Distribution Agreement

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

By

Kaixin Ma
Master of Science

Mathematics and Computer Science

____
Jinho D. Choi, Ph.D.
Advisor

____
Ken Mandelberg, Ph.D.
Committee Member

____
Connie Roth, Ph.D.
Committee Member

Accepted:

____
Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

____
Date

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

By

Kaixin Ma
B.S., Emory University, 2018

Advisor: Jinho D. Choi, Ph.D.

An abstract of
A thesis submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science
in Mathematics and Computer Science
2018

Abstract

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

By Kaixin Ma

This thesis expands a previously constructed corpus and presents 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. Previous researcher constructed a dataset by selecting transcripts from a TV show, generating passages for each dialog through crowdsourcing, and annotating mentions of characters in both the dialog and the passages. This work expands the previously constructed dataset following the same pipeline and fixes errors in the entire dataset. Given this dataset, a deep neural model is developed that integrates rich feature extraction from convolutional neural networks (CNN) into sequence modeling in recurrent neural networks (RNN), optimized by utterance and dialog level attentions. The 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. The analysis shows the effectiveness of the attention mechanisms and suggests a direction to machine comprehension on multiparty dialog.

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

By

Kaixin Ma
B.S., Emory University, 2018

Advisor: Advisor: Jinho D. Choi, Ph.D.

A thesis submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science
in Mathematics and Computer Science
2018

Acknowledgements

First I would like to thank my advisor, Jinho D. Choi. I met Jinho during my sophomore year when I basically know nothing about NLP. He invited me to join the lab and patiently taught me from the beginning piece by piece. His passion and insights about NLP motivate me to delve into research. I appreciate for numerous discussions we had that guided to me find the directions. I am grateful for his encouragement and support for me to participate in conference. I also thank him for days and nights we spent together to improve my papers.

I would like to thank my thesis committee member, Professor ken Mandelberg and Professor Connie Roth. Their comments and suggestions helped me improve the thesis.

Lastly, I would like to thank my friends and collaborators at Emory NLP lab: Hang Jiang, Tomasz Jurczyk, Gary Lai, Bonggun Shin, Catherine Xiao, Sayyed Zahiri and Ethan Zhou. I thank all of them for the insightful discussions we had. Their support helped me find solutions and make progress on research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

1 Introduction

Teaching machine to understand human language has been a long time goal for researchers. Numerous approaches, datasets and evaluation matrics have been developed to improve machine's comprehension ability. Reading comprehension that challenges machine's ability to understand a document through question answering has gained lots of interests recently. 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. [1] 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 the exploration of a corpus consisting of multiparty dialogs and the development of 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,

[1] https://medium.com/hijiffy/10-graphs-that-show-the-immense-power-of-messaging-apps-4a41385b24d6

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 thesis expands a previously constructed corpus for passage completion on multiparty dialog (Section 3), and presents a deep learning architecture that produces robust results for understanding dialog contexts (Section 4). The experiments show that models trained by this architecture significantly outperform the previous state-of-the-art model using bidirectional Long short term memory networks (LSTM), especially on longer dialogs (Section 5). The analysis highlights the comprehension of newly developed models for matching utterances in dialogs to words in passages (Section 6).

2 Related Work

This chapter provides a overview of several topics that are related to this thesis. There are a number of publicly available reading comprehension datasets and passage completion datasets. Unlike the other corpora where documents and passages are written in a similar writing style, they are multiparty dialogs and plot summaries in the corpus this thesis explored, 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.

2.1 Passage Completion

To address the lack of large scale supervised nature language passage completion data, Hermann et al. (2015) introduced the CNN/Daily Mail dataset where documents and passages were news articles and their summaries respectively. They also evaluated neural models with three types of readers on this dataset. Since its release, CNN/Daily Mail dataset has attract a lot of research interests and multiple systems have been developed and experimented on this dataset. Chen et al. (2016) proposed the entity centric model which incorporate traditional feature engineering and ranking algorithm to find the answer. They also built an end-to-end bidirectional LSTM model using attention and conducted a thorough analysis on this dataset. Trischler et al. (2016) presented the EpiReader that tried to mimic human's reasoning process while reading (i.e plug the possible answers into the question to see which makes the most sense). It contains an extractor that selects a set of candidates from documents and reasoner that formulate hypothesis for each candidates and pick the answer that fits the question best. The EpiReader used both CNN and RNN when encoding documents and questions. Dhingra et al. (2017) proposed the gated-attention reader that tried to mimic the rumination of human's reading process. (i.e goes back to re-read some parts of document to confirm) The gated-attention reader incorporated a multi-hop architecture and applied attention on multiplicative interactions between documents and passages. At last, Cui et al. (2017) introduced the attention-over-attention reader which is also based on bi-directional LSTM networks. In addition to the widely used passage-to-document attention, attention-over-attention reader also placed document-to-passage attention on top of that.

There are many other passage completion datasets that are in similar format as CNN/Daily Mail dataset. Hill et al. (2016) released the Children Book Test dataset (CBT) where children's book stories were used to constructed the dataset. The documents consist 20 consecutive sentences from the story and the 21st sentence is used as question in which one of the word is replaced by @placeholder. Paperno et al. (2016) introduced the LAnguage Modeling Broadened to Account for Discourse Aspects (LAMBADA) dataset to encourage development of models that are able to make inferences in broader contexts. LAMBADA dataset comprising novels from the Book corpus, is designed so that the answers are hard to find if only any single sentence is considered but easy if reading the whole document. Onishi et al. (2016) introduced the Who-did-What (WDW) dataset consisting of articles from the LDC English Gigaword newswire corpus. The questions and documents in WDW dataset come from two distinct articles about the same events, so it requires models to make stronger semantic analysis to answer the questions. All corpora described above provide queries, that are passages where certain words are masked by blanks, for the evaluation of passage completion.

2.2 Reading Comprehension

More datasets are available for other types of reading comprehension tasks, such multiple choice question answering and short phrase answer. Richardson et al. (2013) introducted MCTest, which consists stories and associated questions in a variety of topics created by crowd source workers. Stories in this dataset are relatively short and vocabulary used are easy as if they are for children in grade school. Joshi et al. (2017) constructed a challenging dataset TriviaQA containing question-answer-evidence triples. Questions were first collected from various trivia websites, then evidence text were gathered by web search and wikipedia of entities in the questions. Lai et al. (2017) released the ReAding Comprehension Dataset From Examinations dataset (RACE) , which consists real world reading comprehension test questions. The data were collected from English exam from Chinese middle school and high school. Thus the dataset was more well crafted and the questions require higher level of reasoning to answer. Rajpurkar et al. (2016) introduced the Stanford Question Answering Dataset (SQuAD), consisting questions generated by crowdsource workers on Wikipedia articles. The answers are constructed to be a span of text in the reading document. All corpora described above have document-query-answer triples in rather different format and different levels of difficulties. On some of these datasets, the state-of-the-art system's performance have come close to human performance whereas there are still large gaps on others.

2.3 Neural Architecture

Widely utilized for computer vision, CNN models have recently been applied to natural language processing and showed great results for many tasks such document classification Kim (2014), semantic parsing Shen et al. (2014) and question answering Yih et al. (2014). In some other tasks, CNN models are also utilized as feature extractors because of their ability to capture n-grams. RNN models, on the other hand, are originally designed for processing language. Because of RNN's nature, that each hidden states contains information from all previous hidden states, RNN models are thought to catch long distance dependencies in language, which are out of CNN models' reach. However, in practice when the sequences become long, RNN models' performances are not as good as expected due to the vanish gradients. Hochreiter and Schmidhuber (1997) introduced the LSTM networks, which intends to alleviate vanishing gradient of regular RNN. LSTM networks with attention have made remarkable breakthrough in many fields of NLP including machine translation Li et al. (2017); Wu et al. (2017), sentiment analysis Qian et al. (2017), and text summarization Nema et al. (2017); Tan et al. (2017). The combination of CNN and LSTM has also been explored, which take the advantage of CNN in feature extraction and RNN in sequence modeling. Yin et al. (2016) incorporated CNN-LSTM model to capture local character features and lexicon matches in name entity recognition task. Wang et al. (2016) proposed to use regional CNN to encode each sentence and use LSTM to integrate the information for dimensional sentiment analysis. Ma and Hovy (2016) introduced a neural system that consists CNN layers to encode character representation, LSTM layers to form context embedding and Conditional Random Field (CRF) layer in the last to perform sequence labeling. The hybrid of CNN and LSTM has produced promising results in many other tasks as well.

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.

Previous researchers generated passages for the first 8 seasons of the TV show *Friends* using plot summaries collected from fan site and annotated all mentions in the passages and dialogs through the crowdsource platform. To created more samples, more plot summaries for the last two seasons of *Friends* were collected from the same fan sites. Passages were generated for each dialog in last two seasons using the same pipeline as suggested by previous researchers. The details of the generation process is discussed in (Section 3.1), mentions annotation process is discussed in (Section 3.2). Lastly, errors in the entire dataset are fixed.

3.1 Passage Generation

An episode consists of multiple scenes, which may or may not be coherent. In this corpus, each scene is considered a separate dialog. The lengths of the

[1] `nlp.mathcs.emory.edu/character-mining`

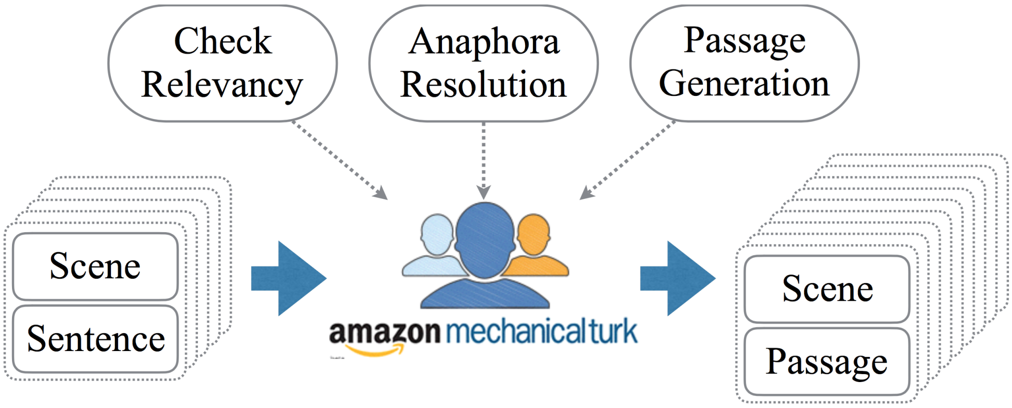

Figure 1: The overview of passage generation. Each episode is split into scenes, and each summary is segmented to sentences. Elasticsearch passes the scene-sentence pairs to crowd workers who are asked to check the relevancy, replace all pronouns with the corresponding names, and generate new passages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

scenes vary from 1 to 256 utterances; only scenes whose lengths are between 5 and 25 utterances are selected 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 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 con-

[2]https://github.com/emorynlp/nlp4j

sidered a passage. [3] Besides the plot summaries, crowd workers are asked to generated new passages which are descriptions about the dialog different from collected plot summaries. Passages created in this procedure, however, may be biased toward frequently appeared characters. To alleviate such issue, the second set of passages are generated. The lists of dominant characters for each dialog are created. Then crowd workers are asked to write descriptions about the dialog without using names on the dominant list. The passages generated in this procedure are even more challenging to answer. Figure 1 shows the overview of passage generation. Note that Amazon Mechanical Turk is used for all crowdsourcing.

The newly generated passages from last two seasons are merged with passages generated by previous researcher. In total, the corpus contains 4,648 passages, 2,994 of them come from plot summaries, 616 of them come from crowd workers generated descriptions and 1,038 of them are descriptions without dominant characters.

3.2 Mention Annotation

For all dialogs and their passages, mentions are first detected automatically by the named entity recognizer(NER) 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:

[3] `https://www.elastic.co`

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?

Table 1: An example dialog from *Friends*: Season 8, Episode 12, Scene 2. All mentions are encoded by their entity IDs. @ent01: Joey, @ent02: Rachel, @ent03: Ross, @ent04: Neuman, @ent05: Paul.

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*

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.

Table 2: Passages generated for the dialog in 1

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 3: Queries generated from passages in 2 The queries are generated by replacing each unique entity in every passage with the variable x (Section 3.2).

may be encoded by **@ent01** and **@ent02** in this dialog (Table 3), 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.

Two issues still remain in the dataset. One is that some entities in the passages and dialogs are not recognized by NER. As a result, some mentions of the same entity are encoded and some are not. The second issue is 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. For example, *Monica* can be called by her nickname *Mon*, honorific *Ms. Geller*, or full name *Monica Geller*. Having the same character encoded to different entity IDs can prevent the model from learning effectively. Thus a heuristic is designed to clean up the dataset. First for every dialog and its corresponding passage, an entity dictionary is created. All of tokens that appear in the entity dictionary but not picked by NER are converted to entities. Then, an entity mapping 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 mapping dictionary is then used to link mentions in both the dialogs and the passages to their character entities.

Type	**Count**
# of dialogs	1,682
# of passages	4,648
# 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 4: The overall statistics of the corpus.

Table 4 shows the overall statistics of the 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 Approaches

This section presents the deep learning architecture that is designed specifically for passage completion task on dialogs. This model 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 this model, where each utterance is spoken by a distinct speaker and contains one or more sentences that are coherent in topics. The 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.

This section also presents the entity centric classifier introduced by Chen et al. (2016) and the attention over attention (AoA) reader introduced by Cui et al. (2017). The entity centric classifier is a traditional linguistic approach, but it outperforms previous deep learning approach by a large margin on CNN/Daily Mail dataset. The AoA reader outperforms various neural systems by a large margin on both CNN news dataset and Children Book Test dataset. The author re-implemented these two models to serve as baselines.

4.1 CNN + LSTM

Each utterance comes with a speaker label encoded by the entity ID in the corpus (Table 3). This entity ID is treated as the first word of the utterance in CNN + LSTM 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.

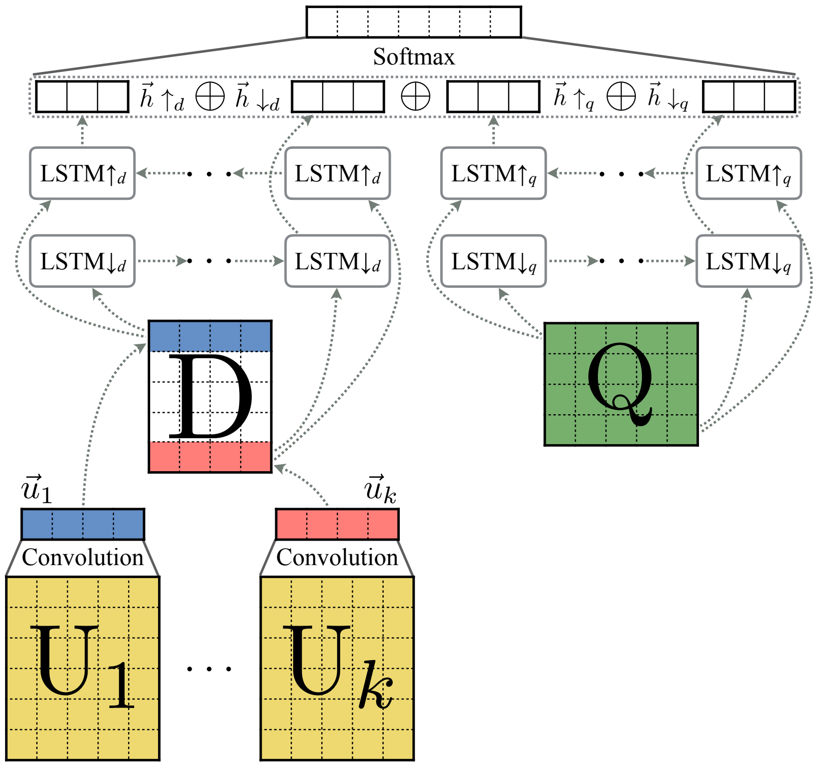

Figure 2: The overview of the CNN+LSTM model.

Given a query and a dialog comprising k-number of utterances, the query matrix $Q \in \mathcal{R}^{n \times d}$ and the utterance matrix $U_i \in \mathcal{R}^{m \times d}$ 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, $\mathtt{LSTM}\!\downarrow_d$ and $\mathtt{LSTM}\!\uparrow_d$, that process the sequence of utterance embeddings in both directions. In parallel, Q is fed into another bidirectional LSTM with $\mathtt{LSTM}\!\downarrow_q$ and $\mathtt{LSTM}\!\uparrow_q$ that process the sequence of word embeddings in Q. Each LSTM returns two vectors from the last hidden states of $\mathtt{LSTM}\!\downarrow_*$ and $\mathtt{LSTM}\!\uparrow_*$:

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

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.

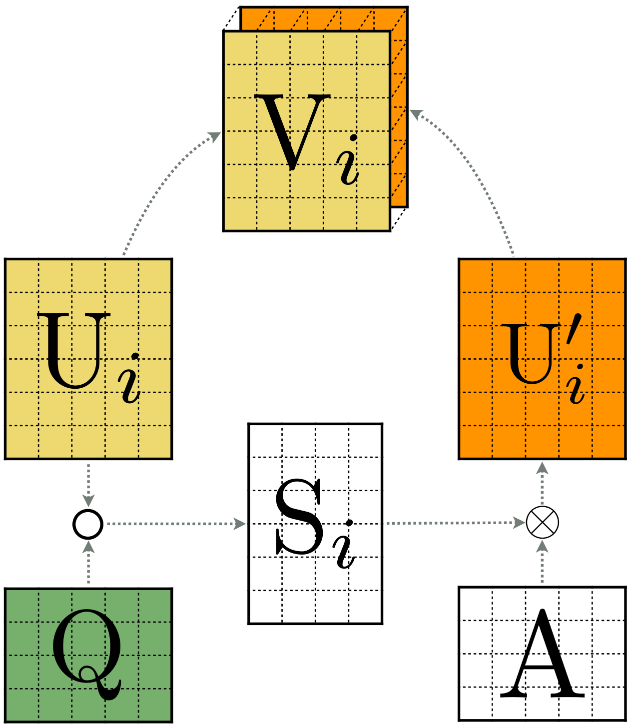

Figure 3: The overview of the utterance-level attention.

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.

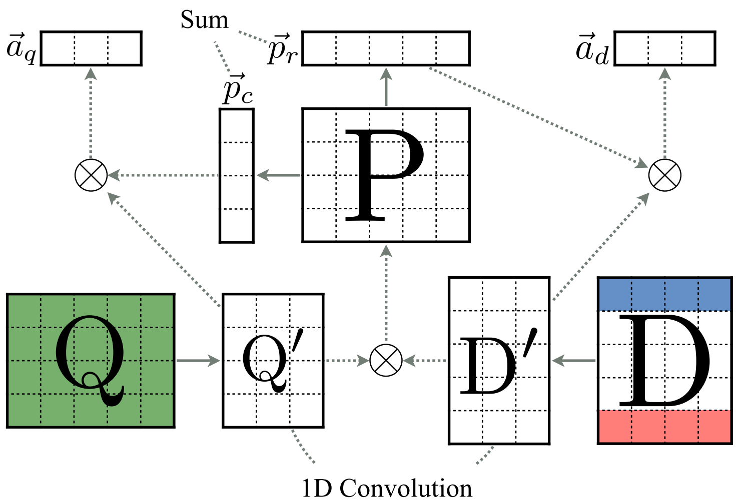

Figure 4: The overview of the dialog-level attention.

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 $\vec{p}_r \in \mathcal{R}^{1 \times m}$:

$$\mathrm{P} = \mathrm{Q}' \cdot \mathrm{D}'^T$$

$$\vec{p}_c[r] = \textstyle\sum_{j=1}^{m} P[r, j]$$

$$\vec{p}_r[c] = \textstyle\sum_{j=1}^{n} P[j, c]$$

$\vec{p}_c^T$ is multiplied to Q' and $\vec{p}_r$ is multiplied to D', producing the attention embeddings $\vec{a}_q \in \mathcal{R}^{1 \times e}$ 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 many other kinds of attention mechanisms have been tried and empirically the combination of these two attentions yields the best result for the passage completion task.

4.4 Entity Centric

This is the conventional feature based classifier from Chen et al. (2016). For each candidate entity in the document, a set of features is extracted. A ranking tool is used to rank each candidate's feature vector and the entity with the highest rank is chosen to be the answer. Since this is the replication of Chen et al. (2016)'s work, the same feature template is used and it is listed below.

- Whether the entity appear in the query
- Whether the entity appear in dialog
- The frequency of the entity in the dialog

- Whether there are exact matches of words surrounding the $\boldsymbol{x}$ and the entity. The combination of left and/or right one or two words are extracted as features.

- The entity is aligned with the $\boldsymbol{x}$ and the minimum distance for every non-stopping word in the question is calculated.

- Whether there is a verb or another entity that co-occur in the query and in some utterances in the dialog.

- Whether the entity share common parent or child with the $\boldsymbol{x}$ in the dependency parse tree.

Both queries and dialogs are first dependency parsed using NLP4J. Choi (2016) Then all features are extracted from the dependency parse trees. Following Chen et al. (2016) the implementation of LambdaMART Wu et al. (2010) in the Ranklib [1] package is used to rank the feature vectors.

4.5 Attention over Attention

This is the Attention over Attention (AoA) Reader introduced by Cui et al. (2017). Similarly, the speaker label is treated as the first word of the utterance. Then all of utterances in the dialog are concatenated into one long document. Given a query consisting m words and a dialog with n words, the query matrix Q $\in \mathcal{R}^{m\times d}$ and dialog matrix D $\in \mathcal{R}^{n\times d}$ are created through embedding layer where d is the embedding dimension. Then the dialog and query matrix are feed into two separate Bi-LSTM networks, which return sequence of hidden states. Thus D $\in \mathcal{R}^{n\times d}$ is encocoded to D$' \in \mathcal{R}^{n\times h}$ and Q $\in \mathcal{R}^{m\times d}$ is encoded to Q$' \in \mathcal{R}^{m\times h}$ where h is the hidden

[1] `https://sourceforge.net/p/lemur/wiki/RankLib/`

dimension size.

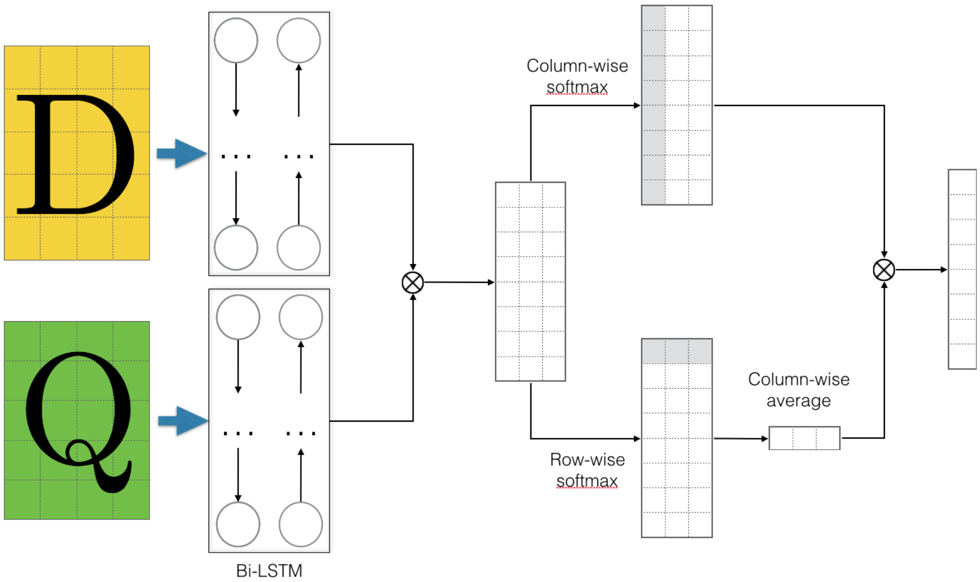

Figure 5: The overview of the AoA Reader.

Then attention matrix A $\in \mathcal{R}^{n\times m}$ is computed by taking the dot product of Q$' \in \mathcal{R}^{m\times h}$ and transpose of D$' \in \mathcal{R}^{n\times h}$. Each row in A denotes the attention of the document word to all question words and each column denotes attention of the question word to all document words. Column-wise softmax and row-wise softmax are perfromed on A $\in \mathcal{R}^{n\times m}$ seperately to get C $\in \mathcal{R}^{n\times m}$ and R $\in \mathcal{R}^{n\times m}$. By doing so, the attention values are normalized. Then the R $\in \mathcal{R}^{n\times m}$ is averaged along columns to get R$' \in \mathcal{R}^{1\times m}$. Since the operation is average, the normalization is maintained and R$' \in \mathcal{R}^{1\times m}$ can be seen as attention from whole document on each question words. Then R$' \in \mathcal{R}^{1\times m}$ is applied on C $\in \mathcal{R}^{n\times m}$ to determine the importance of each question word's attention, hence attention over attention. So the final attention vector $\alpha \in \mathcal{R}^{n\times 1}$ is calculated by taking the dot product of C $\in \mathcal{R}^{n\times m}$ and R$' \in \mathcal{R}^{1\times m}$. The overview of this model is shown in 5.

$$A = Q' \cdot D'^{T}$$

$$C = \text{softmax}(A)$$

$$R = \text{softmax}(A^{T})$$

$$R' = \text{average}(R)$$

$$\alpha = C \cdot R'^{T}$$

Finally, as suggested by Cui et al. (2017), the attention sum mechanism Kadlec et al. (2016) is applied to $R' \in \mathcal{R}^{1\times m}$ to make predictions. The probability of the word in the dialog being correct answer is given by summing up all attention values of this word.

$$\Pr(w \mid D, Q) = \textstyle\sum_{i\in I(w,D)} \alpha[i]$$

A minor modification of this model is also experimented. After computing the final attention vector $\alpha \in \mathcal{R}^{n\times 1}$, instead of summing up attention values for prediction. It is used to weight the document context embedding $D' \in \mathcal{R}^{n\times h}$ and compute final hidden vector $V \in \mathcal{R}^{1\times h}$. The prediction is made by taking softmax of the final hidden vector $V \in \mathcal{R}^{1\times h}$.

$$V = D'^{T} \cdot \alpha$$

$$\text{Prediction} = \text{Argmax}(\text{softmax}(V))$$

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 5 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 5: Dataset split for our experiments, where each query is considered a separate instance.

5.1 Utterance Pruning

Most utterances in the dataset 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

Model	Development Set			
	Org.	**25**	**50**	**100**
Majority	28.61	27.65	21.57	19.79
Word Distance	28.17	28.17	27.43	27.21
Entity Centric	52.28	45.29	45.82	42.17
AoA attention sum	61.25	-	-	-
AoA hidden vector	63.91	-	-	-
Bi-LSTM	72.24	68.90	64.51	55.17
CNN+LSTM	70.97	70.24	69.40	65.43
CNN+LSTM+UA	**72.42**	71.73	70.67	66.46
CNN+LSTM+DA	72.24	71.30	70.21	66.37
CNN+LSTM+UA+DA	72.21	**72.14**	**71.45**	**67.86**

Table 6: Results on the development set from all models.

doing so, the maximum length of utterances is reduced down 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 the 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 the experiments except the human evaluation and the AoA reader.

5.3 Human Evaluation

Human performance is examined on the test dataset of the original length using Amazon Mechanical Turk. Turkers are presented with passages and

Model	Evaluation Set			
	Org.	**25**	**50**	**100**
Human Evaluation	**74.02**	-	-	-
Majorit	30.08	28.23	21.58	17.59
Word Distance	28.08	26.24	25.06	25.94
Entity Centric	47.36	43.83	45.56	42.47
AoA attention sum	60.11	-	-	-
AoA hidden vector	61.07	-	-	-
Bi-LSTM	71.21	67.37	62.95	53.76
CNN+LSTM	70.28	69.20	68.35	64.13
CNN+LSTM+UA	71.84	69.88	69.18	**66.99**
CNN+LSTM+DA	71.46	69.88	69.30	65.51
CNN+LSTM+UA+DA	**72.42**	**71.01**	**69.98**	**66.99**

Table 7: Results on the evaluation set from all models.

corresponding dialogs and they are asked to choose the answer from the list of entities that appear in the dialog. To make fair comparison, the same inputs for models are used in this case. In other words, characters in dialogs and passages are replaced with entity IDs so that workers couldn't rely on the help of external knowledge. Workers are paid at the rate of 6$ per hour. Each hit is designed to take 1 minute to 2 minutes depending on the length of the dialog. The working times of workers are checked and found to be reasonable.

5.4 Baselines

Four models are used to establish comprehensible baseline results:

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

Word Distance Every entity is aligned with the variable x and calculate the minimum distance for every non-stopping word in the question. The entity with average minimum distance is chosen to be the answer.

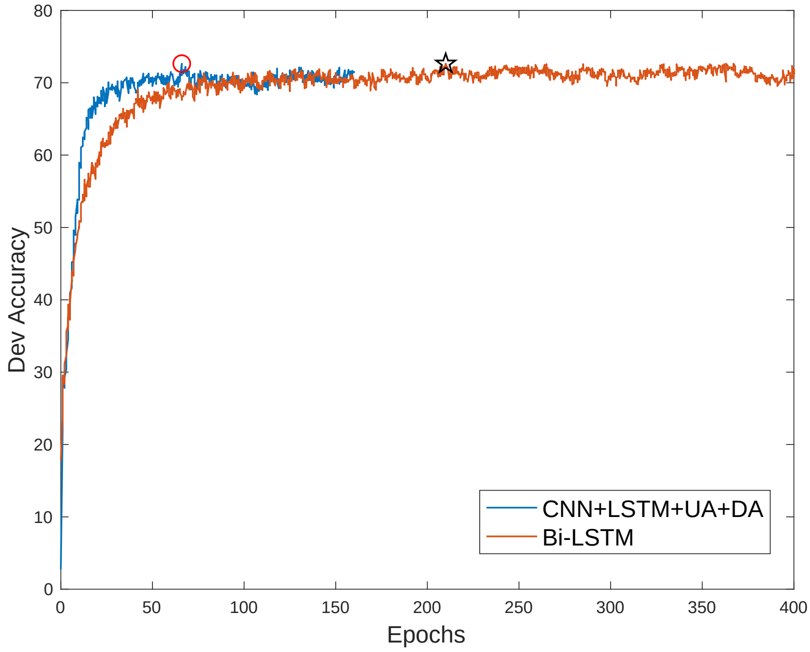

Figure 6: Training curves on the original dataset.

Entity Centric This is the reimplementation of Chen et al. (2016)'s entity centric model. This implementation 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. Chen et al. (2016)'s implementation of this model is used for experiments;[1] 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 Attention-over-Attention

This is the reimplementation of Cui et al. (2017)'s AoA reader. This implementation is first experimented on the CNN dataset and achieved similar

[1] `github.com/danqi/rc-cnn-dailymail`

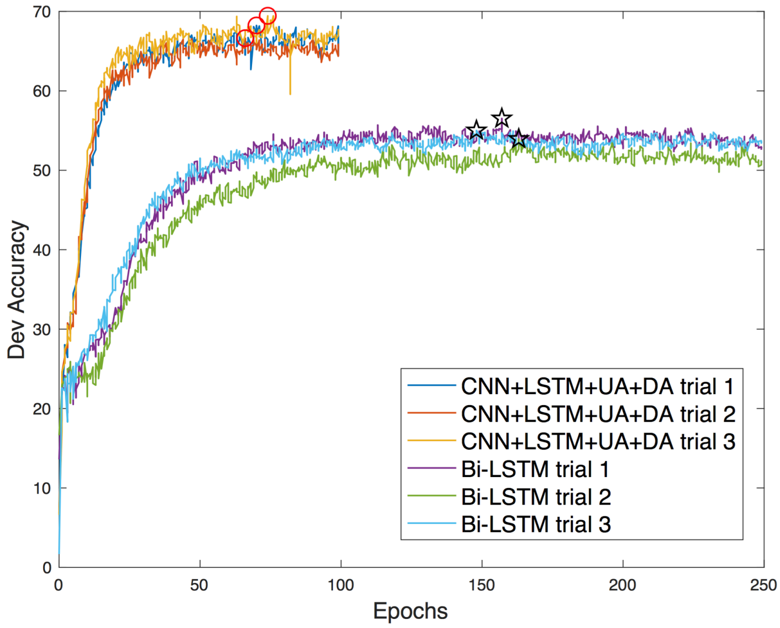

Figure 7: Training curves on the length-100 dataset.

results as reported in their paper. This model is then experiemnted on the original length dataset. However, even after hyperparameter turning on development set, this model couldn't achieve results close to those of either Bi-LSTM or CNN + LSTM models, so further experiments on longer dialogs are not performed.

5.6 Results

Table 6 shows the results from all models on the development set and table 7 shows the results from all models on the test set. 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 this dataset is similar to the ones in the CNN/Daily Mail dataset, which validates the level of difficulty the newly created corpus. When the dialogs get longer, it is expected that majority model's accuracy would drop. The word distance model's performance is consistent across datasets of different lengths. When of dialogs is relatively short, it is on par with majority model, whereas it has significant advantage on longer dialogs. As expected, the entity centric model sets its performance in between the majority model and other deep learning models. For all of CNN + LSTM 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 this dataset. CNN+LSTM model coupled with both the utterance-level and the dialog-level attentions outperform all the other models except for the one on the development set of the original dataset. The purposed neural architectures show significant advantage over Bi-LSTM as the length of the dialog gets larger.

Figure 6 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, CNN+LSTM+UA+DA converges in fewer epochs. Figure 7 shows the learning curves from both models in 3 trials on the length-100 dataset. CNN+LSTM+UA+DA again takes fewer epochs to converge and the variance of performance across trials is smaller, implying that it is not as sensitive to the hyperparameter tuning as Bi-LSTM.

6 Analysis

6.1 Attention Visualization

Figure 8 depicts the dialog-level attention matrix, that is P in Section 4.3, for the example in Table 3. 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, query words such as *misses*, *take*, *good*, and *time* 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 CNN+LSTM+UA+UD model.

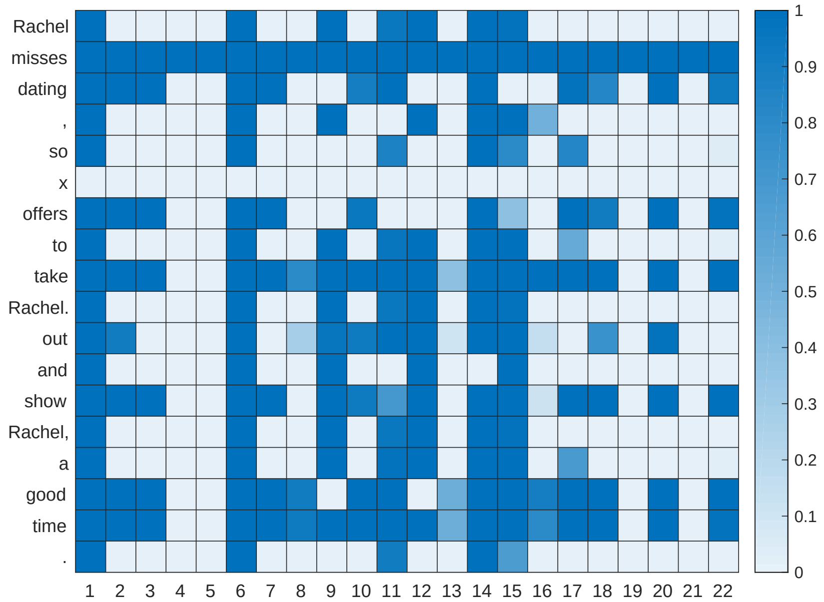

Figure 8: Visualization of the dialog-level attention matrix P for the example in Table 3.

6.2 Comparisons

Table 8 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 9, 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. 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, CNN+LSTM +UA+DA 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.

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

Table 8: The confusion matrix between Bi-LSTM and CNN+ LSTM+UA+DA.

CNN+LSTM +UA+DA is better at answering queries that require inference from multiple utterances. As shown by the last two examples in Table 9, 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, CNN+LSTM +UA+DA'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.

Model	Query	Dialog
Bi-LSTM	@ent12 says that once $\boldsymbol{x}$ 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 $\boldsymbol{x}$ 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	$\boldsymbol{x}$ 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 $\boldsymbol{x}$ 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 9: Examples for model comparison. The first column denotes the model that makes the correct prediction.

7 Conclusion

An existing corpus consisting of multiparty dialogs and crowdsourced annotation for the task of passage completion is expanded and thoroughly examined. A deep learning architecture combining convolutional and recurrent neural networks, coupled with utterance-level and dialog-level attentions is also presented. Models trained by this architecture significantly outperform the one trained by the pure bidirectional LSTM, especially on longer dialogs. Two other previously published models are re-implemented and experimented on this corpus. The analysis demonstrates the comprehension of the CNN+LSTM+UA+DA model using the attention matrix. The advantages of Bi-LSTM and CNN +LSTM+UA+DA are also analyzed with examples respectievely. For the future work, the annotation for more entity types may be extended and an entity linker may be explored to automatically link mentions with respect to their entities. Also, only one mention of entities in the query is replaced with blank currently. Multiple mentions of the same entity or mentions of different entity may be replaced with blanks in the query. Predicting all these blanks at one time could be a more challenging task and interesting to explore in the future.

BIBLIOGRAPHY

D. Chen, J. Bolton, and C. D. Manning. 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)*, pages 2358–2367, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1223`.

H. Y.-H. Chen and J. D. Choi. 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, 2016.

H. Y.-H. Chen, E. Zhou, and J. D. Choi. 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, 2017.

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

J. D. Choi. 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, 2016.

Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu. 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)*, pages 593–602, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1055`.

B. Dhingra, H. Liu, Z. Yang, W. Cohen, and R. Salakhutdinov. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1168`.

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

K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching Machines to Read and Comprehend. In *Annual Conference on Neural Information Processing Systems*, NIPS'15, pages 1693–1701, 2015.

F. Hill, A. Bordes, S. Chopra, and J. Weston. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. In *Proceedings of the 6th International Conference on Learning Representations*, ICLR'16, 2016.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. 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, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-1147.

T. Jurczyk and J. D. Choi. Cross-domain Document Retrieval: Matching between Conversational and Formal Writings. In *Proceedings of the EMNLP Workshop on Building Linguistically Generalizable NLP Systems*, BLGNLP'17, pages 48–53, Copenhagen, Denmark, 2017. URL http://generalizablenlp.weebly.com.

R. Kadlec, M. Schmid, O. Bajgar, and J. Kleindienst. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 908–918, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1086.

Y. Kim. Convolutional Neural Networks for Sentence Classification. *In Proceedings of EMNLP*, 2014.

G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1082.

J. Li, D. Xiong, Z. Tu, M. Zhu, M. Zhang, and G. Zhou. Modeling source syntax for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 688–697, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-1064.

X. Ma and E. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1101.

P. Nema, M. M. Khapra, A. Laha, and B. Ravindran. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-1098.

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

D. Paperno, G. Kruszewski, A. Lazaridou, N. Q. Pham, R. Bernardi, S. Pezzelle, M. Baroni, G. Boleda, and R. Fernandez. 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)*, pages 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1144.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Process-*

ing (EMNLP), pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Q. Qian, M. Huang, J. Lei, and X. Zhu. Linguistically regularized lstm for sentiment classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1679–1689, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1154`.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 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, Austin, Texas, November 2016. Association for Computational Linguistics. URL `https://aclweb.org/anthology/D16-1264`.

M. Richardson, C. J. Burges, and E. Renshaw. 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, 2013.

Y. Shen, X. He, L. D. J. Gao, and G. Mesnil. Learning Semantic Representations Using Convolutional Neural Networks for Web Search. *In Proceedings of WWW*, 2014.

J. Tan, X. Wan, and J. Xiao. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1108`.

A. Trischler, Z. Ye, X. Yuan, P. Bachman, A. Sordoni, and K. Suleman. Natural Language Comprehension with the EpiReader. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 128–137, Austin, Texas, November 2016. Association for Computational Linguistics. URL `https://aclweb.org/anthology/D16-1013`.

J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang. 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)*, pages 225–230, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://anthology.aclweb.org/P16-2037`.

J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv*, 1502.05698, 2015.

Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Adapting bboosting for information retrieval measures. *Information Retrieval*, 13:254–270, June 2010. URL `https://www.microsoft.com/en-us/research/publication/adapting-boosting-for-information-retrieval-measures/`.

S. Wu, D. Zhang, N. Yang, M. Li, and M. Zhou. Sequence-to-dependency neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 698–707, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1065`.

W. Yih, X. He, and C. Meek. Semantic Parsing for Single-Relation Question Answering. *In Proceedings of ACL 2014.*, 2014.

W. Yin, H. Schütze, B. Xiang, and B. Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272, 2016. ISSN 2307-387X. URL `https://transacl.org/ojs/index.php/tacl/article/view/831`.