DISTRIBUTION AGREEMENT

Changmao Li

NOTICE TO BORROWERS

Unpublished theses deposited in the Emory University Library must be used only in accordance with the stipulations prescribed by the author in the preceding statement.

The author of this thesis is :

Changmao Li
Department of Computer Science
Emory University
Atlanta, GA 30322

The director of this thesis is :

Jinho D. Choi, Ph.D.
Department of Computer Science
Emory University
Atlanta, GA 30322

Users of this thesis not regularly enrolled as students at Emory University are required to attest acceptance of the preceding stipulations by signing below. Libraries borrowing this thesis for the use of patrons are required to see that each user records here the information requested.

Name of user	Address	Date	Type of use (Examination only or copying)

Transformers to Learn Hierarchical Contexts in Multiparty Dialogue

By

Changmao Li
Master of Science

Advisor : Jinho D. Choi, Ph.D.
Department of Computer Science

Approved for the Department of Computer Science:

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

____
Committee Member: Shun Yan Cheung, Ph.D.

____
Committee Member: Michelangelo Grigni, Ph.D.

Accepted:

____
Dean of the Graduate School: Lisa A. Tedesco, Ph.D.

____
Date

Transformers to Learn Hierarchical Contexts in Multiparty Dialogue

by

Changmao Li
Master of Science

Advisor : Jinho D. Choi, Ph.D.

Abstract of
A Thesis submitted to the Faculty of the Graduate School
of Emory University in partial fulfillment
of the requirements of the degree of
Master of Science

Department of Computer Science

2020

Abstract

This thesis introduces a novel approach for transformers that learns hierarchical representations in multiparty dialogue. First, three language modeling tasks are used to pre-train the transformers, token- and utterance-level language modeling and utterance order prediction, that learn both token and utterance embeddings for better understanding in dialogue contexts. Then, multi-task learning between the utterance prediction and the token span prediction is applied to fine-tune for span-based question answering (QA). Our approach is evaluated on the FriendsQA dataset and shows improvements of 3.7% and 1.4% over the two state-of-the-art transformer models, BERT and RoBERTa, respectively.

Transformers to Learn Hierarchical Contexts in Multiparty Dialogue

by

Changmao Li
Master of Science

Advisor : Jinho D. Choi, Ph.D.

A Thesis submitted to the Faculty of the Graduate School
of Emory University in partial fulfillment
of the requirements of the degree of
Master of Science

Department of Computer Science

2020

Acknowledgements

First I would like to thank my advisor, Jinho D. Choi. I met Jinho during my first master year when I only knew little 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 the 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 also like to thank my friends and collaborators at Emory NLP lab. I thank all of them for the insightful discussions we had. Their support helped me find solutions and make progress on the research. I also thank my committee members who helped me with the writing of the thesis. Thank you all and hope we all have a good future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Transformer-based contextualized embedding approaches such as BERT [6], XLM [5], XLNet [29], RoBERTa [13], and AlBERT [12] have re-established the state-of-the-art for practically all question answering (QA) tasks on not only general domain datasets such as SQuAD [20, 21], MS Marco [17], TriviaQA [10], NewsQA [26], or NarrativeQA [11], but also multi-turn question datasets such as SQA [8], QuAC [4], CoQA [22], or CQA [24]. However, for span-based QA where the evidence documents are in the form of multiparty dialogue, the performance is still poor even with the latest transformer models [23, 28] due to the challenges in representing utterances composed by heterogeneous speakers.

Several limitations can be expected for language models trained on general domains to process dialogue. First, most of these models are pre-trained on formal writing, which is notably different from colloquial writing in dialogue; thus, fine-tuning for the end tasks is often not sufficient enough to build robust dialogue models. Second, unlike sentences in a wiki or news article written by one author with a coherent topic, utterances in a dialogue are from multiple speakers who may talk about different topics in distinct manners such that they should not be represented by simply concatenating, but rather as sub-documents interconnected to one another.

This thesis presents a novel approach to the latest transformers that learns hierarchical embeddings for tokens and utterances for a better understanding in dialogue contexts. While fine-tuning for span-based QA, every utterance as well as the question are separatedly encoded and multi-head attentions and additional transformers are built on the token and utterance embeddings respectively to provide a more comprehensive view of the dialogue to the QA model. As a result, our model achieves a new state-of-the-art result on a span-based QA task where the evidence documents are multiparty dialogue.

The contributions of this thesis are:

- New pre-training tasks are introduced to improve the quality of both token-level and utterance-level embeddings generated by the transformers, that better suit to handle dialogue contexts.
- A new multi-task learning approach is proposed to fine-tune the language model for span-based QA that takes full advantage of the hierarchical em-

beddings created from the pre-training.

- Our approach significantly outperforms the previous state-of-the-art models using BERT and RoBERTa on a span-based QA task using dialogues as evidence documents.

To begin with, Chapter 2 will introduce current related QA tasks. Transformer based approaches will also be introduced. Any foundation work prior and related to this dataset will be discussed. Chapter 3 will give the details of the proposed approach. Chapter 4 will conduct experiments for the proposed approach and give out the results. Some other experimental settings are also included. Chapter 5 will give the analysis of the proposed approach based on the ablation studies and question types analysis. Also, an error analysis will be provided for future insights. Chapter 6 will conclude the work and propose possible future improvements. Appendix A includes the model results and analysis for other tasks in the character mining project.

Chapter 2

Background

2.1 Language Modeling and Word Embedding

Language modeling and word embeddings have been popular in NLP community since Google proposed word2vec [15] which introduced the word embedding vectors to represent words. Currently there are two kinds of word embedding methods including static word embedding and contextualized word embedding. Static word embedding includes two language modeling strategies including skip-gram which is used to predict the context word for a given target word, and CBOW(The Continuous Bag of Words) which takes the context of each word as the input and tries to predict the word corresponding to the context. Static word embeddings fail to capture polysemy. They could only leverage off the vector outputs from unsupervised models for downstream tasks not the unsupervised models themselves. They were mostly shallow models to begin with and were often discarded after training (e.g. word2vec[15], Glove [18] , FastText [1]). They generate the same embedding for the same word in different contexts. Contextualized words embeddings, however, aim at capturing word semantics in different contexts to address the issue of polysemous and the context-dependent nature of words. The output of contextualized word embedding training is the trained model and vectors(e.g. BERT [6], XLM [5], XLNet [29], RoBERTa [13], and AlBERT [12]).

2.2 Transformers and Muti-head Attention

The Transformer was first proposed by Vaswani et al. [27] to address machine translation task. It has a encoder and a decoder to translate one language into another language. In both the encoder and the decoder it has several layers of following structure: a muti-head attention layer and a feed-forward layer and batch normalization layers. The layers of this structure shows better modeling the language context which finally started to be used as the language model encoder by Radford [19] who proposed GPT and Devlin et al. [6] who proposed BERT.

The muti-head attention layer in the transformer has two perspectives. The first one is the Scaled Dot-Product Attention and the second one is the muti-head. The Scaled Dot-Product Attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The muti-head represents linearly project the queries, keys and values multiple times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions.

2.3 Related Question Answering Tasks

The community constantly proposes new QA datasets and tasks. There are many general domain datasets such as SQUAD [20, 21], MS MARCO [17], TRIVIAQA [10], NEWSQA [26], or NARRATIVEQA [11]. There are another types of dataset called multi-turn question datasets such as SQA [8], QUAC [4], COQA [22], or CQA [24]. They gives questions in conversational form but their evidence documents are still from formal writings such as news, wikipedia, stories, literatures, etc. The other types of QA dataset called multiple choices QA also in dialogue setting which is DREAM [23], which uses dialogue as evidence documents, but is designed for reading comprehension that requires slightly different mechanism.

2.4 Related Transformer Based Approach

Transformer-based contextualized embedding approaches such as BERT [6], XLM [5], XLNet [29], RoBERTa [13], and AlBERT [12] have re-established the state-of-the-art for practically all question answering (QA) tasks. All of them have two stages of training: pre-training and fine-tuning. For BERT, there are two combination pre-training tasks which are masked language modeling task, and next sentence prediction task. RoBERTa, an improvement of BERT, has only masked language modeling task but with more data and dynamic masked language modeling technique. For other approaches such as XLNET and AlBERT they all have improvement of the BERT. For example, AlBERT is a lite version of BERT it is not only cutting the parameters of BERT but also they change the next sentence prediction task into sentence order prediction task which can achieve better results for downstream tasks. In this thesis we only use BERT and RoBERTa as our baseline models.

2.5 Character Mining Project

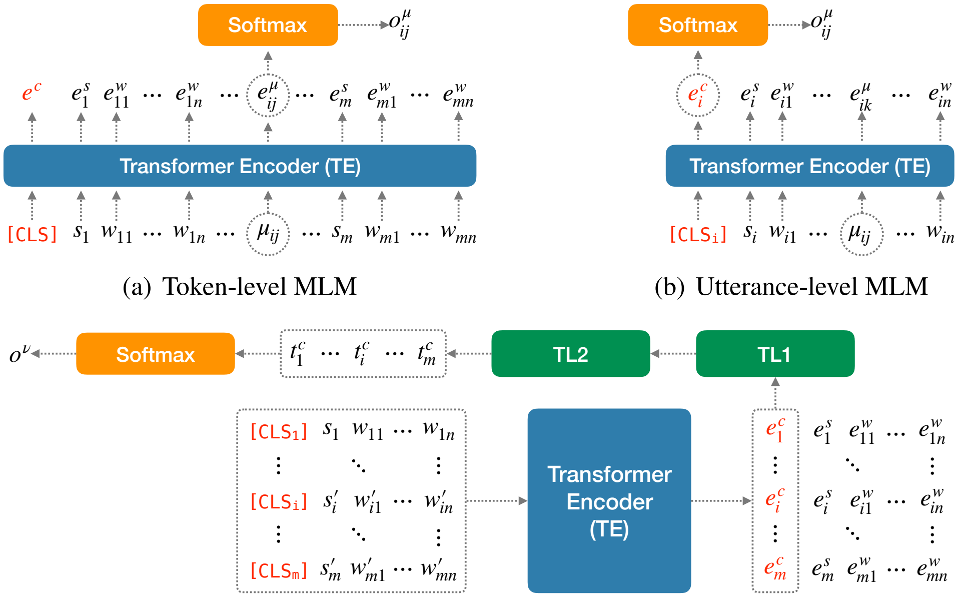
The Character Mining [1] dataset provides transcripts of the TV show Friends as well as annotation for several tasks. Future research could combine this project

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

with FriendsQA to generate more meaningful tasks and tools. The first two seasons are annotated [3] for character identification task, that is an entity linking task identifying personal mentions with character names. This annotation is extended to the next two seasons and ambiguous mentions are further annotated[2]. Building upon that, plural mentions of those four seasons [31] are also annotated for character identification tasks. Moreover, the first four seasons are annotated [30] for fine-grained emotion detection tasks. Further more, selected dialogues from all ten seasons are processed [14] for a cloze-style reading comprehension task. Besides, a personality detection task was annotated by Jiang et al. [9]. Finally a span based question answering task was annotated by Yang et al. [29]

Chapter 3

Approach


Softmax
o^{μ}_{ij}
e^c e^s_1 e^w_{11} ⋯ e^w_{1n} ⋯ e^{μ}_{ij} ⋯ e^s_m e^w_{m1} ⋯ e^w_{mn}
Transformer Encoder (TE)
[CLS] s_1 w_{11} ⋯ w_{1n} ⋯ μ_{ij} ⋯ s_m w_{m1} ⋯ w_{mn}
Softmax
o^{μ}_{ij}
e^c_i e^s_i e^w_{i1} ⋯ e^{μ}_{ik} ⋯ e^w_{in}
Transformer Encoder (TE)
[CLS$_i$] s_i w_{i1} ⋯ μ_{ij} ⋯ w_{in}
o^{ν}
Softmax
t^c_1 ⋯ t^c_i ⋯ t^c_m
TL2
TL1
[CLS$_1$] s_1 w_{11} ⋯ w_{1n}
[CLS$_i$] s'_i w'_{i1} ⋯ w'_{in}
[CLS$_m$] s'_m w'_{m1} ⋯ w'_{mn}
Transformer Encoder (TE)
e^c_1 e^s_1 e^w_{11} ⋯ e^w_{1n}
e^c_i e^s_i e^w_{i1} ⋯ e^w_{in}
e^c_m e^s_m e^w_{m1} ⋯ e^w_{mn}

(a) Token-level MLM

(b) Utterance-level MLM

(c) Utterance order prediction

Figure 3.1: The overview of our models for the three pre-training tasks (Section 3.2).

3.1 Transformers for Learning Dialogue

This section introduces a novel approach for pre-training (section 3.2) and fine-tuning (section 3.3) transformers to effectively learn dialogue contexts. Our approach has been evaluated with two kinds of transformers, BERT [6] and RoBERTa [13], and shown significant improvement to a question answering task (QA) on multiparty dialogue (Chapter 4).

3.2 Pre-training Language Models

Pre-training involves 3 tasks in sequence, the token-level masked language modeling (MLM; Section 3.2.1), the utterance-level MLM (Section 3.2.2), and the utterance order prediction (Section 3.2.3), where the trained weights from each task are transferred to the next task. Note that the weights of publicly available transformer encoders are adapted to train the token-level MLM, which allows our QA model to handle languages in both dialogues, used as evidence documents, and questions written in formal writing. Transformers from BERT and RoBERTa are trained with static and dynamic MLM respectively, as described by Devlin et al. [6], Liu et al. [13].

3.2.1 Token-level Masked LM

Figure 3.1(a) illustrates the token-level MLM model. Let $D = \{U_1, \dots, U_m\}$ be a dialogue where $U_i = \{s_i, w_{i1}, \dots, w_{in}\}$ is the i'th utterance in D, s_i is the speaker of U_i, and w_{ij} is the j'th token in U_i. All speakers and tokens in D are appended in order with the special token `CLS`, representing the entire dialogue, which creates the input string sequence $I = \{\texttt{CLS}\} \oplus U_1 \oplus \dots \oplus U_n$. For every $w_{ij} \in I$, let $I^{\mu}_{ij} = (I \setminus \{w_{ij}\}) \cup \{\mu_{ij}\}$, where μ_{ij} is the masked token substituted in place of w_{ij}. I^{μ}_{ij} is then fed into the transformer encoder (`TE`), which generates a sequence of embeddings $\{e^c\} \oplus E_1 \oplus \dots \oplus E_m$ where $E_i = \{e^s_i, e^w_{i1}, .., e^w_{in}\}$ is the embedding list for U_i, and $(e^c, e^s_i, e^w_{ij}, e^{\mu}_{ij})$ are the embeddings of $(\texttt{CLS}, s_i, w_{ij}, \mu_{ij})$ respectively. Finally, e^{μ}_{ij} is fed into a softmax layer that generates the output vector $o^{\mu}_{ij} \in \mathbb{R}^{|V|}$ to predict μ_{ij}, where V is the set of all vocabularies in the dataset.[1]

3.2.2 Utterance-level Masked LM

The token-level MLM (t-MLM) learns attentions among all tokens in D regardless of the utterance boundaries, allowing the model to compare every token to a broad context; however, it fails to catch unique aspects about individual utterances that can be important in dialogue. To learn an embedding for each utterance, the

[1] n: the maximum number of words in every utterance,
m: the maximum number of utterances in every dialogue.

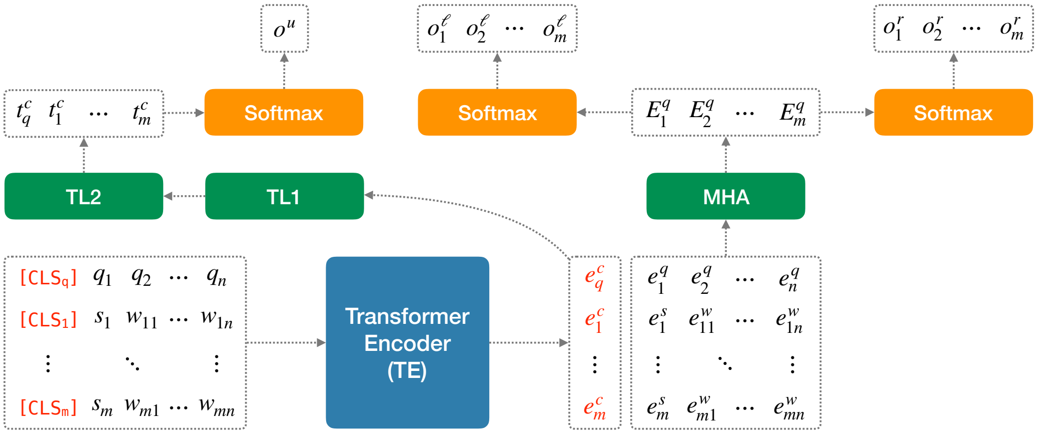

o^u
o^ℓ_1 o^ℓ_2 ⋯ o^ℓ_m
o^r_1 o^r_2 ⋯ o^r_m
t^c_q t^c_1 ⋯ t^c_m
Softmax
Softmax
E^q_1 E^q_2 ⋯ E^q_m
Softmax
TL2
TL1
MHA
[CLS_q] q_1 q_2 ⋯ q_n
[CLS_1] s_1 w_{11} ⋯ w_{1n}
[CLS_m] s_m w_{m1} ⋯ w_{mn}
Transformer Encoder (TE)
e^c_q e^c_1 ⋮ e^c_m
e^q_1 e^q_2 ⋯ e^q_n
e^s_1 e^w_{11} ⋯ e^w_{1n}
e^s_m e^w_{m1} ⋯ e^w_{mn}

Figure 3.2: The overview of our fine-tuning model exploiting multi-task learning (Section 3.3).

utterance-level MLM model is trained (Figure 3.1(b)). Utterance embeddings can be used independently and/or in sequence to match contexts in the question and the dialogue beyond the token-level, showing an advantage in finding utterances with the correct answer spans (section 3.3.1).

For every utterance U_i, the masked input sequence $I^{\mu}_{ij} = \{\texttt{CLS}_i\} \oplus \{(U_i \setminus \{w_{ij}\}) \cup \mu_{ij}\}$ is generated. Note that $\texttt{CLS}_i$ now represents U_i instead of D and I^{μ}_{ij} is much shorter than the one used for t-MLM. I^{μ}_{ij} is fed into $\texttt{TE}$, already trained by t-MLM, and the embedding sequence $E_i = \{e^c_i, e^s_i, e^w_{i1}, .., e^w_{in}\}$ is generated. Finally, e^c_i, instead of e^{μ}_{ij}, is fed into a softmax layer that generates o^{μ}_{ij} to predict μ_{ij}. The intuition behind the utterance-level MLM is that once e^c_i learns enough contents to accurately predict any token in U_i, it consists of most essential features about the utterance; thus, e_i^c can be used as the embedding of U_i.

3.2.3 Utterance Order Prediction

The embedding e_i^c from the utterance-level MLM (u-MLM) learns contents within U_i, but not across other utterances. In dialogue, it is often the case that a context is completed by multiple utterances; thus, learning attentions among the utterances is necessary. To create embeddings that contain cross-utterance features, the utterance order prediction model is trained (Figure 3.1(c)). Let $D = D_1 \oplus D_2$ where D_1 and D_2 comprise the first and the second halves of the utterances in D, respectively. Also, let $D' = D_1 \oplus D'_2$ where D'_2 contains the same set of utterances as D_2 although the ordering may be different. The task is to determine whether or not D' preserves the same order of utterances as D.

For each $U_i \in D'$, the input $I_i = \{\texttt{CLS}_i\} \oplus U_i$ is created and fed into `TE`, already trained by u-MLM, to create the embeddings $E_i = \{e_i^c, e_i^s, e_{i1}^w, .., e_{in}^w\}$. The sequence $E^c = \{e_1^c, \ldots, e_n^c\}$ is fed into two transformer layers, `TL1` and `TL2`, that generate the new utterance embedding list $T^c = \{t_1^c, \ldots, t_n^c\}$. Finally, T^c is fed into a softmax layer that generates $o^\nu \in \mathbb{R}^2$ to predict whether or not D' is in order.

3.3 Fine-tuning for QA on Dialogue

Fine-tuning exploits multi-task learning between the utterance ID prediction and the token span prediction, which allows the model to train both the utterance- and token-level attentions. The transformer encoder (`TE`) trained by the utterance order prediction (UOP) is used for both tasks. Given the question $Q = \{q_1, \ldots, q_n\}$ (q_i is the i'th token in Q) and the dialogue $D = \{U_1, \ldots, U_m\}$, Q and all U_* are fed into `TE` that generates $E_q = \{e^c_q, e^q_1, .., e^q_n\}$ and $E_i = \{e^c_i, e^s_i, e^w_{i1}, .., e^w_{in}\}$ for Q and every U_i, respectively.

3.3.1 Utterance ID Prediction

The utterance embedding list $E^c = \{e^c_q, e^c_1, .., e^c_n\}$ is fed into `TL1` and `TL2` from UOP that generate $T^c = \{t^c_q, t^c_1, .., t^c_n\}$. T^c is then fed into a softmax layer that generates $o^u \in \mathbb{R}^{m+1}$ to predict the ID of the utterance containing the answer span if exists; otherwise, the 0'th label is predicted, implying that the answer span for Q does not exist in D.

3.3.2 Token Span Prediction

For every E_i, the pair (E'_q, E'_i) is fed into the multi-head attention layer, `MHA`, where $E'_q = E_q \setminus \{e^c_q\}$ and $E'_i = E_i \setminus \{e^c_i\}$. `MHA` [27] then generates the attended embedding sequences, $T^a_1, \ldots, T^a_m$, where $T^a_i = \{t^s_i, t^w_{i1}, .., t^w_{in}\}$. Finally, each T^a_i is fed into two softmax layers, `SL` and `SR`, that generate $o^\ell_i \in \mathbb{R}^{n+1}$ and $o^r_i \in \mathbb{R}^{n+1}$ to predict the leftmost and the rightmost tokens in U_i respectively, that yield the answer span for Q. It is possible that the answer spans are predicted in multiple utterances, in which case, the span from the utterance that has the highest score for the utterance ID prediction is selected, which is more efficient than the typical dynamic programming approach.

Chapter 4

Experiments

4.1 Corpus

Despite of all great work in QA, only two datasets are publicly available for machine comprehension that take dialogues as evidence documents. One is DREAM comprising dialogues for language exams with multiple-choice questions [23]. The other is FRIENDSQA containing transcripts from the TV show *Friends* with annotation for span-based question answering [28]. Since DREAM is for a reading comprehension task that does not need to find the answer contents from the evidence documents, it is not suitable for our approach; thus, FRIENDSQA is chosen.

Each scene is treated as an independent dialogue in FRIENDSQA. Yang and Choi [28] randomly split the corpus to generate training, development, and evaluation sets such that scenes from the same episode can be distributed across those three sets, causing inflated accuracy scores. Thus, we re-split them by episodes to prevent such inflation. For fine-tuning (section 3.3), episodes from the first four seasons are used as described in Table 4.1. For pre-training (section 3.2), all transcripts from Seasons 5-10 are used as an additional training set.

Set	D	Q	A	E
Training	973	9,791	16,352	1 - 20
Development	113	1,189	2,065	21 - 22
Evaluation	136	1,172	1,920	23 - *

Table 4.1: New data split for FriendsQA. D/Q/A: # of dialogues/questions/answers, E: episode IDs.

4.2 Models

The weights from the $\text{BERT}_{\text{base}}$ and $\text{RoBERTa}_{\text{base}}$ models [6, 13] are transferred to all models in our experiments. Four baseline models, BERT, BERT_{pre}, RoBERTa, and $\text{RoBERTa}_{\text{pre}}$, are built, where all models are fine-tuned on the datasets in Table 4.1 and the $\star_{\text{pre}}$ models are pre-trained on the same datasets with the additional training set from Seasons 5-10 (section 4.1). The baseline models are compared to BERT_{our} and $\text{RoBERTA}_{\text{our}}$ that are trained by our approach.[1]

4.3 Results

Table 4.2 shows results achieved by all the models. Following Yang and Choi [28], exact matching (EM), span matching (SM), and utterance matching (UM)

[1] Detailed experimental setup are provided in Appendices.

are used as the evaluation metrics. Each model is developed three times and their average score as well as the standard deviation are reported. The performance of RoBERTa$\star$ is generally higher than BERT$\star$ although RoBERTa$_{\text{base}}$ is pre-trained with larger datasets including CC-NEWS [16], OPENWEBTEXT [7], and STORIES [25] than BERT$_{\text{base}}$ such that results from those two types of transformers cannot be directly compared.

Model	**EM**	**SM**	**UM**
BERT	43.3(±0.8)	59.3(±0.6)	70.2(±0.4)
BERT$_{\text{pre}}$	45.3(±0.3)	60.0(±0.5)	70.7(±0.6)
BERT$_{\text{our}}$	**46.6**(±0.9)	**63.0**(±0.7)	**73.0**(±0.5)
RoBERTa	52.6(±0.7)	68.2(±0.3)	80.9(±0.8)
RoBERTa$_{\text{pre}}$	52.8(±0.4)	68.6(±0.2)	81.8(±0.7)
RoBERTa$_{\text{our}}$	**53.4**(±0.5)	**69.6**(±0.3)	**82.7**(±0.5)

Table 4.2: Accuracy (±stdev) achieved by all models.

The $\star_{\text{pre}}$ models show marginal improvement over their base models, implying that pre-training the language models on FRIENDSQA with the original transformers does not make much impact on this QA task. The models using our approach perform noticeably better than the baseline models, showing 3.7% and 1.4% improvements on SM from BERT and RoBERTa, respectively.

Table 4.3 shows the results achieved by RoBERTa$_{\text{our}}$ w.r.t. question types. UM drops significantly for Why that often spans out to longer sequences and also

Type	**Dist.**	**EM**	**SM**	**UM**
`Where`	18.16	66.1(±0.5)	79.9(±0.7)	89.8(±0.7)
`When`	13.57	63.3(±1.3)	76.4(±0.6)	88.9(±1.2)
`What`	18.48	56.4(±1.7)	74.0(±0.5)	87.7(±2.1)
`Who`	18.82	55.9(±0.8)	66.0(±1.7)	79.9(±1.1)
`How`	15.32	43.2(±2.3)	63.2(±2.5)	79.4(±0.7)
`Why`	15.65	33.3(±2.0)	57.3(±0.8)	69.8(±1.8)

Table 4.3: Results from RoBERTa$_{\text{our}}$ by question types.

requires deeper inferences to answer correctly than the others. Compared to the baseline models, our models show more well-around performance regardless the question types.[2]

4.4 Other Experimental Details

The BERT$_{\text{base}}$ model and the RoBERTa$_{\text{BASE}}$ model use the same configuration. The two models both have 12 hidden transformer layers and 12 attention heads. The hidden size of the model is 768 and the intermediate size in the transformer layers is 3,072. The activation function in the transformer layers is `gelu`.

Pre-training The batch size of 32 sequences is used for pre-training. `Adam` with the learning rate of $5 \cdot 10^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, the `L2` weight decay of 0.01, the learning rate warm up over the first 10% steps, and the linear decay of the

[2] Question type results for all models are in Appendices.

learning rate are used. A dropout probability of 0.1 is applied to all layers. The cross-entropy is used for the training loss of each task. For the masked language modeling tasks, the model is trained until the perplexity stops decreasing on the development set. For the other pre-training tasks, the model is trained until both the loss and the accuracy stop decreasing on the development set.

Fine-tuning For fine-tuning, the batch size and the optimization approach are the same as the pre-training. The dropout probability is always kept at 0.1. The training loss is the sum of the cross-entropy of two fine-tuning tasks as in section 3.3.

Chapter 5

Analysis

5.1 Ablation Studies Analysis

To find out which step is important for the question answering task we did the ablation studies by experimenting with approaches step by step, Table is the results of the ablation studies of the experiment. In this ablation studies we add the speaker name into the vocabulary to enable the speaker representation can be learned better. From the results From the results, we found that the main improvement is based on joint learning of utterance id prediction which pre-training tasks help. We found following conclusions: 1. None of the language modelings in Section 2.1 made much impact without getting coupled with the joint learning of the utterance ID prediction (UID) in Section 2.2.1. This is indeed encouraging since all the proposed language modelings are designed to help UID, not the span prediction. 2. With only the token-level LM (2.1.1), UID still didn't make much impact. 3. With the utterance-level LM (2.1.2), UID gave over 1% boost on both SM and UM for BERT models and about 0.5% boost on both SM and UM for RoBERTa models. 4. With the utterance-order prediction (2.1.3), UID gave the rest of the boost, which was slightly higher than the ones achieved by the utterance-level LM.

Method	EM	SM	UM
BERTpre with uid_loss	45.7(±0.8)	61.1(±0.8)	71.5(±0.5)
BERTpre without uid_loss	45.6(±0.9)	61.2(±0.7)	71.3(±0.6)
BERTpre+ulm with uid_loss	46.2(±1.1)	62.4(±1.2)	72.5(±0.8)
BERTpre+ulm without uid_loss	45.7(±0.9)	61.8(±0.9)	71.8(±0.5)
BERTpre+ulm+uop with uid_loss	46.8(±1.3)	63.1(±1.1)	73.3(±0.7)
BERTpre+ulm+uop without uid_loss	45.6(±0.9)	61.7(±0.7)	71.7(±0.6)
RoBERTapre with uid_loss	52.8(±0.9)	68.7(±0.8)	81.9(±0.5)
RoBERTapre without uid_loss	52.6(±0.7)	68.6(±0.6)	81.7(±0.7)
RoBERTapre+ulm with uid_loss	53.2(±0.6)	69.2(±0.7)	82.4(±0.5)
RoBERTapre+ulm without uid_loss	52.9(±0.8)	68.7(±1.1)	81.7(±0.6)
RoBERTapre+ulm+uop with uid_loss	53.5(±0.7)	69.6(±0.8)	82.7(±0.5)
RoBERTapre+ulm+uop without uid_loss	52.5(±0.8)	68.8(±0.5)	81.9(±0.7)

Table 5.1: Results for the ablation studies:where BERTpre or RoBERTapre is the first task of the pretraining, ulm is utterance level language model task and uop is the utterance order prediction task. with uid_loss or without uid_loss represent if we use joint learning of two tasks when doing fine-tuning.

5.2 Question Type Analysis

Tables in this section show the results with respect to the question types using all models (section 4.2) in the order of performance. Our proposed approach can consistently enhance the performance on both BERT and RoBERTa models and all evaluation metrics. The table shows the our model performs better on what, how and why questions.

Type	Dist.	EM	SM	UM
Where	18.16	68.3(±1.3)	78.8(±1.2)	89.2(±1.5)
When	13.57	63.8(±1.6)	75.2(±0.9)	86.0(±1.6)
What	18.48	54.1(±0.8)	72.5(±1.5)	84.0(±0.9)
Who	18.82	56.0(±1.3)	66.1(±1.3)	79.4(±1.2)
How	15.32	38.1(±0.7)	59.2(±1.6)	77.5(±0.7)
Why	15.65	32.0(±1.1)	56.0(±1.7)	68.5(±0.8)

Table 5.2: Results from RoBERTa by question types.

Type	Dist.	EM	SM	UM
Where	18.16	67.1(±1.2)	78.9(±0.6)	89.0(±1.1)
When	13.57	62.3(±0.7)	76.3(±1.3)	88.7(±0.9)
What	18.48	55.1(±0.8)	73.1(±0.8)	86.7(±0.8)
Who	18.82	56.2(±1.4)	64.0(±1.7)	77.1(±1.3)
How	15.32	41.2(±1.1)	61.2(±1.5)	79.8(±0.7)
Why	15.65	32.4(±0.7)	57.4(±0.8)	69.1(±1.4)

Table 5.3: Results from $\texttt{RoBERTa}_{\text{pre}}$ by question types.

5.3 Error Analysis

From the above question type analysis we know that the main error can be found in three types of questions which are who, how and why questions, so we extract 100 specific error examples of those three question types to analyze the specific

Type	Dist.	EM	SM	UM
Where	18.16	66.1(±0.5)	79.9(±0.7)	89.8(±0.7)
When	13.57	63.3(±1.3)	76.4(±0.6)	88.9(±1.2)
What	18.48	56.4(±1.7)	74.0(±0.5)	87.7(±2.1)
Who	18.82	55.9(±0.8)	66.0(±1.7)	79.9(±1.1)
How	15.32	43.2(±2.3)	63.2(±2.5)	79.4(±0.7)
Why	15.65	33.3(±2.0)	57.3(±0.8)	69.8(±1.8)

Table 5.4: Results from $\texttt{RoBERTa}_{\text{our}}$ by question types.

Type	Dist.	EM	SM	UM
Where	18.16	57.3(±0.5)	70.2(±1.3)	79.4(±0.9)
When	13.57	56.1(±1.1)	69.7(±1.6)	78.6(±1.7)
What	18.48	45.0(±1.4)	64.4(±0.7)	77.0(±1.0)
Who	18.82	46.9(±1.1)	56.2(±1.4)	67.6(±1.4)
How	15.32	29.3(±0.8)	48.4(±1.2)	60.9(±0.7)
Why	15.65	23.4(±1.6)	46.1(±0.9)	56.4(±1.3)

Table 5.5: Results from BERT by question types.

Type	Dist.	EM	SM	UM
Where	18.16	62.8(±1.8)	72.3(±0.8)	82.1(±0.7)
When	13.57	60.7(±1.5)	70.7(±1.8)	80.4(±1.1)
What	18.48	43.2(±1.3)	64.3(±1.7)	75.6(±1.8)
Who	18.82	47.8(±1.1)	56.9(±1.9)	69.7(±0.7)
How	15.32	33.2(±1.3)	48.3(±0.6)	59.8(±1.1)
Why	15.65	22.9(±1.6)	46.6(±0.7)	54.9(±0.9)

Table 5.6: Results from $\texttt{BERT}_{\text{pre}}$ by question types.

Type	Dist.	EM	SM	UM
Where	18.16	63.3(±1.2)	72.9(±1.7)	77.0(±1.2)
When	13.57	48.4(±1.9)	66.5(±0.8)	79.5(±1.5)
What	18.48	52.1(±0.7)	69.2(±1.1)	81.3(±0.7)
Who	18.82	51.3(±1.1)	61.9(±0.9)	67.5(±0.9)
How	15.32	30.9(±0.9)	52.1(±0.7)	65.4(±1.1)
Why	15.65	29.2(±1.6)	53.2(±1.3)	65.7(±0.8)

Table 5.7: Results from $\texttt{BERT}_{\text{our}}$ by question types.

error to give some insights for future studies. Table 5.8 shows the errors types and the ratio of the error happened in these three lowest question types. The error types are based on Yang et al. [29]'s work which include entity resolution, paraphrase and partial match, cross-utterance reasoning, question bias, noise in annotation and miscellaneous. The table shows that the two main problem is the entity resolution and cross-utterance reasoning. The entity resolution error often happens when many of the same entities are mentioned in multiple utterances. This error also occurs when the QA system is asked about a specific person, but predicts wrong people where there are many people in multiple utterances. So how to correctly encode the speakers and mentions in the dialogue are one of the major challenges for future study. The cross-utterance reasoning error always happens in the why and how questions where the model always only do the pattern matching and predict the next utterance span of the matched pattern. So how to inference among utterances and summarize answers from them can also be one of the major challenges. Table 5.9, Table 5.10 and Table 5.11 show the examples of those errors.

Error types	Who	How	Why
Entity Resolution	**34%**	23%	20%
Paraphrase and Partial Match	14%	14%	13%
Cross-Utterance Reasoning	25%	**28%**	**27%**
Question Bias	11%	13%	17%
Noise in Annotation	4%	7%	9%
Miscellaneous	12%	15%	14%

Table 5.8: Error types ratio in three lowest question types.

Question	Why is Joey planning a big party ?
Context	...... Joey Tribbiani : Oh , we 're having a big party tomorrow night . Later ! Rachel Green : Whoa ! Hey - hey , you planning on inviting us ? Joey Tribbiani : Nooo , later . Phoebe Buffay : Hey !! Get your ass back here , Tribbiani !! Rachel Green : Hormones ! Monica Geller : What Phoebe meant to say was umm , how come you 're having a party and we 're not invited ? Joey Tribbiani : Oh , it 's Ross 's bachelor party . Monica Geller : Sooo ? ......
Gold Answer	it 's Ross 's bachelor party .
Prediction Answer	we 're having a big party tomorrow night

Table 5.9: Error example for why question

Question	Who opened the vent ?
Context	...... Ross Geller : Ok , got the vent open . Phoebe Buffay : Hi , I ’m Ben . I ’m hospital worker Ben . It ’s Ben ... to the rescue ! Ross Geller : Ben , you ready ? All right , gim me your foot . Ok , on three , Ben . One , two , three . Ok , That ’s it , Ben . note : (Ross and Susan lift Phoebe up into the vent .) Susan Bunch : What do you see ? Phoebe Buffay : Well , Susan , I see what appears to be a dark vent . Wait . Yes , it is in fact a dark vent . note : (A janitor opens the closet door from the outside .) ......
Gold Answer	Ross Geller
Prediction Answer	A janitor

Table 5.10: Error example for who question

Question	How does Joey try to convince the stripper to hang out with him ?
Context	...... Joey Tribbiani : Oh yeah - yeah . And I got the duck totally trained . Watch this . Stare at the wall . Hardly move . Be white . The Stripper : You are really good at that . So uh , I had fun tonight , you throw one hell of a party . Joey Tribbiani : Oh thanks . Thanks . It was great meetin ’ ya . And listen if any of my friends gets married , or have a birthday , or a Tuesday . . . The Stripper : Yeah , that would be great . So I guess umm , good night . Joey Tribbiani : Oh unless you uh , you wan na hang around . The Stripper : Yeah ? Joey Tribbiani : Yeah . I ’ll let you play with my duck . ...
Gold Answer	I ’ll let you play with my duck .
Prediction Answer	Oh unless you uh , you wan na hang around .

Table 5.11: Error example for how question

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

We present a novel transformer approach that effectively interprets hierarchical contexts in multiparty dialogue by learning utterance embeddings. Our approach is evaluated on a span-based QA task and outperforms two of the state-of-the-art transformer approaches, `BERT` and `RoBERTa`. We will further evaluate our approach on non-dialogue domains using other QA datasets to verify the generalizability.

6.2 Future Directions

There are two main challenges remains to be solved. The first one is that what is the correct approach to inference in the dialogue. A Dialogue has multiple utterances and they are always not related to each other, so how to recognize the logic flow of the dialogue can be a future research direction. The second one is that what is the correct way to encode speakers and mentions in the dialogue. A Dialogue has many speakers and in each utterance they may have difference types of mentions such as ”I”,”he” or certain nickname, so the other direction is that how to encoding these mentions to enable the model to know these are the same person and recognize the relation among them and the event happened to them. After solving these challenges we can build new version of the dialogue language model which includes improvement of all tasks not only span QA task.

Appendix A

Results for Other Character Mining Tasks

A.1 Friends Reading Comprehension task

A.1.1 Task Description

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

A.1.2 Results and Analysis

Table A.1 shows the results for friends Reading comprehension task. From the results we can draw two conclusions: 1. RoBERTa model is indeed better than BERT model. 2. Our approach does not help the improvement of the task. The main reason is that the speakers and mentions in the dialogue are replaced by label @entxxx which is not in the orignial BERT or RoBERTa training or our pre-training process.

Method	Accuracy
BERTpre	29.9(±0.8)
BERTpre+ulm	29.8(±0.9)
BERTpre+ulm+uop	29.9(±0.7)
RoBERTapre	31.2(±1.1)
RoBERTapre+ulm	31.2(±1.0)
RoBERTapre+ulm+uop	31.1(±0.9)

Table A.1: Results for Friends RC

A.2 Friends Emotion Detection task

A.2.1 Task Description

Emotion Detection aims to classify a fine-grained emotion for each utterance in multiparty dialogue.Each utterance is annotated with one of the seven emotions, sad, mad, scared, powerful, peaceful, joyful, and neutral, that are the primary emotions in the Feeling Wheel.

A.2.2 Results and Analysis

Table A.2 shows the results for friends emotion detection task. From the results we can draw two conclusions: 1. RoBERTa model is also indeed better than BERT model. 2. Our approach still does not help the improvement of the task. The main reason is that this task is mainly the simple utterance classification and the utterance level information does not count so much.

Method	Accuracy
BERTpre	33.4(±0.3)
BERTpre+ulm	33.2(±0.5)
BERTpre+ulm+uop	33.2(±0.5)
RoBERTapre	34.5(±0.8)
RoBERTapre+ulm	34.2(±0.9)
RoBERTapre+ulm+uop	34.2(±0.7)

Table A.2: Results for Friends ED

A.3 Friends Personality Detection task

A.3.1 Task Description

Multiparty Personality Recognition requires machines to determine the main speaker's personality from a short conversation in binary Big Five personality traits: Agreeableness (AGR): trustworthy, straightforward, generous vs. unreliable, complicated, meager, and boastful; Conscientiousness (CON): efficient and organized vs. sloppy and careless; Extroversion (EXT): outgoing, talkative, and energetic vs. reserved and solitary; Openness (OPN): inventive and curious vs. dogmatic and cautious; Neuroticism (NEU): sensitive and nervous vs. secure and confident.

A.3.2 Results and Analysis

Table A.3 shows the results for friends personality detection task. From the results we can draw two conclusions: 1. RoBERTa model is also indeed better than BERT model. 2. Our approach still does not help the improvement of the task. 3. Here the state of art results were achieved by pre-training the language model of the BERT and RoBERTa with friends corpus. The main reason that our approach failed is similar to the emotion detection task, which is that this task is also mainly the simple utterance classification and the utterance level information does not count so much.

Method	AGR	CON	EXT	OPN	NEU
BERTpre	58.2(±0.5)	57.7(±0.3)	59.2(±0.6)	61.2(±0.5)	59.3(±0.5)
BERTpre+ulm	58.1(±0.7)	57.5(±0.4)	59.1(±0.8)	61.2(±0.5)	59.3(±0.5)
BERTpre+ulm+uop	58.2(±0.5)	57.7(±0.6)	59.1(±0.5)	61.1(±0.5)	59.2(±0.5)
RoBERTapre	59.7(±0.7)	58.6(±0.5)	60.7(±0.7)	65.9(±0.6)	61.1(±0.5)
RoBERTapre+ulm	59.5(±0.5)	58.5(±0.8)	60.7(±0.8)	65.8(±0.9)	61.1(±0.5)
RoBERTapre+ulm+uop	59.6(±0.8)	58.6(±0.6)	60.6(±0.5)	65.8(±0.7)	61.1(±0.5)

Table A.3: Results for Friends PD

Bibliography

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. doi: 10.1162/tacl_a_00051. URL `https://www.aclweb.org/anthology/Q17-1010`.

[2] Henry Y. Chen, Ethan Zhou, and Jinho 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 2017)*, pages 216–225, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1023. URL `https://www.aclweb.org/anthology/K17-1023`.

[3] Yu-Hsin Chen and Jinho 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, pages 90–100, Los Angeles, September 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-3612. URL `https://www.aclweb.org/anthology/W16-3612`.

[4] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. doi: 10.18653/v1/d18-1241. URL `http://dx.doi.org/10.18653/v1/d18-1241`.

[5] Alexis CONNEAU and Guillaume Lample. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7057–7067. Curran Associates, Inc., 2019. URL `http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf`.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL'19, pages 4171–4186, 2019. URL `https://www.aclweb.org/anthology/N19-1423`.

[7] Aaron Gokaslan and Vanya Cohen. *OpenWebText Corpus*, 2019. URL `https://skylion007.github.io/OpenWebTextCorpus/`.

[8] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1167. URL `https://www.aclweb.org/anthology/P17-1167`.

[9] Hang Jiang, Xianzhe Zhang, and Jinho D. Choi. Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings, 2019.

[10] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Proceedings of the 55th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), 2017. doi: 10.18653/v1/p17-1147. URL `http://dx.doi.org/10.18653/v1/P17-1147`.

[11] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, Dec 2018. ISSN 2307-387X. doi: 10.1162/tacl_a_00023. URL `http://dx.doi.org/10.1162/tacl_a_00023`.

[12] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv*, 1907.11692, 2019. URL `http://arxiv.org/abs/1907.11692`.

[14] Kaixin Ma, Tomasz Jurczyk, and Jinho D. Choi. Challenging reading comprehension on daily conversation: Passage completion on multiparty dialog. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol-*

ume 1 (Long Papers), pages 2039–2048, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1185. URL `https://www.aclweb.org/anthology/N18-1185`.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL `http://arxiv.org/abs/1301.3781`.

[16] Sebastian Nagel. *News Dataset Available*, 2016. URL `https://commoncrawl.org/2016/10/news-dataset-available/`.

[17] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, 2016. URL `http://ceur-ws.org/Vol-1773/CoCoNIPS\_2016\_paper9.pdf`.

[18] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global

vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL `https://www.aclweb.org/anthology/D14-1162`.

[19] Alec Radford. Improving language understanding by generative pre-training. 2018.

[20] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016. doi: 10.18653/v1/d16-1264. URL `http://dx.doi.org/10.18653/v1/D16-1264`.

[21] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018. doi: 10.18653/v1/p18-2124. URL `http://dx.doi.org/10.18653/v1/P18-2124`.

[22] Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conver-

sational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, Mar 2019. ISSN 2307-387X. doi: 10.1162/tacl_a_00266. URL http://dx.doi.org/10.1162/tacl_a_00266.

[23] Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. DREAM: A Challenge Data Set and Models for Dialogue-Based Reading Comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231, 2019. URL https://www.aclweb.org/anthology/Q19-1014.

[24] Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. doi: 10.18653/v1/n18-1059. URL http://dx.doi.org/10.18653/v1/N18-1059.

[25] Trieu H. Trinh and Quoc V. Le. A Simple Method for Commonsense Reasoning. *arXiv*, 1806.02847, 2018. URL http://arxiv.org/abs/1806.02847.

[26] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni,

Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 2017. doi: 10.18653/v1/w17-2623. URL http://dx.doi.org/10.18653/v1/W17-2623.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, USA, 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4. URL http://dl.acm.org/citation.cfm?id=3295222.3295349.

[28] Zhengzhe Yang and Jinho D. Choi. FriendsQA: Open-domain question answering on TV show transcripts. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 188–197, Stockholm, Sweden, September 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-5923.

[29] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle,

A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5754–5764. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-
pdf.

[30] Sayyed Zahiri and Jinho Choi. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. 2018. URL https://aaai.org/ocs/index.php/WS/AAAIW18/paper/view/16434.

[31] Ethan Zhou and Jinho D. Choi. They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 24–34, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1003.