Design and Challenges of Cloze-Style Reading Comprehension Tasks on Multiparty Dialogue

Changmao Li
Emory University
400 Dowman Dr.
Atlanta, GA 30322, USA
changmao.li@emory.edu

Tianhao Liu
Emory University
400 Dowman Dr.
Atlanta, GA 30322, USA
tliu236@emory.edu

Jinho Choi
Emory University
400 Dowman Dr.
Atlanta, GA 30322, USA
jinho.choi@emory.edu

arXiv:1911.00773v1 [cs.CL] 2 Nov 2019

Abstract

This paper analyzes challenges in cloze-style reading comprehension on multiparty dialogue and suggests two new tasks for more comprehensive predictions of personal entities in daily conversations. We first demonstrate that there are substantial limitations to the evaluation methods of previous work, namely that randomized assignment of samples to training and test data substantially decreases the complexity of cloze-style reading comprehension. According to our analysis, replacing the random data split with a chronological data split reduces test accuracy on previous single-variable passage completion task from 72% to 34%, that leaves much more room to improve. Our proposed tasks extend the previous single-variable passage completion task by replacing more character mentions with variables. Several deep learning models are developed to validate these three tasks. A thorough error analysis is provided to understand the challenges and guide the future direction of this research.

Introduction

Reading comprehension is an intriguing task that assesses a machine's ability in understanding evidence contexts through question answering. Most previous work in reading comprehension has focused on either formal documents Rajpurkar et al. [2016] or children's stories Richardson, Burges, and Renshaw [2013]. Only few approaches have attempted comprehension on multiparty dialogue Ma, Jurczyk, and Choi [2018]. However, with the explosive expansion of social media, data on dialogue has become dominant on the web. Inspired by various options of analytic models and the potential of the dialogue processing market, we extend the corpus presented by Ma, Jurczyk, and Choi [2018] for comprehensive predictions of personal entities in multiparty dialogue and develop deep learning models to make robust inference on their contexts.

Passage completion on multiparty dialogue is one of the reading comprehension tasks that requires a model to match the conversational dialogues with the formal (passage) writings. By building a robust model for this task, people can tell the status of their favorite characters by checking synopsis to see if their favorite characters appear in specific episodes without watching the entire series, thus allowing them an efficient way to decide whether to watch a particular episode. Involving matching contexts between colloquial (dialog) and formal (passage) writings makes this task extremely challenging.

Distinguished from the previous work that only focused on a single variable per passage Ma, Jurczyk, and Choi [2018], we propose two new passage completion tasks on multiparty dialogue which increase the task complexity by replacing more character mentions with variables with a better motivated data split. The details of these tasks are in Section . Several deep neural network models are developed to validate the three reading comprehension tasks. Based on the experimental results, we aim to identify main challenges in these tasks and suggest future research directions.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Related Work

The CNN/Daily Mail dataset introduced by Hermann et al. [2015] is to infer the missing entity (answer a) of a question q from all the possible entities which appear in a passage p while the passage is a news article, the question is a cloze-style task, in which one of entities is replaced by a placeholder, and the answer is this questioned entity. Many other models have been proposed to tackle this dataset, which are either RNN and attention based Chen, Bolton, and Manning [2016] or CNN and RNN based Trischler et al. [2016] or gated attention based Dhingra et al. [2017] or attention over attention based Cui et al. [2017]. Finally, the pretrained bi-directional transformer encoder(BERT) which was introduced by Devlin et al. [2018] shows that the pretraining of the language representations can bring better performance in several downstream tasks including machine reading comprehension. More datasets are available for another type of a reading comprehension task, that is multiple choice question answering, such as MCTest Richardson, Burges, and Renshaw [2013] , TriviaQA Joshi et al. [2017], RACE Lai et al. [2017] and Dream Sun et al. [2019].

Unlike the above tasks where documents and queries are written in a similar writing style, the multiparty dialogue reading comprehension task introduced by Ma, Jurczyk, and Choi [2018] has a very different writing style between dialogues and queries. However, their randomized assignment

of samples to training and test data substantially decreases the complexity of cloze-style reading comprehension. In this paper, we address this issue by introducing a chronological data split and new variants of the cloze-style reading comprehension task to challenge even higher task complexity.

Corpus

Our corpus comes from the transcripts of the TV show *Friends* with ten seasons collected by the Character Mining project.[1] Each season contains about 24 episodes, each episode is split into about 13 scenes, where each scene comprises a sequence of about 21 utterances. This dataset contains several layers of annotation. The first two seasons of the show for an entity linking task was annotated by Chen and Choi [2016]. Plot summaries of all episodes for the first eight seasons were collected by Jurczyk and Choi [2017] to evaluate a document retrieval task. The rest of the plot summaries were collected by Ma, Jurczyk, and Choi [2018]. Table 1 shows the statistical data of the corpus from Ma, Jurczyk, and Choi [2018].

Type	Count
# of dialogs	1,681
# of plots	4,646
Avg. # of utterances per dialog	15.8
Avg. # of tokens per dialog/plot	290.8 / 19.9
Avg. # of mentions per dialog/plot	24.4 / 3.0
Avg. # of entities per dialog/plot	5.4 / 2.2
Max # of entities per dialog/plot	16 / 7

Table 1: Corpus statistical data from Ma, Jurczyk, and Choi [2018]

Based on the above corpus we created a new data split different from Ma, Jurczyk, and Choi [2018]'s data split. In the previous work of Ma, Jurczyk, and Choi [2018], they used a random data split where 1,187 of 1,349 queries in the development set and 1,207 of 1,353 queries in the test set are generated from the same plot summaries as some queries in the training set with only masking the different character entities which makes the model can see the right answer in the training set. To fix this issue, we created a new data split, the detail of which is in Section .

Tasks

Table 2a and Table 2b show an example of a dialogue and its plots.We propose three tasks, one is from Ma, Jurczyk, and Choi [2018], and another two tasks are new tasks designed by us.

Single Variable task(SV)

The single variable task from Ma, Jurczyk, and Choi [2018] consists a dialogue passage *p*, a query *q* which is from plot summary of the dialogue passage and an answer *a*. In this task, a query *q* replaces only one character entity with an unknown variable *x* and the machine is asked to infer the replaced character entity (answer *a*) from all the possible entities appear in the dialogue passage *p*. This task is evaluated by computing the accuracy of predictions (see Section). Table 2c shows an example of this task.

Multiple Variable task on the Same entity(MVS)

Similar to the first task, the multiple variable task on the same entity also consists a dialogue passage *p*, a query *q* which is from plot summary of the dialogue passage and an answer *a*. The difference is that the query *q* replaces all the same character entities with the same unknown variable *x* and the machine is asked to infer this same character entity (answer *a*) from all the possible entities appear in the dialogue passage *p*. The difficulty of this task is that the machine has to predict a character entity who does not appear in the query. The evaluation of this task is the same as the first task since the model only predicts one entity (see Section). Table 2d shows an example of this task.

Two Variables task(TV)

To make this task more intriguing, we propose two new tasks. This task consists a dialogue passage *p*, a query *q* which is from plot summary of the dialogue passage and an answer pair a_1 and a_2. Here the query replaces two character entities with two different unknown variables x_1 and x_2 and the machine is asked to infer these two missing character entities(the answer pair a_1 and a_2) from all the possible entities appear in the dialogue passage *p*. Two replaced character entities can be the same or different and to avoid bias between two unknown variables, the different order of the variables will be considered as different queries. Since there are several plot summaries which only have one character entity in it, the model needs to be designed to predict one or two entities, evaluated by the F1 score (Section).

Approaches

We selected some published approaches which is to solve dialogue reading comprehension to validate on our new data split and three tasks.

BiLSTM

The LSTM model which is selected as one of the baselines methods in this project. We deal with utterances and query separately. First we generate the utterances and query embedding. For utterances we combine them into document-level matrix and input them into Bidirectional LSTM and get the output h_d. For query, we directly input them into BiLSTM and get the output h_q. Finally we concatenate the output of h_d and h_q and pass them into a softmax classification layer. By using Bidirectional LSTM, we can extract the sequence feature of utterances and queries. Figure 1 shows the architecture for this model.

CNN+BiLSTM

Based on Ma, Jurczyk, and Choi [2018], we first use CNN to extract the gram-level features of utterances and then use

[1] https://github.com/emorynlp/character-mining

(a) A dialogue from *Friends*: Season 1, Episode 21, Scene 1.

ID	Speaker	Utterance
1	**@ent04**	How could someone get a hold of your credit card number ?
2	**@ent00**	I have no idea . But look how much they spent !
3	**@ent01**	**@ent00** , would you calm down ? The credit card people said that you only have to pay for the stuff that you bought .
4	**@ent00**	I know . It 's just such reckless spending .
5	**@ent06**	I think when someone steals your credit card , they 've kind of already thrown caution to the wind .
6	**@ent05**	Wow , what a geek . They spent $ 69.95 on a Wonder Mop .
7	**@ent00**	That 's me .
...	...	...

(b) Plots generated for the dialog in(a)

ID	Plots
1	**@ent00** spent $ 69.95 on a Wonder Mop
2	**@ent04** asks **@ent00** how someone could get a hold of **@ent00** 's credit card number and **@ent00** is surprised at how much was spent .

(c) Queries generated from the passages in (b) for single variable task

ID	Queries
1.a	***x*** spent $ 69.95 on a Wonder Mop
2.a	***x*** asks **@ent00** how someone could get a hold of **@ent00** 's credit card number and **@ent00** is surprised at how much was spent .
2.b	**@ent04** asks ***x*** how someone could get a hold of **@ent00** 's credit card number and **@ent00** is surprised at how much was spent .
2.c	**@ent04** asks **@ent00** how someone could get a hold of ***x*** 's credit card number and **@ent00** is surprised at how much was spent .
2.d	**@ent04** asks **@ent00** how someone could get a hold of **@ent00** 's credit card number and ***x*** is surprised at how much was spent .

(d) Queries generated from the passages in (b) for multiple variables task on the same entity

ID	Queries
1.a	***x*** spent $ 69.95 on a Wonder Mop
2.a	***x*** asks **@ent00** how someone could get a hold of **@ent00** 's credit card number and **@ent00** is surprised at how much was spent .
2.b	**@ent04** asks ***x*** how someone could get a hold of ***x*** 's credit card number and ***x*** is surprised at how much was spent .

(e) Queries generated from the passages in (b) for two variables task

ID	Queries
1.a	***x1*** spent $ 69.95 on a Wonder Mop
1.b	***x2*** spent $ 69.95 on a Wonder Mop
2.a	***x1*** asks ***x2*** how someone could get a hold of **@ent00** 's credit card number and **@ent00** is surprised at how much was spent .
2.b	***x2*** asks ***x1*** how someone could get a hold of **@ent00** 's credit card number and **@ent00** is surprised at how much was spent .
2.c	***x1*** asks **@ent00** how someone could get a hold of ***x2*** 's credit card number and **@ent00** is surprised at how much was spent .
2.d	***x2*** asks **@ent00** how someone could get a hold of ***x1*** 's credit card number and **@ent00** is surprised at how much was spent .
2.e	**@ent04** asks ***x1*** how someone could get a hold of ***x2*** 's credit card number and **@ent00** is surprised at how much was spent .
...	...

Table 2: An dialogue and its plots from the corpus and generated queries for each task. ***x***, ***x1*** and ***x2*** denote unknown variables.

LSTM to capture the sequence feature of both utterances and query. CNN is applied into each utterance to extract more token level n-gram features then we take max pooing and combine them into a document-level matrix, and input it into the Bidirectional LSTM and get the output h_q. For query, we directly into them into Bidirectional LSTM and get the output h_q, too. Finally we concatenate the output of h_d and h_q and pass them into a softmax classification layer. By using cnn and Bidirectional LSTM theoretically we can both capture the n-grams features and sequence features.

CNN+BiLSTM+UA+DA

This method is the SOTA method last year in Ma, Jurczyk, and Choi [2018]'s data split which is also selected as one of our experimental methods. Similar to the last one, they still use CNN to extract token-level n-gram features of utterances and use LSTM to capture the sequence feature, however, they added the utterance-level attention and document-level attention to extract more features related to similarity between query and utterances. This utterance-level attention basically is to compute the similarity between each utterance and query. This document level attention is to compute the similarity between query and whole document matrix. Their output is concatenated with original Bidirectional LSTM output and pass them into a softmax classification layer. By adding the attention layers, theoretically we can capture the some similarity features between utterances and query.

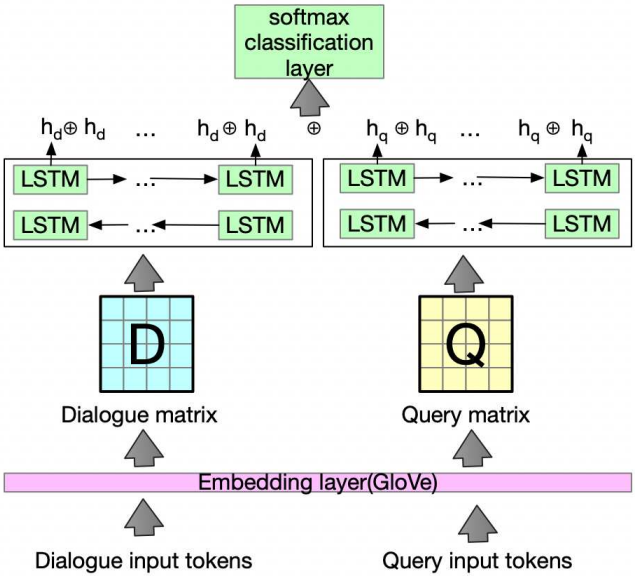

Figure 1: Architecture for BiLSTM model

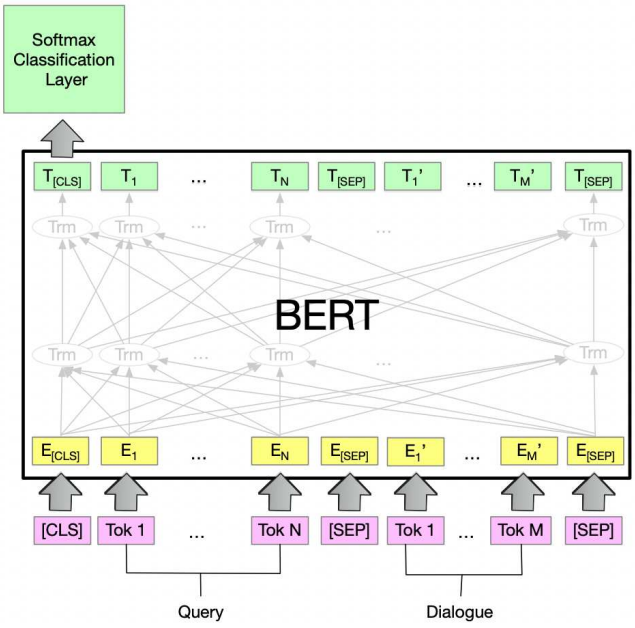

Figure 2: Architecture for BERT+Fine-tuning model

BERT+Fine-tuning

Since BERT model Devlin et al. [2018] shows very wide usage in the NLP field, in this paper, we also use the BERT fine-tuning model as one of our exploratory methods. Our input format for BERT is [CLS]query[SEP]utterances[SEP]. We use $\text{BERT}_{\text{BASE}}$ for fine-tuning since we do not have large memory of GPUs to run $\text{BERT}_{\text{LARGE}}$. What we do is to simply input the [CLS] representation of BERT final output to our top layer. Our top layer is a softmax classification layer $W \in R^{K \times H}$, where H is the dimension of the [CLS] representation and K is the number of entity labels. Figure 2 shows the architecture for this model.

Experiments

Data Split Overview

Our data split has three parts: for each season, we use episodes 1-18 as training data, episodes 19-21 as development data, and the rest of episodes as test data. This data split is chosen to mimic the training data which a model would use in an application setting for reading comprehension, where historic data is used for training while recent data is the subject for prediction. Table 3 shows the dataset split for our experiments that roughly gives 80/10/10% for training/development/test sets.

Task	Training	Development	Test
SV	11,104	1,486	1,242
MVS	8,211	1,099	916
TV	22,332	3,174	2,612

Table 3: The number of queries in each dataset.

Evaluation metrics

We use different evaluation metrics for different tasks. For the SV task and the MVS task: we use simple accuracy since the model only predicts one variable:

$$score = \frac{C_r}{C_t} \tag{1}$$

Where C_r is the number of the right predictions and C_t is the number of the total predictions.

For the TV task: we use F1 score, since model predicts one variable or two variables:

$$precision = \frac{C_r}{C_a} \tag{2}$$

$$recall = \frac{C_r}{C_g} \tag{3}$$

$$F1 = 2 \times \frac{precision * recall}{precision + recall} \tag{4}$$

Where C_r is the number of the right predictions and C_a is the number of the actual predictions and C_g is the number of the gold answer.

Results and error analysis

Results Table 4 shows the results of our experiment. BiLSTM is good at capturing the sequence information of sentences; however, since it only finds some kind of answer distributions on the sequence information, it cannot capture the information of the relation between query and utterance.

Adding a CNN can achieve even lower accuracy because passing sequences to the CNN only keeps important information after the pooling operation, but for dialogue data, most of the time the replaced entity needs to be decided by

Task	Score	
SV	**Dev**	**Test**
BiLSTM	36.72(±0.38)	**33.69(±0.90)**
CNN+BiLSTM	31.11(±0.93)	31.65(±0.25)
CNN+BiLSTM+UA+DA	30.32(±1.72)	29.10(±0.64)
BERT+Fine-tuning	32.30(±0.66)	29.83(±0.75)
MVS	**Dev**	**Test**
BiLSTM	50.26(±0.47)	**52.68(±0.63)**
CNN+BiLSTM	46.82(±2.59)	44.98(±2.18)
CNN+BiLSTM+UA+DA	49.02(±1.27)	51.92(±0.29)
BERT+Fine-tuning	48.39(±4.17)	47.51(±4.25)
TV	**Dev**	**Test**
BiLSTM	31.30(±0.52)	22.13(±1.45)
CNN+BiLSTM	27.92(±0.03)	19.97(±1.43)
CNN+BiLSTM+UA+DA	28.55(±0.13)	21.62(±2.92)
BERT+Fine-tuning	29.50(±0.83)	**30.92(±0.63)**

Table 4: Results from all models.

so much original context, when using CNN, the model loses some original context.

Utterance-level attention and document-level attention proposed by Ma, Jurczyk, and Choi [2018] are not helpful for these tasks on our data split because dialogues contain so many informal expressions and the size of the corpus is small. With limited data size, the attention mechanism does not perform well to match different expressions with the same meaning.

The BERT model only uses attention mechanism to model the language feature, however, the pre-trained model is from formal text(wikipeida, etc) not from dialogue corpus, so it also performs poorly on our data split and tasks.

A counter-intuitive part is that the MVS task appears to be so much easier than the SV task. The main reason is because the model can easily learn that the entity that appears in the query will not appear in the answer which will reduce many answer choices. Another counter-intuitive part is that BiLSTM without any additive method is the best model in two of the three tasks. The main reason of this is because the attention mechanism does not work well for these tasks on this dataset.

Error analysis For further research, we extracted 100 samples from test set of each task on their best performing model to analyze what kind of error may occur. We found that the three main types of errors for these tasks are: Hidden Meaning, Utterances Reasoning & Summary as well as Coreference Resolution. Table 5 shows the error types distribution of each task. Table 6 shows the examples of three main error types.

Hidden Meaning Hidden meaning is one of the most frequent error types and it happens when a query uses formal expressions to represent the same meaning as an utterance with informal expressions. This type of error also happens when a person uses an expression but with another meaning behind it. The model cannot find the right position of the corresponding entities.

Utterances Reasoning and Summary Utterances reasoning and summary is another frequent error type. This type of error occurs when the model needs to predict an entity based on several actions in the query which correspond to several continuous utterances in the dialogue. We noticed that the model also needs to infer the result or effect of continuous or noncontinuous actions happening in several utterances to predict correct entities.

Error types	SV	MVS	TV
Hidden meaning	22%	**28%**	**35%**
Utterance reasoning and summary	**30%**	27%	24%
Coreference resolution	18%	22%	22%
Object Linking	14%	11%	-
Annotation	7%	6%	9%
Handle single variable	-	-	5%
Miscellaneous	9%	6%	5%

Table 5: Percentage of error types occurred in each task

Coreference Resolution This type of error occurs when there are coreferences in the utterances and query and it is confusing to determine what is the correct entity behind the pronoun. This type of error occurs when there are numerous entities in the dialogue and the model cannot differentiate them because of less recognition of coreferences.

Object Linking This type of error refers to a situation when a query needs to know which entity has some kind of relationship with some kind of object(for example, another entity, substance, etc.). It can be especially confusing when the object and entity are not in the same utterance.

Annotation Issues This type of error refers to some issues when doing annotation, such as mislabeling of answers, unanswerable or irrelevant queries, misspellings in the utterance, mislabeling the entity in the dialogues, etc. This type of error is trivial and hard to solve but still reflects the real situation in human communication.

Handle Single Variable This type of error occurs when solving the TV task since the TV task can either have a single variable or two variables. The model cannot tell the differences between the single variable and two variables, and most of the time this occurs when the correct answer is only one variable, the model predicts two variables.

Miscellaneous This type of error represents error samples in which we cannot find apparent reason for the wrong prediction. We need to research more to find them.

Conclusion and Future Work

In this paper, we addressed the issue with the evaluation method of previous work by introducing a new data split and introduced new variants of the cloze-style reading comprehension task to increase task complexity. In addition, we ran several neural network models to validate our new data split and new tasks. By introducing evaluation methods that better motivate the dialogue reading comprehension task, we discovered many challenges that the model cannot easily tackle in the real situation. Finally, we propose what we will do in the next step to address some of these challenges.

In order to understand the hidden meaning in the dialogues, we plan to create dialogue specific meaning representation to enable the model to learn the hidden meaning

(a) Error example for hidden meaning: "you two together" = "have a relationship"

Answer	Query
@ent01	**x** doesn't want to have a relationship with**@ent04** again.
Speaker	**Utterance**
...	...
@ent03	I 'm sorry , for the last time , why aren't you two together again ? (silence from **@ent01** .) no , I know . I know , because you 're not in that place . which would be fine , except you totally are .
...	...

(b) Error example for utterances reasoning and summary: inference of "with" who?

Answer	Query
@ent02	**@ent03** was with x picking out the ring.
Speaker	**Utterance**
...	...
@ent03	what ?
@ent04	**@ent02** 's gonna ask **@ent05** to marry him !
@ent03	oh I know , I helped pick out the ring .
-	(**@ent02** laughs , turns , and sees that **@ent04** and **@ent00** aren't happy .)
...	...

(c) Error example for coreference resolution: resolution of "he"

Answer	Query
@ent05	**@ent01** tries to leave the set , but **@ent02** tells him that so long as **x** is conscious and present on set , they will continue to shoot the film .
Speaker	**Utterance**
...	...
@ent01	**@ent03** ! you got ta let me go . the guy 's hammered !
@ent02	I 'm sorry **@ent01** , as long as he 's here and he 's conscious we 're still shooting .
-	(he walks away and **@ent01** does **@ent04** 's fist thing . he then enters **@ent05** 's dressing room , to find **@ent05** cutting his steak with his sword .)
@ent05	you would n't happen to have a very big fork ?
...	...

Table 6: Error examples for three main error types

of what the entity is trying to express. However, this kind of approach is also limited by its labeled data size. We will try to find unsupervised ways to enable the model to learn the semantic information itself. Additionally, we will intrigue more approaches to enable model's understanding of hidden meaning in the dialogue. For example, we will design more appropriate unsupervised pre-training tasks targeting to enable the model to learn the corresponding hidden meaning of expressions which have hidden meaning.

Furthermore, to overcome reasoning and summary challenges, we plan to incorporate knowledge base reasoning by either constructing a knowledge graph or using more dialogue data to design pre-training tasks to enable the model's ability to do inference from dialgoues. Another approach we plan to try is to use a neural network model such as graphic convolution network Defferrard, Bresson, and Vandergheynst [2016] that can represent a knowledge graph structure based on dialogue dataset.

References

Chen, Y.-H., and Choi, J. D. 2016. Character identification on multiparty conversation: Identifying mentions of characters in tv shows. In *SIGDIAL Conference*.

Chen, D.; Bolton, J.; and Manning, C. D. 2016. A thorough examination of the cnn/daily mail reading comprehension task. *CoRR* abs/1606.02858.

Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; and Liu, T. 2017. Attention-over-attention neural networks for reading comprehension. In *ACL*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 3844–3852. USA: Curran Associates Inc.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

Dhingra, B.; Liu, H.; Cohen, W. W.; and Salakhutdinov, R. 2017. Gated-attention readers for text comprehension. In *ACL*.

Hermann, K. M.; Kociský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *NIPS*.

Joshi, M. S.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. S. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Jurczyk, T., and Choi, J. D. 2017. Cross-genre document retrieval: Matching between conversational and formal writings. *CoRR* abs/1707.04538.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. H. 2017. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*.

Ma, K.; Jurczyk, T.; and Choi, J. D. 2018. Challenging reading comprehension on daily conversation: Passage completion on multiparty dialog. In *NAACL-HLT*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Richardson, M.; Burges, C. J. C.; and Renshaw, E. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*.

Sun, K.; Yu, D.; Chen, J.; Yu, D.; Choi, Y.; and Cardie, C. 2019. Dream: A challenge dataset and models for dialogue-based reading comprehension. *CoRR* abs/1902.00164.

Trischler, A.; Ye, Z.; Yuan, X.; Bachman, P.; Sordoni, A.; and Suleman, K. 2016. Natural language comprehension with the epireader. In *EMNLP*.