Challenging Reading Comprehension on Daily Conversation: Passage Completion on Multiparty Dialog

Kaixin Ma, Tomasz Jurczyk, Jinho D. Choi
Math and Computer Science
Emory University
Atlanta, GA 30322, USA
{kaixin.ma, tomasz.jurczyk, jinho.choi}@emory.edu

Abstract

This paper presents a new corpus and a robust deep learning architecture for a task in reading comprehension, passage completion, on multiparty dialog. Given a dialog in text and a passage containing factual descriptions about the dialog where mentions of the characters are replaced by blanks, the task is to fill the blanks with the most appropriate character names that reflect the contexts in the dialog. Since there is no dataset that challenges the task of passage completion in this genre, we create a corpus by selecting transcripts from a TV show that comprise 1,681 dialogs, generating passages for each dialog through crowdsourcing, and annotating mentions of characters in both the dialog and the passages. Given this dataset, we build a deep neural model that integrates rich feature extraction from convolutional neural networks into sequence modeling in recurrent neural networks, optimized by utterance and dialog level attentions. Our model outperforms the previous state-of-the-art model on this task in a different genre using bidirectional LSTM, showing a 13.0+% improvement for longer dialogs. Our analysis shows the effectiveness of the attention mechanisms and suggests a direction to machine comprehension on multiparty dialog.

1 Introduction

Reading comprehension that challenges machine's ability to understand a document through question answering has gained lots of interests. Most of the previous works for reading comprehension have focused on either children's stories (Richardson et al., 2013; Hill et al., 2016) or newswire (Hermann et al., 2015; Onishi et al., 2016). Few approaches have attempted comprehension on small talks, although they are evaluated on toy examples not suitable to project real-life performance (Weston et al., 2015). It is apparent that the main stream of reading comprehension has not been on the genre of multiparty dialog although it is the most common and natural means of human communication. The volume of data accumulating from group chat or messaging continues to outpace data accumulation from other writing sources. The combination of available and rapidly developing analytic options, a marked need for dialogue processing, and the disproportionate generation of data from conversations through text platforms inspires us to create a corpus consisting of multiparty dialogs and develop learning models that make robust inference on their contexts.

Passage completion is a popular method of evaluating reading comprehension that is adapted by several standardized tests (e.g., SAT, TOEFL, GRE). Given a document and a passage containing factual descriptions about the contexts in the document, the task replaces keywords in the passage with blanks and asks the reader to fill in the blanks. This task is particularly challenging when the document is in a form of dialog because it needs to match contexts between colloquial (dialog) and formal (passage) writings. Moreover, a context that can be described in a short passage, say a sentence, tends to be expressed across multiple utterances in dialog, which requires discourse-level processing to make the full interpretation of the context.

This paper introduces a new corpus for passage completion on multiparty dialog (Section 3), and a deep learning architecture that produces robust results for understanding dialog contexts (Section 4). Our experiments show that models trained by this architecture significantly outperform the previous state-of-the-art model using bidirectional LSTM, especially on longer dialogs (Section 5). Our analysis highlights the comprehension of our models for matching utterances in dialogs to words in passages (Section 6). To the best of our knowledge, this is the first time that the sentence completion task is thoroughly examined with a challenging dataset on multiparty dialog using deep learning models.

Proceedings of NAACL-HLT 2018, pages 2039–2048
New Orleans, Louisiana, June 1 - 6, 2018.

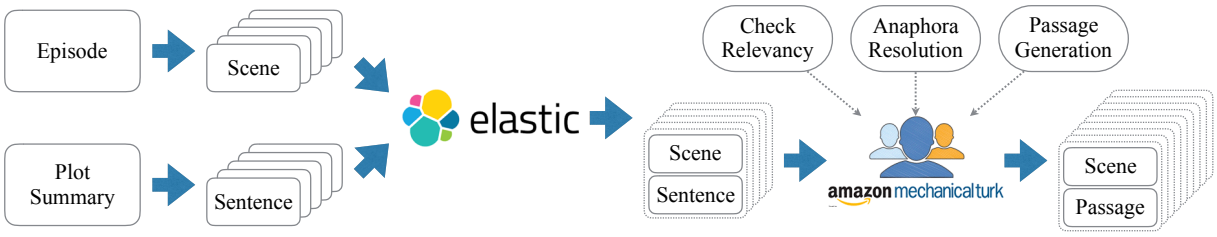

Figure 1: The overview of passage generation. Each episode is split into scenes, and each summary is segmented to sentences. Elasticsearch passes the scene-sentence pairs to crowd workers who are asked to check the relevancy, replace all pronouns with the corresponding names, and generate new passages for the scenes (Section 3.1).

2 Related Work

Hermann et al. (2015) introduced the CNN/Daily Mail dataset where documents and passages were news articles and their summaries respectively, and evaluated neural models with three types of readers. Chen et al. (2016) proposed the entity centric model and the bidirectional LSTM model using attention, and conducted a thorough analysis on this dataset. Trischler et al. (2016) presented the EpiReader that combined a reasoner with an extractor for encoding documents and passages using both CNN and RNN. Dhingra et al. (2017) proposed the gated-attention reader that incorporated attention on multiplicative interactions between documents and passages. At last, Cui et al. (2017) introduced the attention-over-attention reader that placed document-to-passage attention over passage-to-document attention.

Hill et al. (2016) released the Children Book Test dataset where documents were children's book stories and passages were excerpts from those stories. Paperno et al. (2016) introduced the LAMBADA dataset comprising novels from the Book corpus. Onishi et al. (2016) introduced the Who-did-What dataset consisting of articles from the LDC English Gigaword newswire corpus. All corpora described above provide queries, that are passages where certain words are masked by blanks, for the evaluation of passage completion. More datasets are available for another type of a reading comprehension task, that is multiple choice question answering, such as MCTest (Richardson et al., 2013), TriviaQA (Joshi et al., 2017), RACE (Lai et al., 2017), and SQuAD (Rajpurkar et al., 2016).

Unlike the other corpora where documents and passages are written in a similar writing style, they are multiparty dialogs and plot summaries in our corpus, which have very different writing styles. This raises another level of difficulty to match contexts between documents and queries for the task of passage completion.

3 Corpus

The Character Mining project provides transcripts of the TV show *Friends* for ten seasons in the JSON format.[1] Each season contains ≈24 episodes, each episode is split into ≈13 scenes, where each scene comprises a sequence of ≈21 utterances. Chen et al. (2017) annotated the first two seasons of the show for an entity linking task, where personal mentions (e.g., *she*, *mom*, *Rachel*) were identified by their corresponding characters. Jurczyk and Choi (2017) collected plot summaries of all episodes for the first eight seasons to evaluate a document retrieval task that returned a ranked list of relevant documents given any sentence in the plot summaries.

For the creation of our corpus, we collect more plot summaries for the last two seasons of *Friends* from the fan sites suggested by Jurczyk and Choi (2017), generate passages for each dialog using the plot summaries and crowdsourced descriptions (Section 3.1), then annotate mentions of all characters in both the dialogs and the passages for passage completion (Section 3.2).

3.1 Passage Generation

An episode consists of multiple scenes, which may or may not be coherent. In our corpus, each scene is considered a separate dialog. The lengths of the scenes vary from 1 to 256 utterances; we select only scenes whose lengths are between 5 and 25 utterances as suggested by the previous works (Chen and Choi, 2016; Jurczyk and Choi, 2017), which notably improves the readability for crowd workers, resulting higher quality annotation.

The plot summaries collected from the fan sites are associated with episodes, not scenes. To break down the episode-level summaries into scene-level, they are segmented into sentences by the tokenizer in NLP4J.[2] Each sentence in the plot summaries

[1] nlp.mathcs.emory.edu/character-mining
[2] https://github.com/emorynlp/nlp4j

(a) A dialog from *Friends*: Season 8, Episode 12, Scene 2.

ID	Speaker	Utterance
1	-	[Scene: Central Perk, **@ent01** and **@ent02** are there as **@ent03** enters.]
2	**@ent03**	Hey! Oh, I'm so glad you guys are here. I've been dying to tell someone what happened in the Paleontology department today.
3	**@ent01**	(To **@ent02**) Do you think he saw us or can we still sneak out?
4	**@ent03**	Professor **@ent04**, the head of the department, so ...
5	**@ent02**	They made you head of the department!
6	**@ent03**	No, I get to teach one of his advanced classes! Why didn't I get head of the department?
7	**@ent01**	Oh! Hey **@ent02**, listen umm ...
8	**@ent02**	Yeah.
9	**@ent01**	I got a big date coming up, do you know a good restaurant?
10	**@ent02**	Uh, **@ent05**'s Cafe. They got great food and it's really romantic.
11	**@ent01**	Ooh, great! Thanks!
12	**@ent02**	Yeah! Oh, and then afterwards you can take her to the Four Seasons for drinks. Or you go downtown and listen to some jazz. Or dancing - Oh! Take her dancing!
13	**@ent01**	You sure are naming a lot of ways to postpone xxx, I'll tell ya ...
14	**@ent02**	Ooh, I miss dating. Gettin' all dressed up and going to a fancy restaurant. I'm not gonna be able to do that for so long, and it's so much fun! I mean not that sitting at home worrying about giving birth to a sixteen pound baby is not fun.
15	**@ent01**	Hey, y'know what?
16	**@ent02**	Huh?
17	**@ent01**	Why don't I take you out?
18	**@ent02**	What?! **@ent01**, you don't want to go on a date with a pregnant lady.
19	**@ent01**	Yes I do! And we're gonna go out, we're gonna have a good time, and take your mind off of childbirth and c-sections and-and giant baby heads stretching out ...
20	**@ent02**	(interrupting) Okay! I'll go with ya! I'll go! I'll go with ya.
21	**@ent01**	I'll be fun.
22	**@ent02**	All right?

(b) Passages generated for the dialog in (a).

ID	Passage
1	**@ent03** announces that **@ent03** is going to be teaching a graduate class at the university.
2	**@ent02** misses dressing up for romantic dates so **@ent01** promises to take **@ent02** out.
3	**@ent02** misses dating, so **@ent01** promises to show **@ent02** a good time.
4	**@ent01** asks **@ent02** where to go on a date and then **@ent01** decides to take **@ent02** on a date to get **@ent02**'s mind off having a baby.

(c) Queries generated from the passages in (b).

ID	Passage
1.a	$\boldsymbol{x}$ announces that **@ent03** is going to be teaching a graduate class at the university.
1.b	**@ent03** announces that $\boldsymbol{x}$ is going to be teaching a graduate class at the university.
2.a	$\boldsymbol{x}$ misses dressing up for romantic dates so **@ent01** promises to take **@ent02** out.
2.b	**@ent02** misses dressing up for romantic dates so $\boldsymbol{x}$ promises to take **@ent02** out.
2.c	**@ent02** misses dressing up for romantic dates so **@ent01** promises to take $\boldsymbol{x}$ out.
...	

Table 1: An example dialog and its passages and queries from our corpus. All mentions are encoded by their entity IDs. The queries are generated by replacing each unique entity in every passage with the variable x (Section 3.2). @ent01: Joey, @ent02: Rachel, @ent03: Ross, @ent04: Neuman, @ent05: Paul.

is then queried to Elasticsearch that has indexed the selected scenes, and the scene with the highest relevance is retrieved. Finally, the retrieved scene along with the queried sentence are sent to a crowd worker who is asked to determine whether or not they are relevant, and perform anaphora resolution to replace all pronouns in the sentence with the corresponding character names. The sentence that is checked for the relevancy and processed by the anaphora resolution is considered a passage.

Out of 6,014 sentences collected from the plot summaries, 2,994 of them got turned into passages; in other words, about a half of the sentences could not be paired with relevant scenes by Elasticsearch. In addition to these pseudo-generated passages, two more sets of passages are created. For the first set, crowd workers are asked to generate new passages including factual descriptions different from the ones that are pseudo-generated. This produced additional 615 passages; however, passages in this set could be biased toward the dominant characters. To increase the diversity of the character entities in the passages, crowd workers are asked to generate the second set of passages that include factual descriptions related to only non-dominant characters. A total of 1,037 passages are generated in this set, which makes passage completion even more challenging since the chance of the dominant characters being the answers becomes much lower with this second set. Figure 1 shows the overview of passage generation. Note that Amazon Mechanical Turk is used for all crowdsourcing.

3.2 Mention Annotation

For all dialogs and their passages, mentions are first detected automatically by the named entity recognizer in NLP4J (Choi, 2016) using the PERSON entity, then manually corrected. For each passage including multiple mentions, a query is created for every mention by replacing it with the variable x:

Rachel misses dating, so *Joey* offers to take *Rachel* out.
⇒ $\boldsymbol{x}$ misses dating, so *Joey* offers to take *Rachel* out.
⇒ *Rachel* misses dating, so $\boldsymbol{x}$ offers to take *Rachel* out.
⇒ *Rachel* misses dating, so *Joey* offers to take $\boldsymbol{x}$ out.

Following Hermann et al. (2015), all mentions implying the same character are encoded by the same entity ID. A different set of entity IDs are randomly generated for each dialog; for the above example, *Joey* and *Rachel* may be encoded by **@ent01** and **@ent02** in this dialog (Table 1), although they can be encoded by different entity IDs in other dialogs. This random encoding prevents learning models from overfitting to certain types of entities. On the other hand, the same set of entity IDs are applied to the passages associated with the dialog.

One issue still remains that characters in this dataset are often mentioned by several aliases (e.g., nicknames, honorifics) such that it is not trivial to cluster mentions implying the same character using simple string matching. Thus, an entity dictionary is created for each character whose key is the name of the character and the value is a list of aliases for the character, manually inspected throughout the entire show. This entity dictionary is then used to link mentions in both the dialogs and the passages to their character entities.

Type	Count
# of dialogs	1,681
# of passages	4,646
# of queries	13,487
Avg. # of utterances per dialog	15.8
Avg. # of tokens per dialog/passage	290.8 / 19.9
Avg. # of mentions per dialog/passage	24.4 / 3.0
Avg. # of entities per dialog/passage	5.4 / 2.2
Max # of mentions per dialog/passage	117 / 15
Max # of entities per dialog/passage	16 / 7

Table 2: The overall statistics of our corpus.

Table 2 shows the overall statistics of our corpus. It is relatively smaller than the other corpora (Section 2). However, it is the largest, if not the only, corpus for the evaluation of passage completion on multiparty dialog that still gives enough instances to develop meaningful models using deep learning.

4 Approach

This section presents our deep learning architecture that integrates rich feature extraction from convolutional neural networks (CNN) into robust sequence modeling in recurrent neural networks (RNN) (Section 4.1). The combination of CNN and RNN has been adapted by several NLP tasks such as text summarization (Cheng and Lapata, 2016), essay scoring (Dong et al., 2017), sentiment analysis (Wang et al., 2016), or even reading comprehension (Dhingra et al., 2017). Unlike previous works that feed a sequence of sentences encoded by CNN to RNN, a sequence of utterances is encoded by CNN in our model, where each utterance is spoken by a distinct speaker and contains one or more sentences that are coherent in topics. Our best model is optimized by both the utterance (Section 4.2) and the dialog (Section 4.3) level attentions, showing significant improvement over the pure CNN+RNN model.

4.1 CNN + LSTM

Each utterance comes with a speaker label encoded by the entity ID in our corpus (Table 1). This entity ID is treated as the first word of the utterance in our models. Before training, random embeddings are generated for all entity IDs and the variable x with the same dimension d as word embeddings. All utterances and queries are zero-padded to their maximum lengths m and n, respectively.

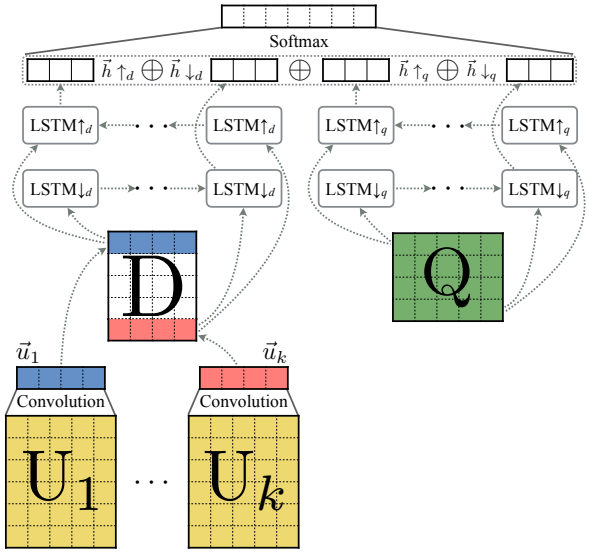

Figure 2: The overview of the CNN+LSTM model.

Given a query and a dialog comprising k-number of utterances, the query matrix $\mathrm{Q} \in \mathcal{R}^{n \times d}$ and the utterance matrix $\mathrm{U}_i \in \mathcal{R}^{m \times d}$ are created using the

word, entity, and variable embeddings $\forall i \in [1, k]$. For each U_i, 2D convolutions are performed for 2-5 grams, where each convolution takes f-number of filters and the output of every filter is max-pooled, resulting a vector of the size f. These vectors are concatenated to create the utterance embedding $\vec{u}_i \in \mathcal{R}^{1\times 4\cdot f}$, then the utterance embeddings are stacked to generate the dialog matrix $\mathrm{D} \in \mathcal{R}^{k\times 4\cdot f}$. This dialog matrix is fed into a bidirectional LSTM consisting of two networks, $\texttt{LSTM}{\downarrow_d}$ and $\texttt{LSTM}{\uparrow_d}$, that process the sequence of utterance embeddings in both directions. In parallel, Q is fed into another bidirectional LSTM with $\texttt{LSTM}{\downarrow_q}$ and $\texttt{LSTM}{\uparrow_q}$ that process the sequence of word embeddings in Q. Each LSTM returns two vectors from the last hidden states of $\texttt{LSTM}{\downarrow_*}$ and $\texttt{LSTM}{\uparrow_*}$:

$$\vec{h}\downarrow_d = \texttt{LSTM}\downarrow_d(\mathrm{D}) \quad \vec{h}\uparrow_d = \texttt{LSTM}\uparrow_d(\mathrm{D})$$
$$\vec{h}\downarrow_q = \texttt{LSTM}\downarrow_q(\mathrm{Q}) \quad \vec{h}\uparrow_q = \texttt{LSTM}\uparrow_q(\mathrm{Q})$$

All the outputs of LSTMs are concatenated and fed into the softmax layer that predicts the most likely entity for x in the query, where each dimension of the output layer represents a separate entity:

$$\mathrm{O} = \mathrm{softmax}(\vec{h}\downarrow_d \oplus \vec{h}\uparrow_d \oplus \vec{h}\downarrow_q \oplus \vec{h}\uparrow_q)$$
$$\mathrm{predict}(\mathrm{U}_1, \ldots, \mathrm{U}_k, \mathrm{Q}) = \mathrm{argmax}(\mathrm{O})$$

Figure 2 demonstrates our CNN+LSTM model that shows significant advantage over the pure bidirectional LSTM model as dialogs get longer.

4.2 Utterance-level Attention

Inspired by Yin et al. (2016), attention is applied to every word pair in the utterances and the query. First, the similarity matrix $\mathrm{S}_i \in \mathcal{R}^{m\times n}$ is created for each utterance matrix U_i by measuring the similarity score between every word in U_i and Q:

$$\mathrm{S}_i[r, c] = \mathrm{sim}(\mathrm{U}_i[r, :], \mathrm{Q}[c, :])$$
$$\mathrm{sim}(x, y) = 1/(1+\|x-y\|)$$

The similarity matrix is then multiplied by the attention matrix $\mathrm{A} \in \mathcal{R}^{n\times d}$ learned during the training. The output of this multiplication produces another utterance embedding $\mathrm{U}'_i \in \mathcal{R}^{m\times d}$, which is channeled to the original utterance embedding U_i and generates the 3D matrix $\mathrm{V}_i \in \mathcal{R}^{2\times m\times d}$ (Figure 3):

$$\mathrm{U}'_i = \mathrm{S}_i \cdot \mathrm{A}$$
$$\mathrm{V}_i = \mathrm{U}_i \oslash \mathrm{U}'_i$$

V_i is fed into the CNN in Section 4.1 instead of U_i and constructs the dialog matrix D.

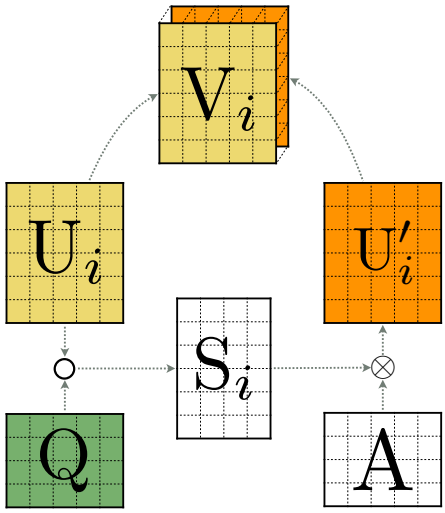

Figure 3: The overview of the utterance-level attention.

4.3 Dialog-level Attention

The utterance-level attention is for the optimization of local contents through word similarities between the query and the utterances. To give a global view to the model, dialog-level attention is applied to the query matrix Q and the dialog matrix D. First, 1D convolutions are applied to each row in Q and D, generating another query matrix $\mathrm{Q}' \in \mathcal{R}^{n\times e}$ and dialog matrix $\mathrm{D}' \in \mathcal{R}^{m\times e}$, where e is the number of filters used for the convolutions.

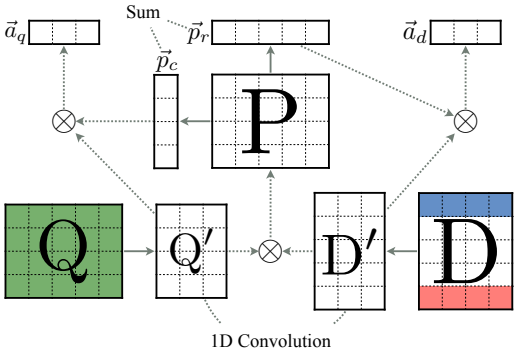

Figure 4: The overview of the dialog-level attention.

Q' is then multiplied to D'^T, resulting another similarity matrix $\mathrm{P} \in \mathcal{R}^{n\times m}$. Furthermore, the sum of each row in P is concatenated to create $\vec{p}_c \in \mathcal{R}^{n\times 1}$, and the sum of each column in P is also concatenated to create $\vec{p}_r \in \mathcal{R}^{1\times m}$:

$$\mathrm{P} = \mathrm{Q}' \cdot \mathrm{D}'^T$$
$$\vec{p}_c[r] = \sum_{j=1}^{m} P[r, j]$$
$$\vec{p}_r[c] = \sum_{j=1}^{n} P[j, c]$$

$\vec{p}_c^T$ is multiplied to Q' and $\vec{p}_r$ is multiplied to D', producing the attention embeddings $\vec{a}_q \in \mathcal{R}^{1\times e}$

Model	Development Set				Evaluation Set			
	Org.	25	50	100	Org.	25	50	100
Human Evaluation	-	-	-	-	**74.02**	-	-	-
Majority	28.61	27.65	21.57	19.79	30.08	28.23	21.58	17.59
Entity Centric	52.28	45.29	45.82	42.17	47.36	43.83	45.56	42.47
Bi-LSTM	72.24	68.90	64.51	55.17	71.21	67.37	62.95	53.76
CNN+LSTM	70.97	70.24	69.40	65.43	70.28	69.20	68.35	64.13
CNN+LSTM+UA	**72.42**	71.73	70.67	66.46	71.84	69.88	69.18	**66.99**
CNN+LSTM+DA	72.24	71.30	70.21	66.37	71.46	69.88	69.30	65.51
CNN+LSTM+UA+DA	72.21	**72.14**	**71.45**	**67.86**	**72.42**	**71.01**	**69.98**	**66.99**

Table 3: Results on the development and the evaluation sets from all models.

and $\vec{a}_d \in \mathcal{R}^{1\times e}$, respectively. Finally, these attention embeddings are concatenated with the outputs of the LSTMs in Section 4.1 then fed into the softmax layer to make the prediction:

$$\vec{a}_q = \vec{p}_c^{\,T} \cdot \mathrm{Q}'$$
$$\vec{a}_d = \vec{p}_r \cdot \mathrm{D}'$$
$$\mathrm{O} = \mathrm{softmax}(\vec{h}\downarrow_d \oplus \vec{h}\uparrow_d \oplus \vec{h}\downarrow_q \oplus \vec{h}\uparrow_q \oplus \vec{a}_d \oplus \vec{a}_q)$$
$$\mathrm{predict}(\mathrm{U}_1, \ldots, \mathrm{U}_k, \mathrm{Q}) = \mathrm{argmax}(\mathrm{O})$$

Similar attentions have been proposed by Yin et al. (2016) and evaluated on NLP tasks such as answer selection, paraphrase identification, and textual entailment; however, they have not been adapted to passage completion. It is worth mentioning that we have tried many other kinds of attention mechanisms and empirically found that the combination of these two attentions yields the best result for the passage completion task.

5 Experiments

The Glove 100-dimensional pre-trained word embeddings (Pennington et al., 2014) are used for all experiments ($d = 100$). The maximum lengths of utterances and queries are $m = 92$ and $n = 126$, and the maximum number of utterances is $k = 25$. For the 2/1D convolutions in Sections 4.1 and 4.3, $f = e = 50$ filters are used, and the `ReLu` activation is applied to all convolutional layers. The dimension of the LSTM outputs $\vec{h}\downarrow\uparrow_*$ is 32, and the `tanh` activation is applied to all hidden states of LSTMs. Finally, the `Adam` optimizer with the learning rate of 0.001 is used to learn the weights of all models. Table 4 shows the dataset split for our experiments that roughly gives 80/10/10% for training/development/evaluation sets.

	Train	Develop	Evaluate	Total
Queries	10,785	1,349	1,353	13,487

Table 4: Dataset split for our experiments, where each query is considered a separate instance.

5.1 Utterance Pruning

Most utterances in our corpus are relatively short except for a few ones so that padding all utterances to their maximum length is practically inefficient. Thus, pruning is used for those long utterances. For any utterance containing more than 80 words, that is about 1% of the entire dataset, stopwords are removed. If the utterance still has over 80 words, all words whose document frequencies are among the top 5% in the training set are removed. If the length is still greater than 80, all words whose document frequencies are among the top 30% in the training set are removed. By doing so, we reduce down the maximum length of utterances from 1,066 to 92, which dramatically speeds up the modeling without compromising the accuracy.

5.2 Datasets with Longer Dialogs

The average number of utterances per dialog is 15.8 in our corpus, which is relatively short. To demonstrate the model robustness for longer dialogs, three more datasets are created in which all dialogs have the fixed lengths of 25, 50, and 100 by borrowing utterances from their consecutive scenes. The same sets of queries are used although models need to search through much longer dialogs in order to answer the queries for these new datasets. The three pseudo-generated datasets as well as the original dataset are used for all our experiments.

5.3 Human Evaluation

Human performance is examined on the evaluation set of the original length using crowdsourcing. The workers are presented with passages and the corresponding dialogs and asked to choose the answer from the list of entities that appear in the dialog. For fair comparisons, the encoded input where the character names are replaced with the entity IDs are used for this evaluation as well, to minimize the bias from external knowledge.

5.4 Baselines

Three models are used to establish comprehensible baseline results:

Majority This model picks the dominant entity in the dialog as the answer for each query.

Entity Centric This is our reimplementation of Chen et al. (2016)'s entity centric model. Our entity centric model was evaluated on the CNN/Daily Mail dataset and showed a comparable result to the previous work.

Bi-LSTM This is the bidirectional LSTM model introduced by Chen et al. (2016), which outperforms their entity centric model by a large margin. We use their implementation of this model;[3] the input to this model is a list of words across all utterances within the dialog. All hyperparameters are tuned using the development set.

5.5 Results

Table 3 shows the results from all models. The human performance on the evaluation set is only 1.6+% higher than the best performing model, which on part shows the difficulty of the task. It should be noted that character anonymization process makes it harder to for people to find the answer. However, it also possible that some participants of the evaluation may enter the answer randomly (i.e the results may not truly reflect human performance). Notice that the performance of the majority model on our dataset is similar to the ones in the CNN/Daily Mail dataset, which validates the level of difficulty in our corpus. As expected, the entity centric model sets its performance in between the majority model and the other deep learning models. For all of our models and Bi-LSTM, experiments are run three times with different random seeds and the accuracies are averaged. The accuracy of Bi-LSTM reported on the CNN dataset is 72.4, which is similar to its performance on our dataset. Our models coupled with both the utterance-level and the dialog-level attentions (CNN+LSTM+UA+DA) outperform all the other models except for the one on the development set of the original dataset. Our models show significant advantage over Bi-LSTM as the length of the dialog gets larger.

Figure 5 shows the learning curves from Bi-LSTM and CNN+LSTM+UA+DA on the original dataset. The red circle and the black star mark the peaks of CNN+LSTM+UA+DA and Bi-LSTM, respectively. Although the accuracies between these models are very similar, our model converges in fewer epochs. Figure 6 shows the learning curves from both models in 3 trials on the length-100 dataset. Our models take fewer epochs to converge and the variance of performance across trials is smaller, implying that our models are not as sensitive to the hyperparameter tuning as Bi-LSTM.

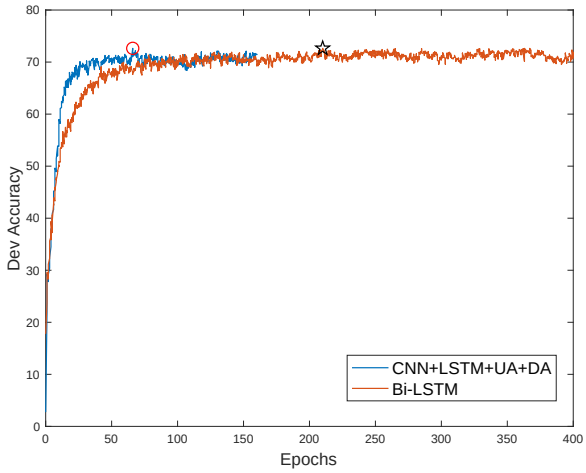

Figure 5: Training curves on the original dataset.

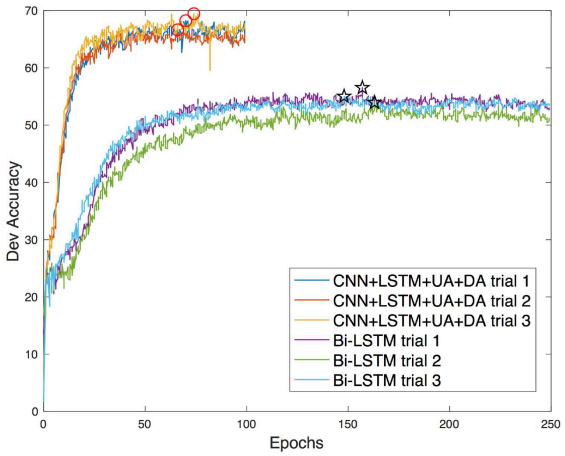

Figure 6: Training curves on the length-100 dataset.

6 Analysis

6.1 Attention Visualization

Figure 7 depicts the dialog-level attention matrix, that is P in Section 4.3, for the example in Table 1. The x-axis and y-axis denote utterances and words in the query, respectively. Each cell represents the attention value between a word in the query and an utterance. From this visualization, we see that query words such as *misses*, *take*, *good*, and *time*

[3] github.com/danqi/rc-cnn-dailymail

have the most attention from utterances as they are the keywords to find the answer entity. The utterances 14, 15 and 17 that give out the answer also get relatively high attention from the query words. This illustrates the effectiveness of the dialog-level attention in our model.

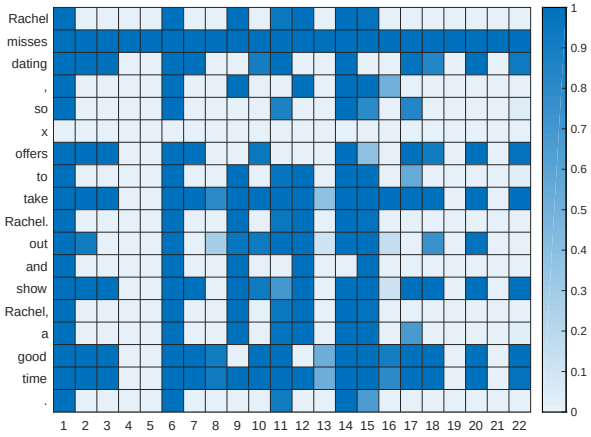

Figure 7: Visualization of the dialog-level attention matrix P for the example in Table 1.

6.2 Comparisons

Table 5 shows the confusion matrix between Bi-LSTM and CNN+LSTM +UA+DA on the original dataset. During the error analysis, it is noticed that Bi-LSTM is better at capturing exact string matches or paraphrases. As shown by the first two examples in Table 6, it is clear that those queries can be answered by capturing just the snippets of the dialogs. In the first example, "$\boldsymbol{x}$ makes up his mind about something" in the query matches "@ent06 sets his mind on something" in the dialog.

Model	Bi-LSTM: T	Bi-LSTM: F
C+L+U+D: T	850	133
C+L+U+D: F	118	252

Table 5: The confusion matrix between Bi-LSTM and CNN+LSTM+UA+DA.

In the second example, query phrase "the closet that $\boldsymbol{x}$ and @ent03 were in" also has the exact string match"the closet @ent18 and @ent03 were in" in the dialog. Although these cues are usually parts of sentences in long utterances, since Bi-LSTM is based on only words, it still is able to locate them correctly. On the other hand, our model encodes each utterance and then feeds encoded vectors to LSTMs, so the high level representation of the cues are mixed with other information, which hinders the model's ability to find the exact string matches. Our model is better at answering queries that require inference from multiple utterances. As shown by the last two examples in Table 6, the cues to the answers distribute across several utterances and there is no obvious match of words or phrases. In the third example, the model needs to infer that in the sentence "(She reaches over to look at the label on the box)", she refers to @ent18 and connect this information with the later utterance by @ent18 "This is addressed to Mrs. @ent16 downstairs" in order to answer the query. In the last example, finding the correct answer requires the model to interpret that the utterances "What the hell was that?!" and "(They both scream and jump away.)" reflect the outcome of *startles*, which is the verb in the query. As dialogs become longer in the padded datasets, because of the utterance encoding procedure, our model's ability's ability to locate relevant part of dialog is not influenced as much, whereas it becomes much more difficult for Bi-LSTM to find the matches.

6.3 Discussion

It is worth mentioning that besides the models presented in Section 4, the attention-over-attention reader was also experimented with our dataset, which outperformed various neural systems by a large margin on both the CNN news dataset and the Children Book Test dataset (Cui et al., 2017). We first reimplemented their model and experimented on the CNN dataset and achieved similar results as reported in the previous paper. We then experimented this model on our original length dataset. However, even after an extensive hyperparameter turning on the development set, this model did not achieve results comparable to those of either Bi-LSTM or our models, so we did not make a further analysis on this model.

7 Conclusion

We introduce a new corpus consisting of multiparty dialogs and crowdsourced annotation for the task of passage completion. To the best of our knowledge, this is the first corpus that can challenge deep learning models for passage completion on this genre. We also present a deep learning architecture combining convolutional and recurrent neural networks, coupled with utterance-level and dialog-level attentions. Models trained by our architecture significantly outperform the one trained by the pure bidirectional LSTM, especially on longer

Model	Query	Dialog
Bi-LSTM	@ent12 says that once ***x*** makes up his mind about something, @ent06 will have xxx with it.	Because you know as well as I do that once **@ent06** sets his mind on something, more often than not, he 's going to have sex with it.
Bi-LSTM	@ent06 points out that people are screwing in the closet that ***x*** and @ent03 were in.	Oh, by the way. Two people screwing in there (points to the closet **@ent18** and @ent03 were in) if you want to check that out.
CNN+LSTM +UA+DA	***x*** saw on the box that the cheesecake was addressed to Mrs. @ent16.	**@ent18** This is the best cheesecake I have ever had. Where did you get this? (She reaches over to look at the label on the box.) @ent10 It was at the front door. When I got home. Somebody sent it to us. **@ent18** @ent10, this is not addressed to you. This is addressed to Mrs. @ent16 downstairs. ...
CNN+LSTM +UA+DA	@ent17 startles @ent02 and ***x*** in the hallway to prove @ent17' point, which sets off an on-going competition of psuedo-attacks.	@ent17 DANGER !!! DANGER !!!!! @ent02 @ent17 !!! **@ent03** What the hell was that ?! (They both scream and jump away.)

Table 6: Examples for model comparison. The first column denotes the model that makes the correct prediction.

dialogs. Our analysis demonstrates the comprehension of our model using the attention matrix. For the future work, we will expand the annotation for more entity types and automatically link mentions with respect to their entities using an entity linker. All our resources including the annotated corpus and source codes of the models are available at: https://github.com/emorynlp/reading-comprehension.

References

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2358–2367. http://www.aclweb.org/anthology/P16-1223.

Henry Yu-Hsin Chen and Jinho D. Choi. 2016. Character Identification on Multiparty Conversation: Identifying Mentions of Characters in TV Shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. SIGDIAL'16, pages 90–100.

Henry Yu-Hsin Chen, Ethan Zhou, and Jinho D. Choi. 2017. Robust Coreference Resolution and Entity Linking on Dialogues: Character Identification on TV Show Transcripts. In *Proceedings of the 21st Conference on Computational Natural Language Learning*. CoNLL'17.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 484–494. http://www.aclweb.org/anthology/P16-1046.

Jinho D. Choi. 2016. Dynamic Feature Induction: The Last Gist to the State-of-the-Art. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. NAACL'16.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 593–602. http://aclweb.org/anthology/P17-1055.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1832–1846. http://aclweb.org/anthology/P17-1168.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 153–162. http://aclweb.org/anthology/K17-1017.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Annual Conference on Neural Information Processing Systems*. NIPS'15, pages 1693–1701.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks Principle: Reading

Children's Books with Explicit Memory Representations. In *Proceedings of the 6th International Conference on Learning Representations*. ICLR'16.

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)*. Association for Computational Linguistics, Vancouver, Canada, pages 1601–1611. http://aclweb.org/anthology/P17-1147.

Tomasz Jurczyk and J. D. Choi. 2017. Cross-domain Document Retrieval: Matching between Conversational and Formal Writings. In *Proceedings of the EMNLP Workshop on Building Linguistically Generalizable NLP Systems*. Copenhagen, Denmark, BLGNLP'17, pages 48–53. http://generalizablenlp.weebly.com.

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*. Association for Computational Linguistics, Copenhagen, Denmark, pages 785–794. https://www.aclweb.org/anthology/D17-1082.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2230–2235. https://aclweb.org/anthology/D16-1241.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1525–1534. http://www.aclweb.org/anthology/P16-1144.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

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*. Association for Computational Linguistics, Austin, Texas, pages 2383–2392. https://aclweb.org/anthology/D16-1264.

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*. EMNLP'13, pages 193–203.

Adam Trischler, Zheng Ye, Xingdi Yuan, Philip Bachman, Alessandro Sordoni, and Kaheer Suleman. 2016. Natural Language Comprehension with the EpiReader. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 128–137. https://aclweb.org/anthology/D16-1013.

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016. Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 225–230. http://anthology.aclweb.org/P16-2037.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv* 1502.05698.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics* 4:259–272. https://transacl.org/ojs/index.php/tacl/article/view/831.