Molweni: A Challenge Multiparty Dialogues-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

We present the Molweni dataset, a machine reading comprehension (MRC) dataset built over multiparty dialogues. Molweni's source samples from the Ubuntu Chat Corpus, including 10,000 dialogues comprising 88,303 utterances. We annotate 32,700 questions on this corpus, including both answerable and unanswerable questions. Molweni also uniquely contributes discourse dependency annotations for its multiparty dialogues, contributing large-scale (78,246 annotated discourse relations) data to bear on the task of multiparty dialogue understanding. 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 the area of multiparty dialogue has grown considerably over the recent few years, partially due to the growing ubiquity of dialogue agents. Multiparty dialogue 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 dialogue, 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 multi-party dialogues, but which has been easily dispatched by CNN+LSTM models using attention. The DREAM corpus (Sun et al., 2019) is an MRC dataset for dialogues, but only features a minute fraction (1%) of multiparty dialogues. FriendsQA (Yang and Choi, 2019) is a small-scale span-based MRC dataset for multiparty dialogues, which derives from TV show *Friends*, including 1,222 dialogues and 10,6100 questions. The number of dialogues in FriendsQA limits to training a power model to represent multiparty dialogues and the lack of discourse structure does not take full account of the characteristics of multiparty dialogues.

Dialogue-based MRC thus varies from other MRC variants in two key differences:

C1. Utterances of multiparty dialogues are not coherent. A passage is a continuous text where there is a discourse relation between every two adjacent sentences. Therefore, we can regard each paragraph in a passage as a linear discourse structure text. However, there could be no discourse relation between adjacent utterances in a multiparty dialogue. The discourse structure of a multiparty dialogue can be regarded as a dependency graph where each node is an utterance.

C2. Two-party dialogues can be regarded as the special case of multiparty dialogues. The discourse structure of a multiparty dialogue is more complex than a two-party dialogue. In most cases, the

Dialogue 1
1. **nbx909**: how do i find the address of a usb device ?
2. **Likwidoxigen**: try taking it out to dinner and do a little wine and dine and it shoudl tell ya
3. **Likwidoxigen**: what sort of device ?
4. **babo**: ca n't i just copy over the os and leave the data files untouched ?
5. **nbx909**: only if you do an upgrade
6. **Nuked**: should i just restart x after installing
7. **Likwidoxigen**: i 'd do a full restart so that it re-loads the modules
Q1: Why does **Likwidoxigen** 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*.

Table 1: A multiparty dialogue example in our Molweni dataset with four speakers and seven utterances. Due to the property of dialogues, the instances in our corpus could have grammar mistakes.

discourse structure of a two-party dialogue is tree-like, where discourse relations mostly occur between adjacent utterances. However, in multiparty dialogues, two utterances may participate in discourse relations though they are very distant.

All works do not consider the properties of multiparty dialogue. To address this gap in understanding of multiparty dialogue, we created Molweni. In Dialogue 1 (*cf.* Table 1) four speakers converse over eight utterances, where our annotators have proposed 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 dialogue without a proper understanding of the input's complex structure.

We derived Molweni from the large-scale multiparty dialogue 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 dialogue in the same language as *Ubuntu*. Our dataset contains 10,000 dialogues with 88,303 utterances and 32,700 questions including answerable and unanswerable questions. All answerable questions are extractive questions whose answer is a span in the source dialogue. For unanswerable questions, we annotate their plausible answers from dialogues. Most questions in Molweni are 5W1H questions – Why, What, Who, Where, When, Which and How. For each dialogue in the corpus, annotators propose three questions and find the answer span (if answerable) in the input dialogues.

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 dialogues, the human performance just can achieve 80.2 of F1 score 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 in Molweni. We further annotated all 78,246 discourse relations in all of Molweni's dialogues, considering the potential help that annotated discourse structure might serve. Prior to Molweni, the STAC corpus is the only dataset for multiparty dialogue discourse parsing (Asher et al., 2016). However, its limited scale (only 1K dialogue) disallow data-driven approaches to discourse parsing for multiparty dialogue. We saw the additional opportunity to empower and drive this direction of research for multiparty dialogue processing.

2 Related work

Discourse parsing for multiparty dialogues STAC is the only corpus of discourse parsing on multi-party chat dialogues (Asher et al., 2016). The corpus derives from online game *The Settlers of Catan*.

	Train	**Dev**	**Test**	**Total**
Dialogues	9,000	900	100	10,000
Utterances	79,487	7971	845	88,303
Questions	27,000	2,700	3,000	32,700

Table 2: Overview of Molweni for MRC.

The game *Settlers* is a multiparty, win-lose game. As mentioned above, the senses of discourse relation in STAC is introduced in Section 3.3. More details for the STAC corpus are described in (Asher et al., 2016). The STAC corpus contains 1091 dialogues with 10,677 utterances and 11,348 discourse relations. Compared with STAC, our Molweni dataset contains 10,000 dialogues comprising 88,303 utterances and 78,246 discourse relations.

Machine reading comprehension Machine reading comprehension is a popular task which aims to help the machine better understand natural language. 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) . Our Molweni dataset is an extractive MRC dataset for multiparty dialogue including answerable questions and unanswerable questions. Similar to Squad 2.0 (Rajpurkar et al., 2018), we also annotate plausible answers for unanswerable questions. Three most related datasets are (Ma et al., 2018), DREAM (Sun et al., 2019) and FriendsQA (Yang and Choi, 2019). Different these three MRC datasets for dialogue, we provide the discourse structure of dialogues, more instances of multiparty dialogues and unanswerable questions.

3 The Molweni corpus

3.1 Data collection

Our dataset derives from the large scale multiparty dialogues dataset the Ubuntu Chat Corpus (Lowe et al., 2015). The Ubuntu dataset is a large scale multiparty dialogues corpus.

There are several reasons to choose the Ubuntu dataset as our raw data for annotation.

- First, the Ubuntu dataset is a large multiparty dataset. Recently, (Hu et al., 2019) used Ubuntu as their dataset for learning dialogues graph representation. After filtering the dataset by choosing all utterances with response relations, there are 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 dialogues, we only need to annotate relations between the same speaker's utterances and the specific sense of discourse relation. Because the length of dialogues in the Ubuntu dataset is not too long, we can easily summarize dialogues and propose some questions for the dialogue.

- Third, there are many papers doing experiments on the Ubuntu dataset, and the dataset has beenwidely recognized. For example, (Kummerfeld et al., 2019) proposed a large-scale dataset for conversation disentanglement based on the Ubuntu IRC log.

The discourse dependency structure of each multiparty dialogue can be regarded as a discourse dependency graph where each node is an utterance. To learn better graph representation of multiparty dialogues, we adopt the dialogues with 8-15 utterances and 2-9 speakers. To simplify the task, we filter the dialogues with long sentences (more than 20 words). Finally, we choose 10,000 dialogues with 88,303 utterances from the Ubuntu dataset.

Because the Ubuntu corpus belongs to the technical question-answering domain dataset, we hire ten undergraduate students whose majors are computer science to annotate the corpus. The annotators are non-native English speakers but all have an English proficiency certificate to prove their English

Dataset	**Answer type**	**Dialogue text**	**Multiparty dialogues**	**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	✔	✔	✔	✔

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

language skills. Each annotator is asked to annotate 1,000 dialogues for the two aspects: machine reading comprehension and discourse structure. All annotators choose to firstly annotate the discourse structure of the dialogue, and then propose questions and find answers for the dialogues. All annotators agree that it would be helpful to annotate the MRC task after annotating the discourse structure.

We totally annotate 10K dialogues with 88,303 utterances, including 78,246 discourse relations and 32,700 questions for machine reading comprehension. The overview of our Molweni corpus is shown in Table 2. There are 9,000 dialogues in Train set for both machine reading comprehension tasks. 900 dialogues are used for Dev set. Each annotator is asked to propose three questions for dialogue. There are 100 dialogues in common for ten annotators, so all annotators totally propose 3,000 questions for 100 dialogues, and we use the 100 dialogues as our test set. Each dialogue in the training set and develop set has three questions.

The average speakers per dialogue in our dataset is 3.52 which means most dialogues are multiparty dialogue, and the max speakers number in Molweni is 9. The number of two-party dialogues and multiparty dialogues in our dataset is respectively 2,117 and 7,883. Because of the complexity of multiparty dialogues discourse structure, we filter the dialogues with long utterances (more than 20 words). In Molweni, the average and max length of dialogues are respectively 8.83 and 14 utterances. The number of answerable questions and unanswerable questions are 27,269 and 5,115.

3.2 Annotation for machine reading comprehension

The comparison of Molweni with other datasets is shown in Table 3. From Table 3, none of the existing dialogue MRC datasets annotate unanswerable questions and discourse structure. Due to the incoherent and complex structure of multiparty dialogues, it is essential to adopt the discourse dependency structure for understanding dialogues better. As we know, Molweni is the only MRC dataset with a discourse structure.

We propose three questions for each dialogue and annotate the span of answers in the input dialogue. There are two types of questions in our corpus, namely, answerable questions and unanswerable questions. In particular, most of the questions in our dataset are questions leading by Why, What, Who, Where, When, Which and How. Only a small part of the questions are other questions in our dataset, such as questions leading by Do, Which, and Whose. Questions leading by Why and How can require a more deep understanding of the input dialogue.

Question types based on whether the question can be answered:

- **Answerable questions** For answerable questions, the answer is a continuous span from source dialogue. The annotators are asked to label answers from input dialogue and ensure answers succinct without useless words.

- **Unanswerable questions** To improve the difficulty of the task, we annotate unanswerable questions

Question	Example	
How	How to do an upgrade?	How can I use this machine?
Why	Why is it not mounted?	Why does *jimcoonact* meet the error?
Who	Who is chart’s service customers?	Who is using ubuntu?
When	When does rhodry have the error?	When is SuperMiguel back?
Where	Where did earthen write in?	where is the device?
What	What does elnomade choose?	What does noone need?
Others	Does elnomade choose the print?	Which version does *xxiao* find?

Table 4: Examples of questions in Molweni.

Relation	Meaning
Comment	Arg2 comments Arg1.
Clarification_question	Arg2 clarifies Arg1.
Elaboration	Arg2 elaborates Arg1.
Acknowledgement	Arg2 acknowledges Arg1.
Continuation	Arg2 is the continuation of Arg1.
Explanation	Arg2 is the explanation of Arg1.
Conditional	Arg1 is the condition of Arg2 or Arg2 is the condition of Arg1.
Question-answer_pair	Arg1 is a question and Arg2 is the answer of Arg1.
Alternation	Arg1 and Arg2 denote alternative situations.
Q-Elab	Arg1 is a question and Arg2 try to elaborate Arg1.
Result	Arg2 is the effect brought about by the situation discribed in Arg1.
Background	Arg2 is the background of Arg1.
Narration	Arg2 is the narration of Arg1.
Correction	Arg2 correct Arg1.
Parallel	Arg2 and Arg1 are parallel and present almost the same meaning.
Contrast	Arg1 and Arg2 share a predicate or property and a difference on shared property.

Table 5: A two-party dialogue example in our Molweni dataset with two speakers and eight utterances.

and their plausible answers (PA). The plausible answers are quite related to unanswerable questions.

We list some examples of different kinds of questions in our Molweni corpus in Table 4. When annotators propose questions, they consider the characteristics of multiparty dialogues. For example, for ’Why’ and ’how’ questions, it is essential to know the questions answer pair and cause-result in the dialogue. For ’How’ questions, it is important to highlight the role of speakers for representing multiparty dialogues.

3.3 Annotation for discourse structure of multiparty dialogues

The task of discourse parsing for multiparty dialogues aims to detect discourse relations among utterances. The discourse structure of a multiparty dialogue is a directed acyclic graph (DAG). In the process of annotation of discourse parsing for multiparty dialogues, there are two parts: predict links between utterances and classify the sense of discourse relations.

The edge between the two utterances represents that there are the discourse dependency relations between these two utterances. The direction of the edge represents the direction of discourse dependency. In this subsection, what we need to do is to confirm whether two utterances have a discourse relation. Like PDTB (Prasad et al., 2008), we call two utterances as *Arg1* and *Arg2* respectively. The forward utterance is *Arg1* and the backward utterance is *Arg2*.

The biggest difference between discourse parsing for well-written documents and dialogues is that discourse relations can exist on two nonadjacent utterances in dialogues. When we annotate the dialogue,

Dialogue 2

1. **toma-**: but its well worth the wait
2. **woodgrain**: i have a decently fast p4 should i still be waiting ?
3. **toma-**: have you run updatedb before ?
4. **woodgrain**: no never before – but it worked and now i have all the files i need .
5. **woodgrain**: i do n't have a path to the jre – do i need to add it ?
6. **toma-**: a path ? ? you compiling somehting ?
7. **woodgrain**: do n't need jdk as witnessed by eclipse irc
8. **woodgrain**: no i 'm installing this newer ver from the eclipse site .

Table 6: A two-party dialogue with two speakers and eight utterances.

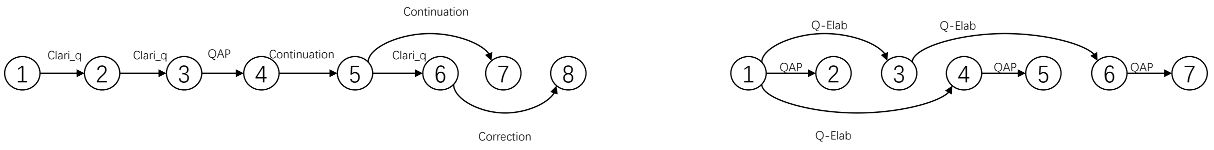

Figure 1: The discourse dependency structure and relations for Dialogue 2 (Left, two-party) and Dialogue 1 (Right, multiparty). Clari_q , QAP and Q-Elab are respectively short for Clarification_question, Question-answer_pair and Question-Elaboration. The label on the link represents the discourse dependency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

we should read the dialogue from the front to the back. For each utterance, we should find its one parent node at least from all its previous utterances. We assume that the discourse structure is a connected graph and no utterance is isolated.

When we find the discourse relation between two utterances, we need to continue to confirm the specific relation sense. We adopt the hierarchy of the same senses with the STAC dataset (Asher et al., 2016). There are sixteen discourse relations in the STAC. All relations are listed as follows: Comment, Clarification_question, Elaboration, Acknowledgement, Continuation, Explanation, Conditional, Question-answer_pair, Alternation, Q-Elab, Result, Background, Narration, Correction, Parallel, Contrast. The meaning of each discourse relation is shown in Table 5.

For the discourse parsing task, we used 500 dialogues for development and 500 dialogues for testing which is different from the MRC tasks. There are four types of relations accounting for less than 1%, namely, Alternation, Background, Narration, and Parallel. The same proportion of these four types of relations in the STAC data set is only 0.5% - 2%. 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 dialogues. The distribution of all sixteen relations in our Molweni dataset is shown in the Appendix.

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

Dialogue 1 (*cf.* Table 1) is a multiparty dialogue with four speakers: **nbx909**,**likwidoxigen**, **babo**, **nuked** and seven utterances. Dialogue 2 (*cf.* Table 6) has two speakers: **toma-** and **woodgrain**, and eight utterances in total. The discourse dependency structures of Dialogue 1 and Dialogue 2 are shown in Figure 1 where each utterance is represented as a node in the dependency graph. The label on the link in the discourse dependency relation. From Figure 1, we can find that most of the discourse relations of two-party dialogue happen between adjacent utterances.

3.4 Data Quality

To ensure the quality of the corpus, we adopt two ways to check the annotation: human check and program check.

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

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

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

- Program check. If answers cannot be found in the source dialogue, the annotator would be asked to annotate the dialogues again until passing the check. We use the only grammar checking website *Grammarly* [1] to check and correct grammar errors.

After several times of revision, we obtain the fourth version of the dataset.

We calculate the Fleiss Kappa value to represent the consistency of all annotators. 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. 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 dialogue, which would easily make ambiguities.

4 Experiments

In this section, we will introduce experiments on our dataset. We consider the following two tasks for multiparty dialogues: discourse parsing and machine comprehension.

4.1 Machine reading comprehension for multiparty dialogues

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 dialogue as a passage, and each utterance includes speaker and text. We used the open-source code of BERT to perform our experiments.

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.

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

- **BERT-large** 24-layer, 1024-hidden, 16-heads, 340M parameters. The only difference between BERT-base and BERT-large is the parameter.

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

Dialogue 3
1. **nuked**: ok likwidoxigen ill reboot and let you know how it goes
2. **likwidoxigen**: who makes the printers ? and they woked before yets ?
3. **nuked**: yes they worked excellently on dapper . they are two hp deskjets
4. **nbx909**: does n't give me the address
5. **likwidoxigen**: and they just dont ' print properly ?
6. **likwidoxigen**: ok let me keep poking
7. **nbx909**: i know but it 's a ups (battery backup) device would it be under sda ?
8. **nuked**: i used kde 's add printer wizard , and only samba printers are allowed
9. **likwidoxigen**: i 'd assume so , it still has to access the device
10. **likwidoxigen**: damn do any usb device work ?
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.

Table 8: An example Dialogue 4 from our Molweni dataset with three speakers: nuked, likwidoxigen and nbx909. Q1 and Q2 are two questions that BERT-wwm wrongly answers. The utterances are directly concatenated as the input for the BERT-wwm model.

Results

We adopt the following existing methods for Squad 2.0 on our dataset. 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 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 dialogue as a passage, and each utterance includes speaker and text.

Evaluation Metric. Our task is quite related to Squad 2.0, so we adopt the same evaluation metrics: exact match (EM) and F1 score to evaluate experiments. Em can measure the percentage of predictions that match all words of the ground truth answers exactly. F1 scores can be looser and measure the average overlap between the prediction and ground truth answer. The results of machine reading comprehension for multiparty dialogues as shown in Table 7.

Upper bound. We enlist two volunteers whose majors are computer science to answer questions in the test set. The volunteers are not annotators for the Molweni. From Table 7, human achieves 64.3% in EM and 80.2% in F1 score. The human performance shows that: (1) People can get good results in F1. (2) it is challenging to detect the accurate boundary of answers. The results of humans show the challenge of machine comprehension for multiparty dialogues because the structure of a multiparty dialogue is very complex and the language style in dialogues is very informal compared with well-written passage text.

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% F1 score on Squad 2.0 which is very close to the human performance, and performance gap between BERT-wwm and human are 0.1% EM and 0.3% F1 on squad 2.0. However, BERT-wwm only gets the 67.7% F1 score on our Molweni dataset which has a 12.5% 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. Table 8 shows an example of Dialogue 3 in our Molweni test set with two bad cases of the BERT-wwm

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

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

Table 9: Results of discourse parsing on multi-party dialogues (F1-score). Deep sequential (C) means combine the training set of STAC and Molweni as the training set and test the model respectively.

model. In Dialogue 4, 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 dialogues. 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 dialogue 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.

4.2 Discourse parsing for multiparty dialogues

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 dialogues which adopted an iterative algorithm to learn the structured representation and highlight the speaker information in the dialogue. The model jointly and alternately learns the dependency structure and discourse relations.

In this paper, we adopt two different kinds of 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 dialogues are shown in Table 9. 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 dialogues dataset for machine reading comprehension (MRC). Compared with traditional textual structure, the dialogue is concatenated by the utterances from multiple participants. We believe that discourse structure can provide potential help for understanding the dialogue. Therefore, we ask annotators to label the discourse dependency structure of the multiparty dialogue and propose questions for the dialogue. Annotation on a large number of dialogues shows that tagging discourse structure can significantly help taggers understand dialogues and raise higher quality questions. In the future, we will try to propose novel discourse parsing models for multiparty dialogues and apply discourse structure in the reading comprehension task of multiparty dialogues.

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, Farah Benamara, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: The stac corpus.

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.

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.

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 dont 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.

A Appendices

Table 10 shows the distribution of all sixteen relations in our dataset. There are four types of relations accounting for less than 1%, namely, Alternation, Background, Narration, and Parallel. The same proportion of these four types of relations in the STAC data set is only 0.5% - 2%. Next, according to the distribution of all kinds of relations, we need to consider merging some rare relation types, so as to propose a more practical sense hierarchy for multiparty dialogues.

Table 11 shows some detailed statistics of our corpus. From Table 5, we can know that most of the dialogues have three or more speakers in the dialogue which would provide enough data for modeling multiparty dialogues. The ratio of answerable questions and unanswerable questions is about 6:1.

	Train	**Dev**	**Test**	**Total**
Comment	22,510	1,205	1,079	24,794
Clarification_question	16,949	850	954	18,753
Elaboration	1,624	57	60	1,741
Acknowledgement	2,281	116	131	2,528
Continuation	4,679	319	278	5,276
Explanation	1,061	93	119	1,273
Conditional	704	46	46	796
Question-answer_pair	14,172	746	807	15,765
Alternation	178	9	0	187
Q-Elab	2,122	144	117	2,383
Result	1,747	135	141	2,023
Background	260	8	19	287
Narration	204	8	60	272
Correction	916	44	53	1,013
Parallel	166	11	12	189
Contrast	880	49	37	966
TOTAL	70,453	3880	3,913	78,246

Table 10: The distribution of sixteen discourse relations in our Molweni corpus.

Metric	**Number**
Avg./Max. of speakers per dialogue	3.52 / 9
Avg./Max. question length (in tokens)	5.86 / 18
Avg./Max. answer length (in tokens)	3.82 / 19
Avg./Max. dialogue length (in tokens)	85 / 169
Avg./Max. dialogue length (in utterances)	8.83 / 14
Questions per dialogue in Train/Dev	3
Questions per dialogue in Test	30
Vocabulary size	17,924
Answerable questions	26,376
Unanswerable questions	4,386

Table 11: Detailed statistics for our corpus .