is assumed to be the answer.

2.2. Classification-Based System

Note that this is a nontrivial system which covers most cases and is fairly accurate as shown in the next section. However in certain cases it fails:

- Rhetorical questions, i.e., questions that do not require any answer.
- Questions with answers consisting of multiple sentences. For the example above, all the questions have answers consisting of two sentences. Furthermore, in some cases it is even hard for humans to detect when the answer ends.
- Questions followed by backchannels or overlap speech. For the example above, the first question is followed by a disrupted sentence.
- Disrupted questions that are not answered typically but marked as questions. For the example above, the first disruption demonstrates this.
- Questions followed by a clarification question; hence, the answer may answer both questions.
- More than one person may answer the same question.

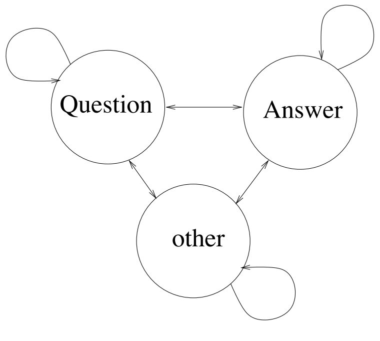

Fig. 2. Conceptual scheme of the question/answer pair extraction system

In light of these observations, we considered modeling question/answer sequences using contextual information. Our algorithm consists of two steps.

In the first step, we build a three-way statistical classification model to detect questions, answers, or other using the annotated data. The features we use include word ngrams and the dialog act tags (obtained from the tagger described above) for the current, two previous, and two subsequent sentences. Other features encode whether the given utterance has the same speaker as the previous utterance. Furthermore, we use a variety of acoustic features derived from the final word in the utterance. We also use a contextual feature encoding the distance to the most recent question as given by the dialog act tagger. For model building, we build a three-way classifier for detecting questions, answers, or utterances that are neither questions nor answers. We have also explored using two binary classifiers, for questions and answers, respectively. So far it has not shown to be superior to the three-way classifier approach pursued here.

While the step described above provides a system for extracting question/answer pairs, a significant weakness is that it considers little contextual information. A sentence may be selected as an answer with no prior question. To alleviate issues of this kind, we frame this problem as a sequence classification task and convert the question/answer detection as the task of finding the most probable tag sequence, $T \in \{question, answer, other\}$, given the sentences, W, using maximum a posteriori decoding via the Viterbi algorithm:

$$argmax_T P(T|W) \approx argmax_T P(W|T)^{\lambda} \times P(T)^{(1-\lambda)}$$

where λ is used to compensate for the dynamic ranges of both probability distributions and is optimized on a held-out set as typically done in speech processing. To this end, we trained a Markov model consisting of three states, *question*, *answer*, and *other* as shown in Figure 2. The state observations are simply the sentences. The state transition probabilities, $P(T)$, are obtained from the language model trained using the labeled data. The state observation likelihoods, $P(W|T)$, are obtained by converting the posterior probabilities obtained in the previous step into likelihoods by dividing into their priors using Bayes rule:

$$P(W|T) = \frac{P(T|W)P(W)}{P(T)}$$

Note that $P(W)$ is irrelevant since it is constant for all possible choices of T and this is taken care of using the weight λ during the Viterbi search.

[1] We assume that the utterances are already segmented into sentences.

	F-measure	Recall	Precision
M-question	0.969	0.940	1.000
M-answer	0.799	0.733	0.881
M-QA pair	0.913	0.884	0.857

Table 1. Performance figures for the baseline approach using manual dialog act tags.

	F-measure	Recall	Precision
M-question	0.681	0.534	0.939
M-answer	0.541	0.397	0.852
M-QA pair	0.609	0.457	0.913

Table 2. Performance figures for the baseline approach using hypothesized dialog act tags.

	F-measure	Recall	Precision
M-question	0.965	0.948	0.982
M-answer	0.757	0.698	0.827
M-QA pair	0.874	0.870	0.879

Table 3. Performance figures after the first step of the proposed approach using manual dialog act tags.

	F-measure	Recall	Precision
M-question	0.700	0.603	0.833
M-answer	0.537	0.405	0.797
M-QA Pair	0.592	0.457	0.840

Table 4. Performance figures after the first step of the proposed approach using hypothesized dialog act tags.

3. EXPERIMENTS AND RESULTS

Below we presents the experiments we performed using the CALO meeting corpus. First we show the results using the nontrivial baseline, then the results after the first step, and finally the results obtained using the proposed approach. We report results using both manually and automatically annotated dialog act tags.

3.1. Data

Our data set consists of 14 meetings collected in 2005 as part of the CALO project. These comprise a total of 1435 utterances. The entire corpus has been annotated in terms of high-level dialog act tags, that is, statement, question, disruption, floor grabber/holder, and backchannel. The only two DATs of interest for our purposes are questions (“q”) and statements (“s”).

The corpus has also been hand-annotated to indicate the status of an utterance within a meeting as a question or as an answer to some question. To avoid confusion with dialog act tags, we will refer to these annotations as “meeting questions” (“m-questions”) and “meeting answers” (“m-answers”). There are a total of 116 m-questions and 116 m-answers in the corpus. These constitute 92 clearly identifiable m-question/m-answer pairs (“m-QA” pairs). The discrepancy in numbers is due, in part, to the reasons listed in the previous section. Furthermore, in a number of cases, an utterance was categorized both as an m-answer to a previous m-question, and as an m-question its own right.

> Speaker 1: are there any other questions? (Q1)
> Speaker 2: could you tell me where your office is located? (A1, Q2)

3.2. Baseline Results

We established a baseline performance by means of a simple heuristic. Every utterance DAT-marked “q” was considered to be an m-question, and the next non-q utterance by a different speaker was counted as its corresponding m-answer. The resulting performance using manually labeled DATs for m-questions, individual m-answers, and m-QA pairs is given in Table 1.[2]

Since manually annotated DATs may not be available, a more realistic condition is to assume that the DAT information at hand has been produced automatically by a classifier trained using the ICSI MRDA corpus [14]. The error rate of the dialog act tagger is found to be 9.6%. As is shown in Table 2, the baseline performance is significantly worse than that for manually annotated DATs. This is to be expected given the amount of noise introduced by automatic DAT labeling.

3.3. Question/Answer Detection via Discriminative Classification

We subsequently trained an adaptive Boosting classifier [20] on the meeting data, with the goal of predicting m-question/m-answer status on the basis of a number of features. Because of the small amount of data available, a 14-fold cross-validation scheme was used in training, optimizing and testing. Each fold consists of 11 meetings for training and two for optimization. The final meeting is held out for testing so that each meeting is used as a test set in some fold.

The resulting performance in terms of precision, recall and f-measure is given in Table 3. The classifier performs slightly below baseline for m-question and m-answer detection, while performing considerably worse in terms of m-QA-pair detection. This shows the importance of context information. Although the distance to the previous question is used as a feature, apparently this is not enough.

When we use automatically tagged examples, a slightly different picture emerges as shown in Table 4. M-answer detection via classifier still performs slightly worse than in the baseline. However, m-question detection now is slightly better than the baseline. m-QA-pair detection performance is also slightly worse.

3.4. Question/Answer Detection via Contextual Optimization

The final set of experiments implements the idea of incorporating contextual information by means of the state transitions between m-questions and m-answers, as described in Section 2.2. In this step we used the SRILM toolkit [21] for both modeling the trigram tag sequence, $P(T)$, and performing Viterbi search.

First, in Table 5, we list the results for corrected DATs. Compared to the simple classifier in Table 3, we see a modest decline in m-question detection and some improvement in m-answer and m-QA pair detection. Moreover, this system performs worse than the baseline (Table 1) on all three scores.

In contrast, the use of contextual optimization *does* lead to a notable improvement over the baseline in the case of hypothesized DATs, as shown in Table 6. Specifically, we obtain a 13% relative improvement in m-answer detection and a 2% relative gain in m-question detection. In addition, this system surpasses the baseline in detecting m-QA pairs by 4% relative.[3]

[2] One of the reasons why the set of m-QA pairs is not identical to the union of m-questions and m-answers is that not all m-questions are necessarily answered in a given dialog.

[3] We defer to further study the question whether the lack of improvement

	F-measure	Recall	Precision
M-question	0.952	0.940	0.965
M-answer	0.759	0.750	0.770
M-QA Pair	0.884	0.913	0.857

Table 5. Performance figures using the proposed approach with manual dialog act tags.

	F-measure	Recall	Precision
M-question	0.716	0.672	0.765
M-answer	0.607	0.560	0.663
M-QA pair	0.633	0.620	0.648

Table 6. Performance figures using the proposed approach with hypothesized dialog act tags.

4. CONCLUSIONS

We have presented a new meeting understanding task, namely, extracting question/answer pairs in multi-party meetings. We propose a two-step classification-based approach combining discriminative and generative classification methods for an effective sequence classification. Our results indicate that while the proposed approach outperforms a nontrivial baseline using automatically labeled DATs.

Our future work includes using additional features for this task, such as the annotated adjacency pairs in the ICSI MRDA corpus and exploiting the addressee detection methods to determine the speaker who responds to the question.

5. ACKNOWLEDGMENTS

We thank Elizabeth Shriberg for many helpful discussions. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) CALO program under Contract No. NBCHD030010. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or the Department of Interior-National Business Center (DOI-NBC).

6. REFERENCES

[1] “SRI Cognitive Assistant that Learns and Organizes (CALO) Project,” http://www.ai.sri.com/project/CALO.

[2] “Augmented multi-party interaction,” http://www.amiproject.org.

[3] “Computers in the human interaction loop,” http://chil.server.de.

[4] Jonathan G. Fiscus, Nicolas Radde, John S. Garofolo, Audrey Le, Jerome Ajot, and Christophe Laprun, “The Rich Transcription 2005 Spring meeting recognition evaluation,” in *MLMI, Revised Selected Papers*, Steve Renals and Samy Bengio, Eds. 2006, vol. 3869 of *Lecture Notes in Computer Science*, pp. 369–389, Springer.

[5] J. Ang, Y. Liu, and E. Shriberg, “Automatic dialog act segmentation and classification in multiparty meetings,” in *Proceedings of the ICASSP*, Philadelphia, PA, March 2005.

[6] M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke, “Toward joint segmentation and classification of dialog acts in multiparty meetings,” in *Proceedings of the MLMI*, Edinburgh, U.K., July 2005.

[7] Matthew Purver, Konrad Körding, Thomas Griffiths, and Joshua Tenenbaum, “Unsupervised topic modelling for multi-party spoken discourse,” in *Proceedings of the COLING-ACL*, Sydney, Australia, July 2006, pp. 17–24, Association for Computational Linguistics.

[8] Satanjeev Banerjee and Alexander Rudnicky, “A TextTiling based approach to topic boundary detection in meetings,” in *Proceedings of the ICSLP*, Pittsburgh, Pennsylvania, Sept. 2006.

[9] Matthew Purver, Patrick Ehlen, and John Niekrasz, “Detecting action items in multi-party meetings: Annotation and initial experiments,” in *MLMI, Revised Selected Papers*. 2006.

[10] Paul N. Bennett and Jaime G. Carbonell, “Combining probability-based rankers for action-item detection,” in *Proceedings of the HLT/NAACL*, Rochester, NY, Apr. 2007, pp. 324–331, Association for Computational Linguistics.

[11] A. Waibel, M. Bett, M. Finke, and R. Stiefelhagen, “Meeting browser: Tracking and summarizing meetings,” in *Proceedings of the DARPA Broadcast News Tracsription and Understanding Workshop*, Lansdowne, VA, June 1998.

[12] J Carletta G Murray, S Renals, “Extractive summarization of meeting recordings,” in *Proceedings of the Interspeech*, Lisbon, Portugal, September 2005.

[13] Pei-Yun Hsueh and Johanna Moore, “What decisions have you made?: Automatic decision detection in meeting conversations,” in *Proceedings of NAACL/HLT*, Rochester, New York, 2007.

[14] E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey, “The ICSI Meeting Recorder Dialog Act (MRDA) Corpus,” in *Proceedings of the SigDial Workshop*, Boston, MA, May 2004.

[15] Rainer Stiefelhagen, Jie Yang, and Alex Waibel, “Modeling focus of attention for meeting indexing based on multiple cues,” *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 928–938, 2002.

[16] Surabhi Gupta, John Niekrasz, Matthew Purver, and Daniel Jurafsky, “Resolving “you” in multi-party dialog,” in *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, 2007.

[17] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. van Ess-Dykema, and M. Meteer, “Dialogue act modeling for automatic tagging and recognition of conversational speech,” *Computational Linguistics*, vol. 26, no. 3, pp. 339–373, 2000.

[18] 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.

[19] G. Tur, U. Guz, and D. Hakkani-Tür, “Model adaptation for dialog act tagging,” in *Proceedings of the IEEE/ACL SLT Workshop*, 2006.

[20] 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.

[21] A. Stolcke, “SRILM – An Extensible Language Modeling Toolkit,” in *Proceedings of the ICSLP*, Denver, CO, September 2002.

for corrected vs. hypothesized DATs is inherent to our approach or an incidental property of the relatively small corpus.