Molweni: A Challenge Multiparty Dialogue-based Machine Reading Comprehension Dataset with Discourse Structure

Jiaqi Li[1], Ming Liu[1,3], Min-Yen Kan[2], Zihao Zheng[1], Zekun Wang[1]
Wenqiang Lei[2], Ting Liu[1,3], Bing Qin[1,3]*
1. Harbin Institute of Technology, Harbin, China.
2. National University of Singapore, Singapore.
3. Peng Cheng Laboratory, Shenzhen, China.
{jqli, mliu, zhzheng,zkwang,tliu,qinb}@ir.hit.edu.cn
kanmy@comp.nus.edu.sg, wenqianglei@gmail.com

Abstract

Research into the area of multiparty dialog has grown considerably over recent years. We present the *Molweni* dataset[1], a machine reading comprehension (MRC) dataset with discourse structure built over multiparty dialog. Molweni's source samples from the Ubuntu Chat Corpus, including 10,000 dialogs comprising 88,303 utterances. We annotate 30,066 questions on this corpus, including both answerable and unanswerable questions. Molweni also uniquely contributes discourse dependency annotations in a modified Segmented Discourse Representation Theory (SDRT; (Asher et al., 2016)) style for all of its multiparty dialogs, contributing large-scale (78,245 annotated discourse relations) data to bear on the task of multiparty dialog discourse parsing. Our experiments show that Molweni is a challenging dataset for current MRC models: `BERT-wwm`, a current, strong SQuAD 2.0 performer, achieves only 67.7% F_1 on Molweni's questions, a 20+% significant drop as compared against its SQuAD 2.0 performance.

1 Introduction

Research into multiparty dialog has recently grown considerably, partially due to the growing ubiquity of dialog agents. Multiparty dialog applications such as discourse parsing and meeting summarization are now mainstream research (Shi and Huang, 2019; Hu et al., 2019; Li et al., 2019; Zhao et al., 2019; Sun et al., 2019; Perret et al., 2016; Afantenos et al., 2015). Such applications must consider the more complex, graphical nature of discourse structure: coherence between adjacent utterances is not a given, unlike standard prose where sequential guarantees hold.

In a separate vein, the area of machine reading comprehension (MRC) research has also made unbridled progress recently. Most existing datasets for machine reading comprehension (MRC) adopt well-written prose passages and historical questions as inputs (Richardson et al., 2013; Rajpurkar et al., 2016; Lai et al., 2017; Choi et al., 2018; Reddy et al., 2019).

Reading comprehension for dialog — as the intersection of these two areas — has naturally begun to attract interest. Ma et al. (2018) constructed a small dataset for passage completion on multiparty dialog, but which has been easily dispatched by CNN+LSTM models using attention. The DREAM corpus (Sun et al., 2019) is an MRC dataset for dialog, but only features a minute fraction (1%) of multiparty dialog. FriendsQA is a small-scale span-based MRC dataset for multiparty dialog, which derives from TV show *Friends*, including 1,222 dialogs and 10,610 questions (Yang and Choi, 2019). The limited number of dialogs in FriendsQA makes it infeasible to train more complex model to represent multiparty dialogs due to overfitting, and the lack of annotated discourse structure prevents models from making full use of the characteristics of multiparty dialog.

Dialog-based MRC thus varies from other MRC variants in two key aspects:

C1. Utterances of multiparty dialog are much less locally coherent than in prose passages. A passage is a continuous text where there is a discourse relation between every two adjacent sentences. Therefore,

[*] Corresponding author.

[1] https://github.com/HIT-SCIR/Molweni

Dialogue 1

nbx909: how do i **find the address of a usb device** ? U_1
likwidoxigen: try taking it out to dinner and do a little wine and dine and it shoudl tell ya U_2
likwidoxigen : what sort of device ? U_3
babo: ca n't i just copy over the os and leave the data files untouched ? U_4
nbx909 : only if you do an upgrade U_5
nuked: should i just restart x after installing U_6
likwidoxigen: i 'd do a full restart so that **it re-loads the modules** U_7

(a)

Q1: Why does *likwidoxigen* do a full restart?
A1: it re-loads the modules

Q2: What does *nbx909* want to do?
A2: find the address of a usb device

Q3: How to restart network?
A3: *NA.*

(b)

Figure 1: (Dialog 1) A corpus example from Molweni. There are four speakers in the dialog: *nbx909*, *Likwidoxigen*, *babo*, and *nuked*. In total, the speakers make seven utterances: U_1 to U_7. Our annotators proposed three questions against the provided dialog: Q1–3, where Q1 and Q2 are answerable questions, and Q3 is unanswerable. Due to the properties of informal dialog, the instances in our corpus often have grammatical errors.

we can regard each paragraph in a passage as a linear discourse structure text. In contrast, there may be no discourse relation between adjacent utterances in a multiparty dialog. As such, we regard the discourse structure of a multiparty dialog as a dependency graph where each node is an utterance.

C2. Multiparty dialog subsumes the special case of two-party dialog. In most cases, the discourse structure of a two-party dialog is tree-like, where discourse relations mostly occur between adjacent utterances. However, in multiparty dialog, such assumptions hold less often as two utterances may participate in discourse relations, though they are very distant.

Prior MRC works do not consider the properties of multiparty dialog. To address this gap in understanding of multiparty dialog, we created Molweni. In Dialog 1 (*cf.* Fig 1), four speakers converse over seven utterances. We additionally employ annotators to read the passage and contribute questions: in the example, the annotators propose three questions: two answerable and one unanswerable. We observe that adjacent utterance pairs can be incoherent, illustrating the key challenge. It is non-trivial to detect discourse relations between non-adjacent utterances; and crucially, difficult to correctly interpret a multiparty dialog without a proper understanding of the input's complex structure.

We derived Molweni from the large-scale multiparty dialog Ubuntu Chat Corpus (Lowe et al., 2015). We chose the name *Molweni*, as it is the plural form of "Hello" in the Xhosa language, representing multiparty dialog in the same language as *Ubuntu*. Our dataset contains 10,000 dialogs with 88,303 utterances and 30,066 questions including answerable and unanswerable questions. All answerable questions are extractive questions whose answer is a span in the source dialog. For unanswerable questions, we annotate their plausible answers from dialog. Most questions in Molweni are 5W1H questions – *Why*, *What*, *Who*, *Where*, *When*, and *How*. For each dialog in the corpus, annotators propose three questions and find the answer span (if answerable) in the input dialog.

To assess the difficulty of Molweni as an MRC corpus, we train BERT's whole word masking model on Molweni, achieving a 54.7% exact match (EM) and 67.7% F_1 scores. Both scores show larger than 10% gap with human performance, validating its difficulty. Due to the complex structure of multiparty dialog, human performance just achieves 80.2% F_1 on Molweni. In particular, annotators agreed that knowledge of the correct discourse structure would be helpful for systems to achieve better MRC performance.

This comes to the second key contribution of Molweni. We further annotated all 78,245 discourse relations in all of Molweni's dialogs, in light of the potential help that annotated discourse structure might serve. Prior to Molweni, the STAC corpus is the only dataset for multiparty dialog discourse parsing (Asher et al., 2016). However, its limited scale (only 1K dialogs) disallow data-driven approaches to discourse parsing for multiparty dialog. We saw the additional opportunity to empower and drive this direction of research for multiparty dialog processing.

2 Related work

Discourse parsing for multiparty dialog. Prior to Molweni, STAC was the only corpus containing annotations for discourse parsing on multiparty chat dialogs (Asher et al., 2016). The corpus derives from the online version of the game *The Settlers of Catan*. The game is a multiparty, win–lose game. We introduce the senses of discourse relation in STAC in Section 3.2. The STAC corpus contains 1,091 dialogs with 10,677 utterances and 11,348 discourse relations. Compared with STAC, our Molweni dataset contains 10,000 dialogs comprising 88,303 utterances and 78,245 discourse relations.

Machine reading comprehension. There are several types of datasets for machine comprehension, including multiple-choice datasets (Richardson et al., 2013; Lai et al., 2017), answer sentence selection datasets (Wang et al., 2007; Yang et al., 2015) and extractive datasets (Rajpurkar et al., 2016; Joshi et al., 2017; Trischler et al., 2017; Rajpurkar et al., 2018) . To extend existing corpora, our Molweni dataset is constructed to be an extractive MRC dataset for multiparty dialog, which includes both answerable questions and unanswerable questions. Similar to Squad 2.0 (Rajpurkar et al., 2018), we also annotate plausible answers for unanswerable questions. Three closely related datasets are the extended crowdsourced *Friends* corpus (Ma et al., 2018), DREAM (Sun et al., 2019) and FriendsQA (Yang and Choi, 2019). Different from these three MRC datasets for dialog, Molweni contributes the discourse structure of dialogs, and additional instances of multiparty dialogs and unanswerable questions.

3 The Molweni corpus

Our dataset derives from the large scale multiparty dialogs dataset — the Ubuntu Chat Corpus (Lowe et al., 2015). We list our three reasons in choosing the Ubuntu Chat Corpus as the base corpus for annotation.

- First, the Ubuntu dataset is a large multiparty dataset. After filtering the dataset by only retaining all utterances with response relations, there are still over 380K sessions and 1.75M utterances. In each session, there are 3-10 utterances and 2-7 interlocutors.
- Second, it is easy to annotate the Ubuntu dataset. The Ubuntu dataset already contains Response-to relations that are discourse relations between different speakers' utterances. For annotating discourse dependencies in dialog, we only need to annotate relations between the same speaker's utterances and the specific sense of discourse relation. Because the length of dialogs in the Ubuntu dataset is not too long, we can easily summarize dialogs and propose some questions for the dialog.
- Third, there are many papers doing experiments on the Ubuntu dataset, and the dataset has been widely recognized. For example, Kummerfeld et al. (2019) proposed a large-scale, offshoot dataset for conversation disentanglement based on the Ubuntu IRC log. Also recently, Hu et al. (2019) also used the Ubuntu Chat Corpus as their dataset for learning dialog graph representation.

The discourse dependency structure of each multiparty dialog can be regarded as a discourse dependency graph where each node is an utterance. To learn better graph representation of multiparty dialogs, we filter the Ubuntu Chat Corpus for complex dialogs – those dialogs with 8–15 utterances and 2–9 speakers. As multiparty dialog is already intensely complex for the current state of the art, we chose to further simplify in our selection criteria, additionally filtering out dialogs with long utterances (more than 20 words). Finally, we randomly chose a subset of ∼10,000 dialogs with 88,303 utterances from the Ubuntu dataset. We give an overview of Molweni's key demographics in Table 1.

Table 1: Overview of Molweni for MRC.

	Train	**Dev**	**Test**	**Total**
Number of Dialogs	8,771	883	100	9,754
Number of Utterances	77,374	7,823	845	86,042
Number of Questions	24,682	2,513	2,871	30,066

10,000 dialogs are divided into two parts: 100 dialogs in common (public dialog) and 9,900 dialogs for different annotators (private dialog). Each annotator is asked to annotate 1,090 dialogs (990 private dialogs and 100 public dialogs) in two aspects: machine reading comprehension and discourse structure. All annotators chose to annotate the discourse structure of the dialog, and then propose questions and find answer spans for the dialog. All annotators agreed that it would be helpful to annotate the MRC task after annotating the discourse structure.

In total, our subjects annotated 9,754 dialogs, slightly fewer than 10,000 dialogs, consisting of 88,303 utterances, and contributed 30,066 questions for machine reading comprehension and 78,245 discourse relation annotations. There are 8,771 dialogs in the demarcated TRAIN set for both machine reading comprehension and discourse parsing tasks. 883 dialogs are used for DEV set. Each annotator is asked to propose three questions per dialog. There are 100 dialogs in common for all ten annotators, and these 100 dialogs comprise our TEST set. Each dialog in the training set and develop set has three questions. Our annotation team proposed a total of 2,871 questions for the 100 dialogs in TEST sets.

Detailed statistics are shown in Table 2. The average number of speakers per dialog is 3.51, which means that most dialogs are multiparty (as opposed to 2-party) dialogs. The number of two-party dialogs and multiparty dialogs in our dataset is 2,117 and 7,883, respectively. In Molweni, the average and maximum length of the selected dialogs are 8.82 and 14 utterances, respectively, and the number of answerable and unanswerable questions are 25,779 and 4,287, respectively.

Table 2: Detailed statistics for the *Molweni* corpus.

Metric	**Number**
Average / Maximum number of speakers per dialog	3.51 / 9
Average / Maximum question length (in tokens)	5.91 / 18
Average / Maximum answer length (in tokens)	4.08 / 19
Average / Maximum dialogue length (in tokens)	104.4 / 208
Average / Maximum dialogue length (in utterances)	8.82 / 14
Vocabulary size	24,615
Answerable questions	25,779
Unanswerable questions	4,287

3.1 Annotation for machine reading comprehension

We hired ten annotators to construct our Molweni dataset. As the Ubuntu corpus is technical in nature, all annotators are undergraduate students whose major is computer science to annotate the corpus. Annotators are non-native English speakers but who have an English proficiency certificate. They are all familiar with Linux operation system.

Annotators propose three questions for each dialog and annotate the span of answers in the input dialog. There are two types of questions in our corpus, namely, answerable questions and unanswerable questions:

1. **Answerable questions.** For these questions, the answer is a continuous span from source dialog. Annotators were asked to label answers from input dialog and ensure answers were succinct, without including extraneous text.

2. **Unanswerable questions.** To make the reading comprehension task more challenging, we annotate unanswerable questions and their plausible answers (PA). The plausible answers are quite related to unanswerable questions.

We compare Molweni against other datasets in Table 3. We see that existing dialog MRC datasets neither contribute either unanswerable questions, nor annotated discourse structure. Due to the complex structure of multiparty dialogs, we believe that it is essential to adopt the discourse dependency structure for the machine modeling towards multiparty dialog understanding. To the best of our knowledge to date, Molweni is the only MRC dataset that is annotated with discourse structure.

We give example questions from Molweni in Table 4. In particular, most of the questions in our dataset are questions lead by *Why*, *What*, *Who*, *Where*, *When*, and *How*. Only a small proportion of the questions are Other questions; questions lead by words such as *Do*, *Which*, and *Whose*. When annotators propose

Table 3: Comparison of Molweni with other MRC datasets on answer type, text type (dialogue or written text), multiparty dialogs or not, unanswerable questions, and discourse structure.

Dataset	**Answer type**	**Dialogue text**	**Multiparty dialogue**	**Unanswerable questions**	**Discourse structure**
RACE (Lai et al., 2017)	multiple-choice	✗	✗	✗	✗
NarrativeQA (Kocisky et al., 2018)	abstractive	✗	✗	✗	✗
CoQA (Choi et al., 2018)	abstractive	✗	✗	✔	✗
SQuAD 2.0 (Rajpurkar et al., 2018)	extractive	✗	✗	✔	✗
QuAC (Choi et al., 2018)	extractive	✗	✗	✔	✗
(Ma et al., 2018)	cloze	✔	✔	✗	✗
DREAM (Sun et al., 2019)	multiple-choice	✔	✔	✗	✗
FriendsQA (Yang and Choi, 2019)	extractive	✔	✔	✗	✗
Molweni (Our)	extractive	✔	✔	✔	✔

questions, they are asked to consider the characteristics of multiparty dialogs. For example, for *Why* and *How* questions, it is essential to know the question–answer pair and the cause–result in the dialog. For *How* questions, it is important to understand the role of speakers in order to properly represent the multiparty dialog. As such, *Why* and *How* often require a deeper understanding of the dialog.

Table 4: Examples of questions in Molweni.

Question	Example		Proportion(%)
How	How to do an upgrade?	How can I use this machine?	9.9
Why	Why is it not mounted?	Why does *jimcoonact* meet the error?	4.3
Who	Who is chart's service customers?	Who is using ubuntu?	4.7
When	When does *rhodry* have the error?	When is *SuperMiguel* back?	1.7
Where	Where did *earthen* write in?	Where is the device?	5.7
What	What does *elnomade* choose?	What does *noone* need?	71.7
Others	Does *elnomade* choose the print?	Which version does *xxiao* find?	1.9

3.2 Annotation for discourse structure of multiparty dialogs

The task of discourse parsing for multiparty dialogs is to determine the discourse relations among utterances. To enable better future modeling of such multiparty discourse, we represent a multiparty dialog by a directed acyclic graph (DAG). The process of annotating the discourse structure consists of two parts: predicting the links between utterances, and classifying the sense of the resultant discourse relation. Table 5 gives an overview of the statistics for Molweni's discourse parsing annotations.

An edge between two utterances represents the existence of a discourse dependency relation. The direction of the edge represents the direction of discourse dependency. In this subtask, what annotators need to do is to confirm whether two utterances have a discourse relation. Following the convention in the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), we term the two utterances as *Arg1* and *Arg2*, sequentially.

Discourse relations on two non-adjacent utterances are rare in prose but common in multiparty dialogs. When we annotate dialogs, annotators should sequentially read the dialog from its beginning to its final utterance. For each utterance, annotators need to find at least one parent node from among the previous utterances. We assume that the discourse structure is a connected graph and no utterance is isolated.

After we find the discourse relation between two utterances, we then need to confirm the specific relation sense. Although the sense hierarchy in PDTB has been broadly adopted (Lei et al., 2017; Lei et al., 2018), we adopt the modified Segmented Discourse Reprsentation Theory (SDRT) hierarchy, defined in STAC dataset (Asher et al., 2016), as it is designed specifically for multiparty dialog. There are 16 discourse relations in the STAC schema, as given in Table 6, where the top four most frequent relations (Comment, Clarification Question, Question Answer Pair (QAP), and Continuation) make up over 80% of the relations in the corpus.

Table 5: Statistics of Molweni's annotated discourse relations.

	Train	Dev	Test	Total
Number of Dialogs	9,000	500	500	10,000
Number of Utterances	79,487	4,386	4,430	88,303
Number of Relations	70,454	3,880	3,911	78,245

Table 6: Discourse relation types in Molweni and their meanings, listed in order of descending frequency in the corpus.

Relation	Meaning	Proportion(%)
Comment	Arg2 comments Arg1.	31.7
Clarification_question	Arg2 clarifies Arg1.	24.0
Question-answer_pair	Arg1 is a question and Arg2 is the answer of Arg1.	20.1
Continuation	Arg2 is the continuation of Arg1.	6.7
Acknowledgement	Arg2 acknowledges Arg1.	3.2
Q-Elab	Arg1 is a question and Arg2 tries to elaborate Arg1.	3.0
Result	Arg2 is the effect brought about by the situation described in Arg1.	2.6
Elaboration	Arg2 elaborates Arg1.	2.2
Explanation	Arg2 is the explanation of Arg1.	1.6
Correction	Arg2 corrects Arg1.	1.2
Contrast	Arg1 and Arg2 share a predicate or property and a difference on shared property.	1.2
Conditional	Arg1 is the condition of Arg2 or Arg2 is the condition of Arg1.	1.0
Background	Arg2 is the background of Arg1.	0.4
Narration	Arg2 is the narration of Arg1.	0.3
Alternation	Arg1 and Arg2 denote alternative situations.	0.2
Parallel	Arg2 and Arg1 are parallel and present almost the same meaning.	0.2

For the discourse parsing task, we used 500 dialogs for development and 500 dialogs for testing which is different from the MRC tasks. In opposition to the frequent relationships, there are also four types of relations that individually account for less than one percent of the corpus, namely, Alternation, Background, Narration, and Parallel. This is similar to the proportion of these four types of relations in the STAC dataset as well: only 0.5–2.0%. Next, according to the distribution of all kinds of relations, we need to consider merging some rare relation types in future work, so as to propose a more practical sense hierarchy for multiparty dialogs.

Multi-relational link prediction aims to predict missing links in an edge-labeled graph. This task focuses on the relations between entities (Bordes et al., 2013). However, discourse parsing focus on finding the discourse dependency arcs between different utterances.

The discourse dependency structures of Dialog 1 and Dialog 2 are shown in Fig. 3 where each utterance is represented as a node in the dependency graph. The label on the link in the discourse dependency relation. Dialog 1 (*cf.* Table 1) is a multiparty dialog with four speakers: *nbx909*,*likwidoxigen*, *babo*, *nuked* and seven utterances. Dialog 2 (*cf.* Fig. 2) has two speakers: *toma-* and *woodgrain*, and eight utterances in total. From Fig. 1, we can find that most of the discourse relations of two-party dialog occurs between adjacent utterances.

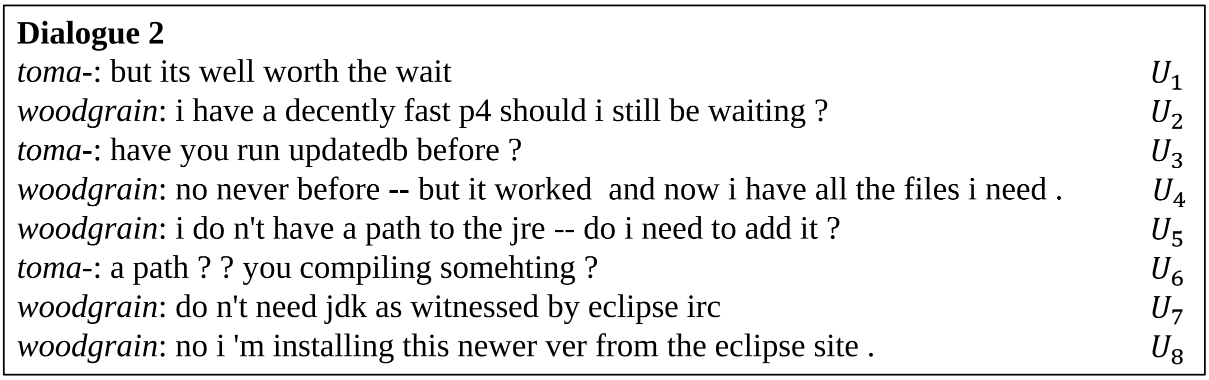
3.3 Data Quality

To ensure the quality of the corpus, we adopt two ways to check the annotation: a manual, human check as well as a programmatic check.

- **Manually Check**. Two authors of our Molweni dataset sample some instances to check the quality of proposed questions and feedback bad questions to the annotator.

- **Programmatic Check**. If answers cannot be found in the source dialog, the annotator would be asked to annotate the dialogs again until passing the check. We additionally check that the questions are grammatically correct using the *Grammarly* web application [2].

[2] https://app.grammarly.com/



Dialogue 2

toma-: but its well worth the wait U_1

woodgrain: i have a decently fast p4 should i still be waiting ? U_2

toma-: have you run updatedb before ? U_3

woodgrain: no never before -- but it worked and now i have all the files i need . U_4

woodgrain: i do n't have a path to the jre -- do i need to add it ? U_5

toma-: a path ? ? you compiling somehting ? U_6

woodgrain: do n't need jdk as witnessed by eclipse irc U_7

woodgrain: no i 'm installing this newer ver from the eclipse site . U_8

Figure 2: Dialog 2 is a two-party dialog example with eight utterances — U_1 to U_8 — proposed by two speakers: *toma-* and *woodgrain*.

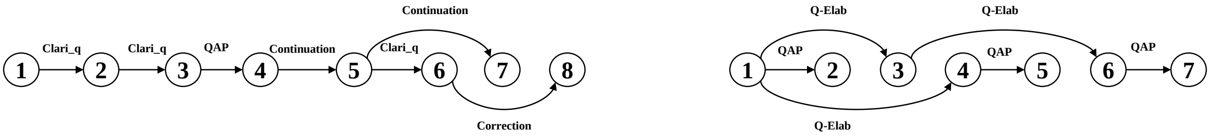


Figure 3: The discourse dependency structure and relations for Dialog 2 (Left, two-party) and Dialogue 1 (Right, multiparty). Clari_q, QAP, and Q-Elab are respectively short for Clarification_question, Question-answer_pair, and Question-Elaboration. The label on the link represents the discourse dependency relations between two utterances.

After four rounds of revision, we obtain the currently published version of the dataset.

We calculate the Fleiss Kappa value to check on the interannotator consistency. A Kappa value of 1.0 signifies complete agreement, and a value of 0.0 signifies completely uncorrelated judgments. Kappa values above The Kappa value of discourse dependency links is 0.91 which is an almost perfect agreement because the Ubuntu dataset initially contains the response-to relations, and annotators adopt most of the links. The final Kappa value of both links and relations is 0.56 among annotators, close to that of 0.58 obtained in the original STAC corpus. One reason for the drop of Kappa after labeling relation types is the discourse relation recognition is a multi-label task. There could be more one relation between two utterances in a dialog, which would easily make ambiguities.

4 Experiments

We now introduce our baseline experiments on our dataset. We consider the two tasks of discourse parsing and machine comprehension for multiparty dialog.

4.1 Machine reading comprehension for multiparty dialogs

Methods. SQuAD 2.0 is an MRC dataset that adopts a passage as the input and the answer is a span from input passage (Rajpurkar et al., 2018). We adopt the following existing methods for SQuAD 2.0 on our dataset. In this paper, we use three different kinds of settings of BERT: BERT-base, BERT-large, and BERT-whole word masking (BERT-wwm). We concatenate all utterances from input dialog as a passage, and each utterance includes speaker and text. We used the open-source code of BERT to perform our experiments[3].

BERT is a bidirectional encoder from transformers (Devlin et al., 2019). To learn better representations for text, BERT adopts two objectives: masked language modeling and the next sentence prediction during pretraining. In the BERT-wwm, if a part of a complete word WordPiece is replaced by [mask], the other parts of the same word will also be replaced by mask, which is the whole word mask.

[3] https://github.com/google-research/bert

Table 7: Results of machine reading comprehension for multiparty dialogs.

Method	EM		F1	
	Squad 2.0	**Our**	**Squad 2.0**	**Our**
BERT-base	73.1	45.3	76.2	58.0
BERT-large	80.0	51.8	83.1	65.5
BERT-wwm	86.7	54.7	89.1	67.7
Human performance	86.8	64.3	89.4	80.2
Human-machine gap	0.1	9.6	0.3	12.5

- **BERT-base:** 12-layer, 768-hidden, 12-heads, 110M parameters.

- **BERT-large:** 24-layer, 1024-hidden, 16-heads, 340M parameters. The difference between BERT-base and BERT-large is in the number of the parameters; there is no difference in model architecture.

- **BERT-wwm:** 24-layer, 1024-hidden, 16-heads, 340M parameters. The original word segmentation method based on WordPiece segments a complete word into several affixes. When generating training samples, these separated affixes are randomly replaced by `[mask]`.

Evaluation Metric. As our task is quite related to SQuAD 2.0, we adopt the same evaluation metrics: exact match (EM) and F_1 score to evaluate experiments. EM measures the percentage of predictions that match all words of the ground truth answers exactly. F_1 scores are a looser interpretation of match, measuring the average overlap between predictions and the ground truth answer. The results of machine reading comprehension for multiparty dialogs is shown in Table 7.

Human upper bound. We enlist two non-annotator volunteers whose majors are computer science to answer questions in the TEST set. From Table 7, they achieved 64.3% in EM and 80.2% in F_1. This result show that (1) People can get good results in F_1, and (2) it is challenging to detect the accurate boundary of answers. The results of humans show the challenge of machine comprehension for multiparty dialogs because the structure of a multiparty dialog is very complex and the language style in dialogs is very informal compared with well-written passage text.

Results. For three BERT models, the BERT-wwm model achieves the best results on both SQuAD 2.0 and our Molweni dataset, followed by BERT-large and BERT-base. Especially, the BERT-wwm model gets 89.1% F_1 score on SQuAD 2.0, very close to human performance. The performance gap between BERT-wwm and human are 0.1% EM and 0.3% F_1 on SQuAD 2.0. However, on Molweni, BERT-wwm achieves only 67.7% F_1, which has a significant large 12.5% performance gap with human performance.

Case study In this part, we will analyze the reason why BERT-wwm does not perform as well as it does on SQuAD 2.0. Fig.4 shows an example of dialog 3 in our Molweni test set with two bad cases of the BERT-wwm model. In Dialog 3, there are three speakers and ten utterances. The first question Q1 is about the user that asked for the address. The answer to BERT-wwm of Q1 is *likwidoxigen*, but the gold answer is *nbx909*. The second question Q2 is about the status of printers, but the model answers the status of people who makes the printers.

We concatenate all utterances as the input which doesn't highlight the speaker information of the utterance. For Q1, after concatenating all utterances, *likwidoxigen* would be the closest speaker in the input with the word 'address'. The speaker of utterances is the essential information for better understanding dialogs.

On the other hand, when concatenating all utterances, the language model could automatically model the coherence between two adjacent utterances. But there could be no coherence between adjacent utterances, and the discourse structure of a multiparty dialog should not be regarded as a sequence but a graph. In most cases, every node (utterance) in the discourse dependency graph only has one parent node.

Dialogue 3

nuked: ok likwidoxigen ill reboot and let you know how it goes U_1

likwidoxigen: who makes the printers ? and they woked before yets ? U_2

nuke: yes they worked excellently on dapper . they are two hp deskjets U_3

nbx909: does n't give me the address U_4

likwidoxigen: and they just dont ' print properly ? U_5

likwidoxigen: ok let me keep poking U_6

nbx909: i know but it 's a ups (battery backup) device would it be under sda ? U_7

nuked: i used kde 's add printer wizard , and only samba printers are allowed U_8

likwidoxigen: i 'd assume so , it still has to access the device U_9

likwidoxigen: damn do any usb device work ? U_{10}

(a)

Q1: Who does ask for the address?
Gold answer: *nbx909*
BERT-wwm answer: *likwidoxigen*

Q2: how are printers working?
Gold answer: *NA.*
BERT-wwm answer: they worked excellently on dapper.

(b)

Figure 4: Dialogue3. (a) A real example from Molweni dataset with three speakers and ten utterances. (b) Two questions for Dialog 3 and the pridected answers of BERT-wwm model.

Table 8: Results of discourse parsing on multiparty dialogs (F_1-score). Deep sequential (C) means combine the training set of STAC and Molweni as the training set and test the model respectively.

Method	**Link**		**Link & Relation**	
	STAC	**Our**	**STAC**	**Our**
Deep sequential	73.2	78.1	55.7	54.8
Deep sequential(C)	78.0	77.0	54.7	54.3

4.2 Discourse parsing for multiparty dialogs

Methods We perform the Deep Sequential model on our Molweni corpus which is the state-of-the-art model on STAC. Shi and Huang (2019) proposed the deep sequential model for discourse parsing on multiparty chat dialogs which adopted an iterative algorithm to learn the structured representation and highlight the speaker information in the dialog. The model jointly and alternately learns the dependency structure and discourse relations.

In this paper, we adopt two different kinds of the setting of the Deep Sequential model.

- **Deep sequential** This is the original deep sequential model.
- **Deep sequential(C)** Considering that we adopt the same discourse relation hierarchy with the STAC corpus, we combine the training sets of STAC and Molweni as the training set for this model, we respectively test the model on STAC and Molweni.

Results We adopt the F1 score to evaluate both links prediction and relation classification tasks, which is the same as previous literature. The results of discourse parsing for multiparty dialogs are shown in Table 8. For link prediction, we achieved higher results than the deep sequential model performed on STAC. On the other hand, we achieve comparable results for relations classification compared with STAC. After combining the training set of Molweni, the deep sequential model achieves better results on STAC which means the Molweni dataset can be beneficial to predict discourse dependency links.

5 Conclusion

In this paper, we introduce Molweni, a multiparty dialog dataset for machine reading comprehension (MRC). Compared with traditional textual structure, the dialog is concatenated by the utterances from multiple participants. We believe that discourse structure can provide potential help for understanding the dialog. Therefore, we ask annotators to label the discourse dependency structure of the multiparty dialog and propose questions for the dialog. Annotation on a large number of dialog shows that tagging discourse structure can significantly help taggers understand dialog and raise higher quality questions. In the future, we will try to propose novel discourse parsing models for multiparty dialog and apply discourse structure in the reading comprehension task of multiparty dialog.

Acknowledgements

We thank anonymous reviewers for their helpful comments. Thanks to Yibo Sun, Tiantian Jiang, Daxing Zhang, Rongtian Bian, Heng Zhang, Hu Zhenyu and other students of Harbin Institute of Technology for their support in the process of data set annotation. Thanks to Wenpeng Hu, a Ph.D. student of Peking University, for providing preprocessed Ubuntu data. The research in this article is supported by the Science and Technology Innovation 2030 - ”New Generation Artificial Intelligence” Major Project (2018AA0101901), the National Key Research and Development Project (2018YFB1005103), the National Science Foundation of China (61772156, 61976073) and the Foundation of Heilongjiang Province (F2018013).

References

Stergos Afantenos, Eric Kow, Nicholas Asher, and Jérémy Perret. 2015. Discourse parsing for multi-party chat dialogues. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 928–937. Association for Computational Linguistics.

Nicholas Asher, Julie Hunter, Mathieu Morey, Benamara Farah, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2721–2727.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. 2019. Gsn: A graph-structured network for multi-party dialogues. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5010–5016. International Joint Conferences on Artificial Intelligence Organization, 7.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Tomas Kocisky, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy, July. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Wenqiang Lei, Xuancong Wang, Meichun Liu, Ilija Ilievski, Xiangnan He, and Min-Yen Kan. 2017. Swim: A simple word interaction model for implicit discourse relation recognition. In *IJCAI*, pages 4026–4032.

Wenqiang Lei, Yuanxin Xiang, Yuwei Wang, Qian Zhong, Meichun Liu, and Min-Yen Kan. 2018. Linguistic properties matter for implicit discourse relation recognition: Combining semantic interaction, topic continuity and attribution. *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).*

Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196, Florence, Italy, July. Association for Computational Linguistics.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Kaixin Ma, Tomasz Jurczyk, and Jinho D Choi. 2018. Challenging reading comprehension on daily conversation: Passage completion on multiparty dialog. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2039–2048.

Jérémy Perret, Stergos Afantenos, Nicholas Asher, and Mathieu Morey. 2016. Integer linear programming for discourse parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 99–109. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don¡t know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, March.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA, October. Association for Computational Linguistics.

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7007–7014.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. Dream: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32.

Zhengzhe Yang and Jinho D. Choi. 2019. FriendsQA: Open-domain question answering on TV show transcripts. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 188–197, Stockholm, Sweden, September. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018.

Zhou Zhao, Haojie Pan, Changjie Fan, Yan Liu, Linlin Li, and Min Yang. 2019. Abstractive meeting summarization via hierarchical adaptive segmental network learning. In *The World Wide Web Conference*, pages 3455–3461. ACM.