

國立臺灣大學電機資訊學院資訊工程學研究所

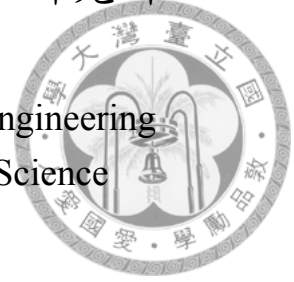
碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis



準確且穩固的問答模型

Toward Accurate and Robust Question Answering Systems

葉奕廷

Yi-Ting Yeh

指導教授：陳縉儂博士

Advisor: Yun-Nung Chen, Ph.D.

中華民國 109 年 1 月

January, 2020





誌謝

首先我想要感謝的是陳縉儂老師兩年來在研究上的悉心指導，老師不管何時都給予學生毫無保留的支持，是我們最強力的後盾。再來，感謝實驗室的夥伴們，上育、尚錡、大中、志文、健嘉、婷雲、子騰、兆緯、廷睿、浩同、佑安、逸軒，和你們相處的這兩年，我發現了自己的不足，更從每個人身上學到了很多，無論到哪裡，我們都是攜手並肩、共同前進。我還想感謝我的家人，有你們從小到大的栽培與陪伴，才有現在的我。

「在滿地都是六便士的街上，他抬起頭看到了月光。」

感謝所有在碩士期間幫助過我的人們，所有人一路上的支持，是我完成碩士學位的最大動力。





摘要

本篇論文主要目的在於解決問答模型的問題，因為問答問題常被研究者們拿來測試模型對自然語言的理解及推理能力。解決的問題主要有二，第一是藉由提出一個簡單且有效的模組將原先只能處理單回合問答的模型延伸至多回合問答。第二是改善問答模型對對抗性攻擊的穩固性，我們設計了一個基於最大化相互資訊的正則化來達到這個目標。

基於對話的多回合問答需要模型對交談過程有進一步的理解，而先前被提出的模型藉由隱含的對模型推理的過程建模來改善表現。本篇論文的第一部分在這上面做了更進一步的改善，我們提出藉由明確的對模型推理的過程進行建模，以使模型可以更好地擷取對回答問題有用的資訊。模型在 QuAC, CoQA 以及 SCONE 三個資料集上皆得到很好的效果，顯著的改善了表現且證明了其可以被應用在不同種類的模型上。

本篇論文的第二部分專注在改善模型對對抗性樣本的穩固性。雖然現在問答模型已經可以在傳統的測量標準上得到非常好的成績，它們仍是非常容易地被特別設計的混淆句子所欺騙，使人們對這些模型是否真正理解問題感到存疑。為了解決這個問題，我們首先專注在單回合的問答資料集上，並提出了一個藉由最大化問題、答案以及文章的相互資訊來實現的正則化。我們的正則化可以幫助模型不再只是用資料集中存在的膚淺相關性來回答問題。實驗結果顯示模型在 Adversarial-SQuAD 這個資料集上達到現在最好的表現。

在未來工作方面，將影像、聲音及常識引入問答模型是個重要的方

向，而進一步研究如何防禦對抗性攻擊可以幫助模型對問題及文章有更深一步的了解。除此之外，問答模型的半監督學習和自監督學習也是一個重要的研究主題，因為儘管是小孩也不需要現在模型需要的龐大資料集來學習如何解決簡單的閱讀測驗。我們的未來方向放在如何開發有效率，穩固，且可以應用在各情境的問答模型。



關鍵字：問答, 機器理解, 對話模型, 對抗攻擊, 穩固性



Abstract

The main purpose of this thesis is to solve problems related to question answering (QA), for it being widely used for training and testing machine comprehension and reasoning. We focus on two problems about generalization of single-turn QA models. Firstly, we propose a simple and effective module which models the information gain in the reasoning process to extend the single-turn QA models to multi-turn setting. Secondly, we aim to improve the robustness of QA models to adversarially generated examples by designing a novel regularizer utilizing mutual information maximization to guide the training process.

Multi-turn question answering as the dialog requires deep understanding of the dialogue flow, and the prior work proposed FlowQA to implicitly model the context representation in reasoning for better understanding. The first part of this thesis proposes to explicitly model the information gain through dialogue reasoning in order to allow the model to focus on more informative cues. The proposed module is evaluated on two conversational QA datasets Question Answering in Context (QuAC) and Conversational Question Answering Challenge (CoQA), and one sequential instruction understanding dataset Sequential Context-dependent Execution (SCONE) to shows the effectiveness. The proposed approach achieves significant improvement over baselines in all three datasets and demonstrates its capability of generalization to different QA models and tasks. ¹

The second part of this thesis focuses on improving the robustness of

¹The code is available at <https://github.com/MiuLab/FlowDelta>

QA models to adversarial examples. Standard accuracy metrics indicate that modern reading comprehension systems have achieved strong performance in many question answering datasets. However, the extent these systems truly understand language remains unknown, and existing systems are not good at distinguishing distractor sentences, which look related but do not actually answer the question. To address this problem, we first focus on models trained on single-turn extractive QA datasets, and propose QAInfomax as a regularizer in reading comprehension systems by maximizing mutual information among passages, questions, and answers. QAInfomax helps regularize the model to not simply learn the superficial correlation for answering questions. The experiments show that our proposed QAInfomax achieves the state-of-the-art performance on the benchmark Adversarial-SQuAD dataset.²

As for future work, QA can be extended to incorporate commonsense and features in multiple-modalities, and studying how to defense adversarial attacks in QA can lead models to deeper understanding of questions and paragraphs. Moreover, semi-supervised and self-supervised approaches of QA are worth exploring, as even children does not need so much training data to learn how to solve these simple questions. The efficient, robust, and generalizable QA systems is our most important research direction.

Keywords: question answering, machine comprehension, dialog modeling, adversarial attacks, robustness

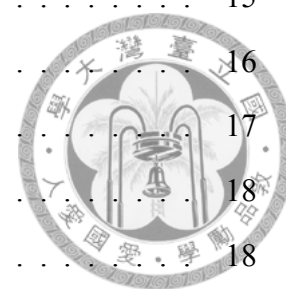
²The code is available at <https://github.com/MiuLab/QAInfomax>



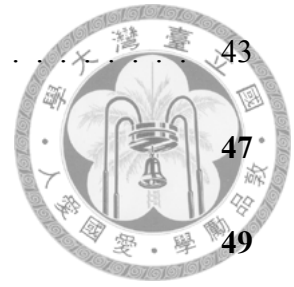
Contents

誌謝	iii
摘要	v
Abstract	vii
1 Introduction	1
1.1 Motivation	1
1.2 Main Contributions	3
1.3 Thesis Structure	3
2 Background	5
2.1 Task Formulation	5
2.2 Recurrent Neural Models	5
2.3 Pretrained Models for Language Understanding	7
2.3.1 Multi-Head Attention	8
2.3.2 Transformer Block	8
2.3.3 BERT	9
2.4 Mutual Information (MI) Estimation	10
2.5 Optimization and Metrics	12
2.5.1 Cross Entropy	12
2.5.2 F1 Score and Exact Match	12
3 Related Work	15

3.1	Question Answering Datasets	15
3.2	Dialog Datasets	16
3.3	Question Answering Models	17
3.4	Mutual Information Estimation	18
3.5	Adversarial Attacks	18
4	Accurate Conversational Question Answering	21
4.1	Notations	21
4.2	FusionNet	21
4.3	FlowQA	23
4.4	FlowDelta	24
4.4.1	FlowDeltaQA	24
4.4.2	BERT-FlowDelta	25
4.5	Experiments	26
4.5.1	Setup	27
4.5.2	Reducing SCONE to Conversational QA	28
4.5.3	Main Results	29
4.5.4	Ablation Study	30
4.5.5	Flow Information Gain Variants	30
4.6	Qualitative Analysis	32
5	Robust Question Answering	35
5.1	Notation	35
5.2	Methodology	36
5.2.1	Local Constraint	38
5.2.2	Global Constraint	39
5.2.3	QAInfomax	39
5.3	Experiments	40
5.3.1	Setup	40
5.3.2	Adversarial-SQuAD	41



5.3.3	Results	42
5.3.4	Qualitative Analysis	43
6	Discussion and Conclusion	47
	Bibliography	49







List of Figures

2.1	Example of the span-based QA dataset	6
4.1	Illustration of the flow information gain modeled by the FlowDelta mechanism.	23
4.2	Illustration of the proposed FlowDeltaQA model.	25
4.3	Illustration of the proposed BERT-FlowDelta model.	27
4.4	Example of the SCONE dataset and its reduction	28
4.5	Qualitative analysis of FlowDeltaQA.	33
5.1	Illustration of the LC and GC.	37
5.2	An example from the Adversarial-SQuAD dataset. BERT originally gets the answer correct, but is fooled by adversarial distracting sentence (in blue).	38
5.3	Examples from the Adversarial-SQuAD dataset. BERT originally gets the answer correct, but is fooled by adversarial distracting sentence (in blue).	45





List of Tables

3.1	A summarized view of QA datasets. Span, Multi-D and No-Answer refer to span-based, multiple context documents and unanswerable questions respectively.	17
3.2	A summarized view of QA models. RL, C-Embedding and P-LM refers to reinforcement learning, contextualized word embedding and pretrained language model respectively.	18
4.1	Conversational QA results on CoQA, where (N-ctx) refers to using previous N QA pairs (%).	30
4.2	Conversational QA results on QuAC, where (N-ctx) refers to using previous N QA pairs (%).	30
4.3	Dialogue accuracy for SCONE test (%).	31
4.4	The ablation study of BERT-FlowDelta (%).	31
4.5	CoQA results of different variants of flow interaction. All models are provided with previous 1 gold answer.	31
5.1	Ablation study with F1 scores on AddSent / AddOneSent. The speed is measured on RTX 2080Ti.	42
5.2	F-measure on AdversarialSquad (S: single, E: ensemble). [†] indicates the significant improvement over baselines with p-value < 0.05.	42
5.3	Different summarization functions for GC.	43





Chapter 1

Introduction

1.1 Motivation

Natural Language Processing (NLP) has been one of the central research area in artificial intelligence. By studying NLP, it can give us new insights into all aspect of our language, from how to acquire linguistics ability to why people and machine can understand and use natural language. In NLP, Question Answering (QA) tasks are widely used for training and testing machine comprehension and reasoning [1, 2, 3]. Recently, different variations [4, 5, 6] and attacks [7, 8] of QA dataset has been proposed. They can be considered as the generalization of well-studied QA datasets SQuAD to different scenarios and adversarial examples which are out of the training distribution. The generalization in QA is an important topic. Thus in this thesis, we focus on two essential generalization issues in QA - how to extend single-turn QA to multi-turn setting and how to make our QA systems more robust to adversarial examples.

QA or machine reading comprehension has been increasingly studied in the NLP area, which aims to read a given passage and then answer questions correctly. However, human usually seeks answers in a conversational manner by asking follow-up questions given the previous answers. Traditional QA tasks such as SQuAD [1] focus on a single-turn setting, and there is no connection between different questions and answers to the same passage. To address the multi-turn issue, several datasets about conversational question answering (CQA) were introduced, such as CoQA [9] and QuAC [10].

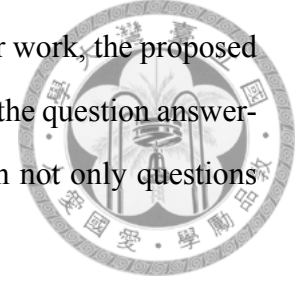
Most existing machine comprehension models [11] apply single-turn methods and augment the input with question and answer history, ignoring previous reasoning processes in the models. Recently proposed FlowQA [12] attempted at modeling such multi-turn reasoning in dialogues in order to improve performance for conversational QA. However, the proposed Flow operation is expected to incorporate salient information in an *implicit* manner, because the learned representations captured by Flow would change during multi-turn questions. It is unsure whether such change correlates well with the current answer or not. In order to *explicitly* model the information gain in Flow and further relate the current answer to the corresponding context, we present a novel mechanism, FlowDelta, which focuses on modeling the difference between the learned context representations in multi-turn dialogues.

On the other hand, in QA, high performance in standard automatic metrics has been achieved with only superficial understanding, as QA models exploit simple correlations in the data that happen to be predictive on most test examples [8]. Jia and Liang [7] addressed this problem and proposed an adversarial version of the SQuAD dataset, which was created by adding a distractor sentence to each paragraph. The distractor sentences challenge the model robustness, and the created Adversarial-SQuAD data shows the inability of a model about distinguishing a sentence that actually answers the question from one that merely has words in common with it, where almost all state-of-the-art machine comprehension systems are significantly degraded on adversarial examples.

Lewis and Fan [13] argued that over-fitting to superficial biases is partially caused by discriminative loss functions, which saturate when simple correlations allow the question to be answered confidently, leaving no incentive for further learning on the example. Therefore, they designed generative QA models, which use a generative loss function in question answering instead, and showed the improvement on Adversarial-SQuAD.

Instead of regularizing models by generative loss functions, we propose an alternative approach named “QAInfomax” by maximizing mutual information (MI) among passages, questions, and answers, aiming at helping models be not stuck with superficial biases in the data during learning. To efficiently estimate MI, QAInfomax incorporates the recently

proposed deep infomax (DIM) in the model [14], which was proved effective in learning representations for image, audio [15], and graph domains [16]. In our work, the proposed QAInfomax further extends DIM to the text domain, and encourages the question answering model to generate answers carrying information that can explain not only questions but also itself, and thus be more sensitive to distractor sentences.



1.2 Main Contributions

Conversational Question Answering

- We propose a simple and effective mechanism to explicitly model information gain in flow-based reasoning for multi-turn dialogues, which can be easily incorporated in different MC models.
- FlowDelta consistently improves the performance on various conversational MC datasets, including CoQA and QuAC.
- The proposed method achieves the state-of-the-art results among published models on QuAC and sequential instruction understanding task (SCONE).

Robustness of Question Answering

- We first attempt at applying DIM-based MI estimation as a regularizer for representation learning in the NLP domain.
- The proposed QAInfomax achieves the state-of-the-art performance on the Adversarial-SQuAD dataset without additional training data, demonstrating its better robustness.

1.3 Thesis Structure

The thesis is organized as below.

- Chapter 2 - Background
This chapter reviews background knowledge utilized in the proposed methods.
- Chapter 3 - Related Work
This chapter summarizes related work and discusses current challenges of the field.

- Chapter 4 - Accurate Conversational Question Answering

This chapter focuses on introducing the model dealing with conversational question answering and shows the conducted experiments for evaluation. Part of this research work has been presented in the following publication [17]:



- Y.-T. Yeh and Y.-N. Chen, “QAInfomax: Learning robust question answering system by mutual information maximization,” in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),(HongKong, China), pp. 3368—3373, Association for Computational Linguistics, Nov. 2019.

- Chapter 5 - Robust Question Answering

This chapter is dedicated to present a regularizer improving the robustness of QA models, and examines the effectiveness and efficiency of proposed regularizer. Part of this research work has been presented in the following publication [18]:

- Y.-T. Yeh and Y.-N. Chen, “FlowDelta: Modeling flow information gain in reasoning for conversational machine comprehension,” in Proceedings of the 2nd Workshop on Machine Reading for Question Answering, (Hong Kong, China), pp. 86—90, Association for Computational Linguistics, Nov. 2019.

- Chapter 6 - Discussion and Conclusion

This chapter discusses and concludes the contributions, and describes the potential future research directions.



Chapter 2

Background

In this chapter, we will give some background knowledge about tasks, models, training algorithms, and evaluation metrics.

2.1 Task Formulation

We first briefly introduce the detail of question answering. Given a context document (paragraph) and a question, the goal of QA models is to reason over the document and the question, and then generate the answer. It is similar to what we did in school tests, which is considered as a basic ability of human intelligence. While QA tasks can be in different formats such as multiple choices and free-form answers, in this thesis, we tackle the QA tasks in the extractive setting. In the extractive setting which is also referred as span-based QA, answers are guaranteed to be the span in the context document. The goal of QA models thus become to find the most appropriate span in the paragraph given the question. We give a simple example of span-based QA in Figure 2.1, and discuss different types of QA datasets in section 3.

2.2 Recurrent Neural Models

In this section, we introduced long short term memory (LSTM) [19] and gated recurrent unit (GRU) [20] used in conversational QA. As sequences we deal with get longer and

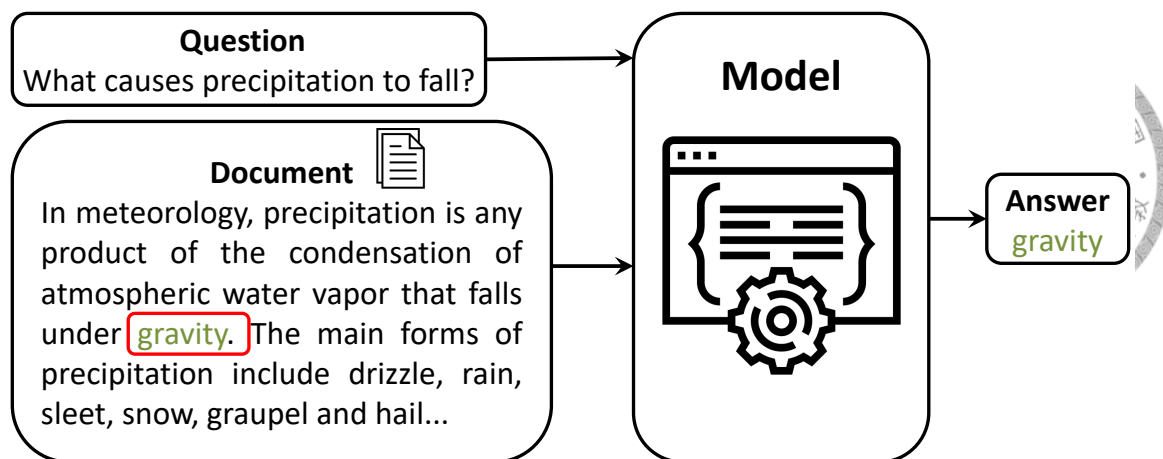


Figure 2.1: Example of the span-based QA dataset

longer, the vanilla RNN [21] encounters the gradient vanishing problem. The gradient vanishing problem occurs because neural network uses back propagation. In the back propagation step, the previous time steps may receive very small gradient after the gradient multiplies scalar smaller than one several times, which makes it untrainable. To avoid this problem, the gated mechanism is introduced in LSTM and GRU. The internal structure of a LSTM cell is:

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
 C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
 o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
 h_t &= o_t * \tanh(C_t)
 \end{aligned} \tag{2.1}$$

where W_f , W_i , W_C , and W_o are learnable weights. f_t , i_t and o_t are often referred as forget, input and output gates. By interacting with these three gates, we get the new hidden state h_t from h_{t-1} .

The internal structure of a GRU cell is:

$$\begin{aligned}z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\ \tilde{h}_t &= \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\ h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t\end{aligned}\tag{2.2}$$



where W_z , W_r and W are learnable weights. The new hidden state h_t is updated by the linear interpolation of original hidden state h_{t-1} and transformed hidden state \tilde{h}_t weighted by update gate.

2.3 Pretrained Models for Language Understanding

Pretrained Language model has been shown to be effective for improving many natural language processing tasks including QA [22, 23, 24, 25]. There are two existing strategies for applying pretrained language representation to down-stream tasks: *feature-based* and *fine-tuning*. The feature-based approach such as ELMo [23] use task-specific architectures that include the pretrained representations as additional features. Generally, the parameters of pretrained representations are freezed during training, or only the part of parameters will be fine-tuned. The fine-tuning approach such as BERT [26] introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning all pretrained parameters.

BERT [26] with fine-tuning recently has reached the state-of-the-art in many single-turn QA tasks, such as SQuAD [1, 27]. Each layer of BERT is a Transformer block [28] that consists of multi-head attention (MH) and fully-connected feed forward network (FFN).

2.3.1 Multi-Head Attention

An general attention function can be described as mapping query Q and a set of key-value pairs K - V to an output:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.3)$$



where d_k as scaling factor is the dimension of query Q and key K . The output is computed as a weighted sum of the values V , where the weight assigned to each value is computed by a compatibility function of the query Q with the corresponding key K .

Assuming the dimension of our model is d_{model} , instead of performing a single attention function with d_{model} -dimensional queries, keys, and values, it was found beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k , d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values, we can then perform the attention function in parallel, yielding d_v -dimensional output values. These are concatenated and once again projected, resulting in the final output of multi-head attention:

$$MH(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (2.4)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V),$$

where W^O , W_i^Q , W_i^K , and W_i^V are learned linear projection matrices. MH denotes multi-head attention.

2.3.2 Transformer Block

As described above, each Transformer block consists of multi-head attention (MH) and fully-connected feed forward network (FFN).

$$h_{l+1} = Transformer(h_l) = LN(h_l + FFN(LN(h_l + MH(h_l, h_l, h_l)))) \quad (2.5)$$

where h_l is the hidden representation of the l -th layer, and LN is layer normalization [29]:

$$\begin{aligned}
 LN(h) &= \frac{g}{\sigma} \odot (h - \mu) + b \\
 \mu &= \frac{1}{d_{model}} \sum_{i=1}^{d_{model}} h_i, \\
 \sigma &= \sqrt{\frac{1}{d_{model}} \sum_{i=1}^{d_{model}} (h_i - \mu)^2},
 \end{aligned} \tag{2.6}$$



where g and b are learned scaling parameters and \odot is element-wise multiplication between two vectors. Note that here we abuse the subscripts to denote the i -th scalar in h vector for simplicity.

$MH(h_l, h_l, h_l)$ is generally called self-attention or intra-attention. Self-attention is useful to relating different positions of a single sequence in order to compute a representation of the sequence. It has been used in a wide variety of NLP tasks including QA, abstractive summarization, textual entailment and learning task-independent sentence representations [30, 31, 32, 33].

2.3.3 BERT

The model architecture of BERT is the stack of multiple Transformer Blocks. Thus we apply multiple times equation 2.5 and get the output representation of BERT:

$$\text{BERT}(S) = \{r_1, r_2, \dots, r_n\} \tag{2.7}$$

where S is the input sequence with n words, and $\{r_1, r_2, \dots, r_n\}$ is the output representation of BERT. There are two steps in the BERT framework as described in the beginning of section 2: pretraining and fine-tuning. Here, we first introduce two pretraining tasks, Masked LM and NSP, of BERT.

The Masked Language Model (Masked LM) is analogue to the traditional language model, but the Masked LM is in bidirectional manner and thus more powerful than either a left-to-right LM or the shallow concatenation of a left-to-right and right-to-left model.

In Masked LM, we mask some percentage of the input tokens with $[MASK]$ token at random and then predict those masked tokens. It is worth noting that Masked LM is often referred to as a Cloze task in the literature [34].

Besides language modeling, many important downstream tasks such as QA and Natural Language Inference (NLI) are based on understanding the relationship between two sentences. In order to make a model understand sentence relationships, BERT is also pre-trained on Next Sentence Prediction (NSP). Specifically, while choosing the sentence A and B for each Masked LM training example, 50% of the time B is the actual next sentence that follows A , and 50% of the time it is a random sentence from the corpus. Then NSP train a model to predict whether the sentence B is the next sentence of A . It was showed the NSP task is beneficial on both QA and NLI.

The BERT model can be easily fine-tuned on span-based QA tasks. Specifically, we concatenate question and paragraph into a input sequence, and then feed it into BERT to get the representation. After obtaining the passage representation of the training instance, we use such representation to do the multi-class classification task. By classifying which word is the start and end of the answer span, we can compute the cross entropy loss and minimize them to fine-tune our BERT. In this thesis, due to its state-of-the-art performance, we use BERT as our baseline QA models and applied the proposed methods on it to show the improvement.

2.4 Mutual Information (MI) Estimation

In this section, we introduce how scalable estimation of mutual information is performed in terms of practical scenarios via mutual information neural estimation (MINE) [35] and the deep infomax (DIM) [14] described below.

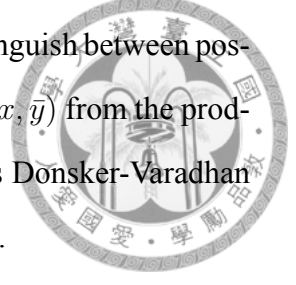
The mutual information between two random variable X and Y is defined as:

$$MI(X, Y) = D_{KL}(p(X, Y) \parallel p(X)p(Y)) \quad (2.8)$$

where D_{KL} is the Kullback-Leibler (KL) divergence between the joint distribution $p(X, Y)$

and the product of marginals $p(X)p(Y)$.

MINE estimates mutual information by training a classifier to distinguish between positive samples (x, y) from the joint distribution and negative samples (x, \bar{y}) from the product of marginals. Mutual information neural estimation (MINE) uses Donsker-Varadhan representation (DV) [36] as a variational lower-bound to estimate MI.



$$\text{MI}(X, Y) \geq \mathbb{E}_{\mathbb{P}}[g(x, y)] - \log(\mathbb{E}_{\mathbb{N}}[e^{g(x, \bar{y})}]) \quad (2.9)$$

where $\mathbb{E}_{\mathbb{P}}$ and $\mathbb{E}_{\mathbb{N}}$ denote the expectation over positive and negative samples respectively, and g is the discriminator function that outputs a real number modeled by a neural network.

While the DV representation is the strong bound of mutual information shown in MINE, we are primarily interested in maximizing MI but not focusing on its precise value. Thus DIM proposes an alternative estimation using Jensen-Shannon divergence (JS), which can be efficiently implemented using the binary cross entropy (BCE) loss:

$$\text{MI}(X, Y) \geq \mathbb{E}_{\mathbb{P}}[\log(g(x, y))] + \mathbb{E}_{\mathbb{N}}[\log(1 - g(x, \bar{y}))] \quad (2.10)$$

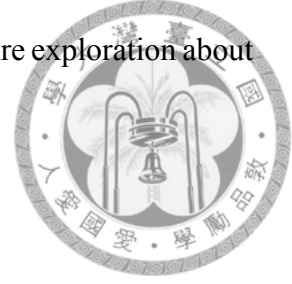
While two representations should behave similarly, considering that both act like classifiers with objectives maximizing the expected log-ratio of the joint over the product of marginals, it is found that the BCE loss empirically works better than the DV-based objective [14, 15, 16]. The reason may be that the BCE loss is bounded (i.e., its maximum is zero), making the convergence of the network more numerically stable. In our experiments, we primarily use the JS representation to estimate mutual information.

Recently, Tian et al. [37] showed strong empirical performance through the improved multiview CPC training [38], which shares many common ideas as mutual information maximization. Inspired by their work, we modify (2.10) by first switching the role of x and y and summing them up:

$$\text{MI}(X, Y) \geq \mathbb{E}_{\mathbb{P}}[\log(g(x, y))] + \frac{1}{2}\mathbb{E}_{\mathbb{N}}[\log(1 - g(x, \bar{y}))] + \frac{1}{2}\mathbb{E}_{\mathbb{N}}[\log(1 - g(\bar{x}, y))] \quad (2.11)$$

where (\bar{x}, y) is also the negative sample sampled from the product of marginals.

We empirically find that (2.11) gives the best performance, and more exploration about parameterization of MI is left as our future work.



2.5 Optimization and Metrics

2.5.1 Cross Entropy

The training process in section 2.3.3, and the MI maximization in section 2.4 both uses the cross entropy loss to maximize the likelihood. Here, we give the detail definition of cross entropy loss. After we get the output representation vectors $r = r_1, \dots, r_n$ from the model, we first feed them into the softmax function to get the probability distribution over all possible answers:

$$p_i = \text{softmax}(r)_i = \frac{\exp(r_i)}{\sum_{j=1}^n \exp(r_j)} \quad (2.12)$$

In binary classification cases such as NSP and MI estimation, the binary cross entropy (BCE) loss can be calculated as

$$BCE = -(y \log(p) + (1 - y) \log(1 - p)) \quad (2.13)$$

where y is the target binary label and p is the output probability.

In multiclass classification such as Masked LM, the cross entropy loss is calculated as $-\log(p_y)$, where p_y is the output probability assigned to target label y .

2.5.2 F1 Score and Exact Match

In extractive QA setting, we usually uses F1 score and Exact Match (EM) as main evaluation metrics. Given ground truth answer a , we can calculate the precision and recall of

our prediction p . The F1 score is defined as the harmonic mean of precision and recall:

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (2.14)$$

On the other hand, EM simply computes the exact match $\mathbb{1}[p = a]$.







Chapter 3

Related Work

3.1 Question Answering Datasets

Ideally, in QA we want answers to be arbitrary text and thus not to be limited as spans in documents. However, answers in free-form text are generally more hard to model as it involves the generation of natural language. The span-based question answering provides researchers a handful framework to focus on how to model the interaction between questions and documents. Recently, there are many different span-based question answering dataset proposed. For example, SQuAD [1] is the well-known span-based QA dataset and was built upon documents from Wikipedia. TriviaQA [2] pairs each question to multiple documents, so models also need to learn to rank potential documents to efficiently extract useful information. SQuAD 2.0 [27] and NewsQA [39] allow questions to be unanswerable. Hotpot QA [40] is a challenging dataset as it requires model to do multi-hop reasoning over multiple supporting evidences in different documents. Natural Questions [41] aims at providing natural questions for QA research, as most QA dataset use Amazon Mechanical Turk to crowdsource the data. Natural Questions uses search queries from Google Search Engine to generate QA examples. QuAC [10] extends QA to multi-turn setting, where each question may be conditioned on dialog in previous turns.

For real world applications, there are QA datasets with open-ended questions and free-form answers. Question in MS MARCO [42] are derived from real search queries and may be unanswerable. QA models need to rank set of candidate documents retrieved by Bing

search engine. ELI5 [43] collects data from Reddit forum "Explain Like I'm Five", where an online community provides answers to questions which are comprehensible by five year olds. ELI5 features long form answers consisting of multiple sentences. CoQA [9] is also the conversational QA dataset and its answers can be free-form text. There is a difference between CoQA and QuAC that when collecting data, questioners for QuAC can only see the title of the document, but questioners for CoQA can see the whole document.

Besides finding spans in the document or generating free-form answers, QA in the multiple-choice format is popular. RACE [44] collects English exams for middle and high school Chinese students. DREAM [45] is the QA in dialog format, but model only need to choose the most appropriate choice. CommonsenseQA [46] contains questions which require commonsense knowledge to solve. We provide a summarized view of these QA datasets in table 3.1.

There are also QA datasets which does not only tackle natural language data. Mohit et al.[47] created a sequential QA dataset about inquiring about tables from Wikipedia, which is similar to QuAC and CoQA. ComplexWebQuestions [48] propose a dataset consisting of complex questions and a framework which answers questions by first decomposing them into a sequence of simple questions. Each question is paired with a SPARQL query which can be executed against knowledge base, so ComplexWebQuestions can be used as a semantic parsing task. VQA [6] and AVSD [49] are multi-modal question answering datasets, which require models to integrate information from different modalities such as image, audio and text. It is worth to note that questions in AVSD is in the dialog format, which is highly related to conversational QA.

3.2 Dialog Datasets

Dialog datasets are highly related to conversational QA. Such datasets are mostly studied in the context of open-domain social chit-chat [50, 51, 52]. In the dialog, agents need to generate fluent, meaningful and coherent responses, and chat with users engagingly. Different from pure chit-chat, knowledge grounded dialog [53] require responses of agents are based on evidences from the knowledge base. Visual Dialog [54] rely on images as

Dataset	Span	Mutli-D	No-Answer	Multi-hop	Multi-Turns
SQuAD [1]	✓	-	-	-	-
TriviaQA [2]	✓	✓	-	-	-
NewsQA [39]	✓	-	✓	-	-
SQuAD 2.0 [27]	✓	-	✓	-	-
Hotpot QA [40]	✓	✓	-	✓	-
Natural Questions [41]	✓	-	✓	-	-
QuAC [10]	✓	-	✓	-	-
MS MARCO [42]	-	✓	✓	-	-
ELI5 [43]	-	-	-	-	-
CoQA [9]	-	-	✓	-	✓

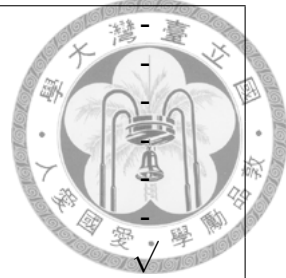


Table 3.1: A summarized view of QA datasets. Span, Multi-D and No-Answer refer to span-based, multiple context documents and unanswerable questions respectively.

evidence instead of text.

3.3 Question Answering Models

Although some QA datasets have free-form answers, by using the most similar span to the answer in the document, we can reduce such datasets to extractive QA setting in most cases. Thus here we mainly introduce extractive QA models, and do not discuss generative QA models using sequence-to-sequence model [55] and pretrained language models [56].

Match-LSTM [57] is the early representative work incorporating attention mechanism [58] to solve the machine reading comprehension task. Following Match-LSTM, DCN [59] proposes to use dynamic decoder to recover from initial local maxima corresponding to incorrect answers. BiDAF [11] uses bi-directional attention flow to fuse information from documents and questions and obtains better representation. DCN+ [60] and Reinforced Mnemonic Reader [61] improve the training objective by incorporating policy gradient [62]. R-Net [63], FusionNet [64] and QANet [65], which uses powerful Transformer architecture [28], apply self-attention to help model better relate different parts of long documents. By using contextualized word embedding such as Cove [66] and ELMo [23], QA models have further improvement by incorporating context information. Recently, pretraining language models on large corpus and then fine-tuning them on different datasets become a standard workflow of training QA models [67, 68]. Here, we provide

Model	Attention	RL	Self-Attention	C-Embedding	P-LM
Match-LSTM [57]	✓	-	-	-	-
DCN [59]	✓	-	-	-	-
BiDAF [11]	✓	-	-	-	-
DCN+ [60]	✓	✓	-	-	-
Mnemonic Reader [61]	✓	✓	✓	-	-
R-Net [63]	✓	-	✓	-	-
FusionNet [64]	✓	-	✓	✓	-
QANet [65]	✓	-	✓	-	-
BiDAF ++ [23]	✓	-	✓	✓	-
BERT [26]	✓	-	✓	-	✓

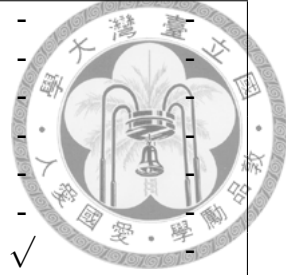


Table 3.2: A summarized view of QA models. RL, C-Embedding and P-LM refers to reinforcement learning, contextualized word embedding and pretrained language model respectively.

a summarized view of those models and important techniques used in table 3.2.

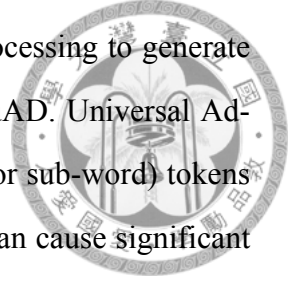
3.4 Mutual Information Estimation

How to efficiently measure mutual information in high dimensional space is a long-tailed and unsolved problem in information theory. There are estimators based on graphs and nearest neighbors [69, 70, 71]. By using neural network as a function approximator, recently there are works utilizing the variational bound of mutual information [36] to measure mutual information [35, 38, 72]. Following the estimator based on variational bound, there are works proposing to learn representations keeping desired properties by optimizing mutual information. Representations learned from mutual information estimators have been proved to be useful in different domains, such as language [73], image [14] and audio [15]. However, the limitation of learning representation by mutual information [74, 75] is also proposed, and researchers are trying to mitigate the problem [76].

3.5 Adversarial Attacks

Adversarial examples has been proved to be a a important issue in Deep Learning, and there are many different attack and defense methods proposed [77, 78, 79, 80]. However,

most of the works are done on image domain, and there are fewer ones targeted to NLP models. Adversarial SQuAD [7] uses templates and human post-processing to generate adversarial examples of the famous question answering dataset SQuAD. Universal Adversarial Triggers [81] performs gradient-guided search over word (or sub-word) tokens to find short trigger sequences. Such adversarial trigger sequences can cause significant performance drop of models on different NLP tasks such as NLI, QA and language generation. We can defend the attack of adversarial word substitutions by minimizing an upper bound on the worst-case loss [82]. It was shown that we could improve model robustness to adversarial examples and common input corruption by self-supervised learning [83], which is highly related to our work.







Chapter 4

Accurate Conversational Question

Answering

The main challenge in conversational QA is that current question may depend on the conversation history, which differs from the classic machine comprehension. Therefore, how to incorporate previous history into the QA model is especially important for better understanding. Prior work [12] proposes an effective way to model the reasoning in multi-turn dialogues summarized below.

4.1 Notations

We denote the paragraph as a sequence of N words $\mathbf{P} = \{p_1, p_2, \dots, p_N\}$, and the i -th question in the dialog $\mathbf{Q}_i = \{q_{i,1}, q_{i,2}, \dots, q_{i,K}\}$ as a sequence of K words. In the extractive question answering, the i -th answer in the dialog $\mathbf{A}_i = \{a_{i,1}, \dots, a_{i,M}\}$ is guaranteed to be the span $\{p_m, \dots, p_{m+M}\}$ in the paragraph.

4.2 FusionNet

Since our baseline FlowQA uses FusionNet as its backbone machine comprehension model, here we briefly introduce it. FusionNet is the model targeting SQuAD, which is the single-turn span-based QA dataset. Like most of the other famous QA models, FusionNet con-

sists of LSTM which integrates representation sequence, and attention modules which fuses information from different parts of the input.

Given word embedding of input paragraph p and question q , FusionNet first computes the attention between embedding, and concatenate attention output with embedding to form word features for paragraph $\{w_1^p, \dots, w_N^p\}$ and question $\{w_1^q, \dots, w_N^q\}$. We feed word features into two LSTM layers to get higher level concepts of paragraph and question.

$$h_1^{p1}, \dots, h_N^{p1} = \text{LSTM}(w_1^p, \dots, w_N^p), h_1^{q1}, \dots, h_K^{q1} = \text{LSTM}(w_1^q, \dots, w_N^q) \quad (4.1)$$

$$h_1^{p2}, \dots, h_N^{p2} = \text{LSTM}(h_1^{p1}, \dots, h_N^{p1}), h_1^{q2}, \dots, h_K^{q2} = \text{LSTM}(h_1^{q1}, \dots, h_K^{q1}) \quad (4.2)$$

We then collect features in different levels of reasoning to form history-of-word (HoW) features for each word in paragraph and question:

$$\text{HoW}_i^p = [w_i^p, h_i^{p1}, h_i^{p2}], \text{HoW}_i^q = [w_i^q, h_i^{q1}, h_i^{q2}] \quad (4.3)$$

where HoW_i^p is HoW of the i -th word in paragraph, and HoW_i^q is HoW of the i -th word in question. We perform attention between HoW of the paragraph and the question to fuse information.

$$\tilde{h}_i^{p1} = \sum_j \alpha_{ij}^1 h_j^{q1}, \alpha_{ij}^1 \propto \exp(S^1(\text{HoW}_i^p, \text{HoW}_j^q)) \quad (4.4)$$

$$\tilde{h}_i^{p2} = \sum_j \alpha_{ij}^2 h_j^{q2}, \alpha_{ij}^2 \propto \exp(S^2(\text{HoW}_i^p, \text{HoW}_j^q)) \quad (4.5)$$

$$S(x, y) = f(U(x))^T D f(U(y)) \quad (4.6)$$

where U and D are trainable parameter matrix, and D is diagonal. S^1 and S^2 indicates they have different parameters.

We now adds attention output into HoW of the paragraph and feed them into LSTM again to get higher level concept $\{v_1^p, \dots, v_N^p\}$. We also uses self-attention over HoW of the paragraph to consider distant parts of the paragraph $\tilde{v}_i^p = \sum_j \alpha_{ij}^v v_j^p, \alpha_{ij}^v \propto \exp(S^v(\text{HoW}_i^p, \text{HoW}_j^p))$. Finally, we uses $[v_i^p, \tilde{v}_i^p]$ as the representation of i -th word in the paragraph, and perform multi-class classification to predict where is the start and end of the answer span.

Conversation Flow (over Context)

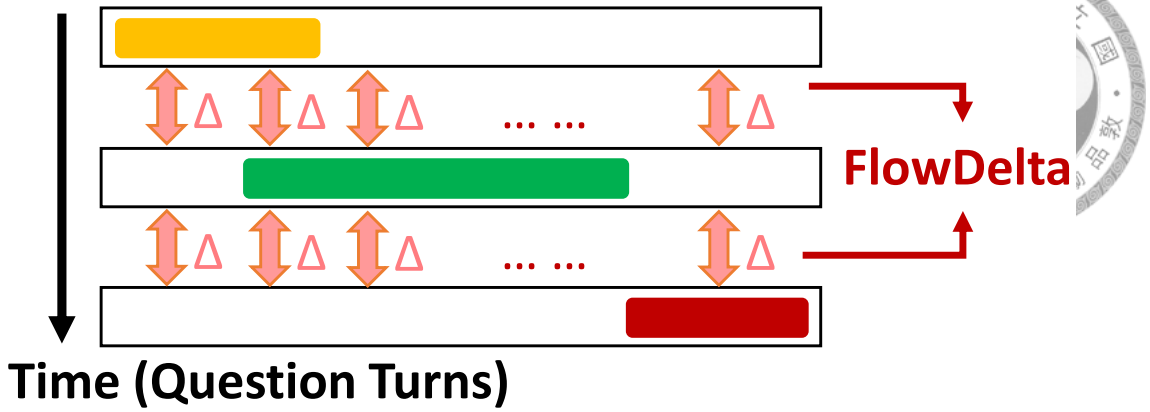


Figure 4.1: Illustration of the flow information gain modeled by the FlowDelta mechanism.

4.3 FlowQA

Instead of only using shallow history like previous questions and answers, Huang et al.[12] proposed the Flow operation that feeds the model with entire hidden representations generated during the reasoning process when answering previous questions. Flow is defined as *a sequence of latent representations based on the context tokens* and is demonstrated effective for conversational QA tasks, because it well incorporates multi-turn information in dialogue reasoning.

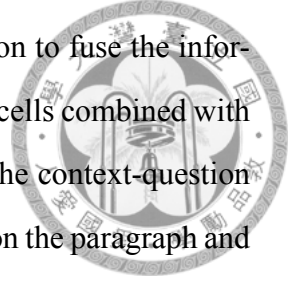
Let the document representation for i -th question be $\mathbf{R}_i = r_{i,1}, \dots, r_{i,M}$ and the dialogue length is T . When answering questions in the dialogue, there are T document representation sequences of length M , one for each question. We reshape it to become M sequences of length t , one for each document word, and then pass each sequence into a unidirectional GRU. All document word representation j ($1 \leq j \leq M$) are processed in parallel in order to model the information via the Flow direction (vertical direction illustrated in Figure 4.1).

$$h_{1,j}, \dots, h_{T,j} = GRU(r_{1,j}, \dots, r_{T,j}) \quad (4.7)$$

Then we reshape the outputs from GRU back and form $F_i = \{h_{i,1}, \dots, h_{i,M}\}$, where F_i is the output of the Flow layer.

The Flow layer described above is incorporated in FlowQA for conversational QA,

which is built on the single-turn QA model FusionNet [64], and the full structure is shown in the Figure 4.2. Briefly, FlowQA first performs word-level attention to fuse the information of i -th question Q_i into paragraph P . Then it uses two LSTM cells combined with Flow layers to integrate the paragraph representations, followed by the context-question attention computation. Finally, FlowQA performs self-attention [65] on the paragraph and predict the answer span. Modeling Flow is shown effective to improve the performance for conversational QA.



4.4 FlowDelta

We further extends the concept of Flow and proposes a flow-based approach, FlowDelta, to *explicitly* model information gain in flow during dialogues illustrated in Figure 4.1. The proposed mechanism is flexible to integrate with different models, including FlowQA and others. To examine such flexibility and generalization capability, we further apply Flow and FlowDelta to BERT [26] described in section 2.3.3 to allow model to grasp dialogue history.

4.4.1 FlowDeltaQA

In the original Flow operation in (4.7), the k -th step computation of GRU is $h_{k,j} = GRU(r_{k,j}, h_{k-1,j})$. We assume that the difference of previous hidden representations $h_{k-1,j}$ and $h_{k-2,j}$ indicates whether the flow change is important, which can be viewed as the information gain through the reasoning process. For example, 3 consecutive questions Q_{k-2}, Q_{k-1}, Q_k . Q_{k-1} and Q_k all discuss the same event described in the span $\{r_j, r_{j+1}, \dots, r_l\}$ of the context, while Q_{k-2} is about another topic. We expect the hidden state $\{h_{k-1,j}, h_{k-1,j+1} \dots, h_{k-1,l}\}$ of the span in turn $k-1$ is dissimilar to the hidden state in the turn $k-2$, because their topics are different. By explicitly modeling such difference, our model more easily relates the current reasoning process to the corresponding context.

Following the intuition above, we propose FlowDelta by modifying the single step

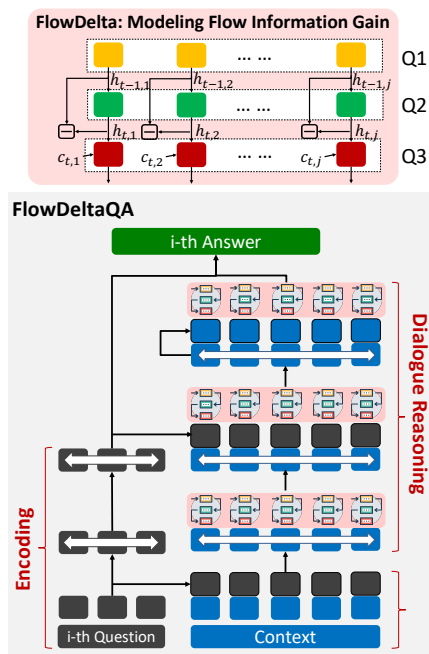


Figure 4.2: Illustration of the proposed FlowDeltaQA model.

computation of Flow into:

$$h_{k,j} = GRU([r_{k,j}; h_{k-1,j} - h_{k-2,j}], h_{k-1,j}), \quad (4.8)$$

where $[x; y]$ is the concatenation of the vectors x and y . We also investigate other variants such as Hadamard product ($h_{k-1,j} * h_{k-2,j}$) detailed in the experiment part. In optimization perspective, FlowDelta acts like a skip connection to help gradient flow more smoothly in the dialog. Learning to model the difference of hidden states is also conceptually similar to residual connection [84], which has been shown effective in many different tasks. The model overview is illustrated in Figure 4.2.

4.4.2 BERT-FlowDelta

As described in section 2.3.3, BERT [26] with fine-tuning recently has reached the state-of-the-art in many single-turn MC tasks, such as SQuAD [1, 27]. However, how to extend BERT to the multi-turn setting remains unsolved. We argue that the concatenation of previous dialog history and current question is infeasible to deal with the multi-turn problem. The main concern is because the general pretrained language model based on Transformer [28] limited the maximum input sequence length (512 tokens), so we need to truncate the

input question and context if the length exceeds limit. Although recently there are some works trying to mitigate the fixed-length context limitation in Transoformer [85, 68], it will introduce additional computation and memory overhead into the model. In the same time, it is also time-consuming to tune the best number of previous QA pairs for concatenation. To address this problem, we propose to incorporate the FlowDelta mechanism to deal with the multi-turn problem, where the Flow layer automatically integrates multi-turn information instead of tuning the number of QA pairs for inclusion.

BERT-FlowDelta incorporates the proposed FlowDelta mechanisms for two parts shown in the Figure 4.3. First, we add FlowDelta layer before the final prediction layer, $P^S, P^E = NN([h_L; \text{FlowDelta}(h_L)])$. Second, we further insert FlowDelta into the last BERT layer, considering that modeling dialogue history *within* BERT may be beneficial.

$$h_L = LN(h_{L-1} + FFN(LN(h_{L-1} + MH(h_{L-1}, h_{L-1}, h_{L-1}) + \text{FlowDelta}(h_{L-1}))) \quad (4.9)$$

These two modifications are called exFlowDelta and inFlowDelta respectively, and the latter also meets the idea from Stickland and Murray [86] who added additional parameters into BERT layers to improve the performance of multi-task learning. In our experiments, we only modify the last BERT layer to avoid largely increasing model size.

We also feed a additional paragraph features indicating which words are the answers of questions in previous turns into the BERT by learning an additional embedding matrix. Specifically, we use the binary vector to indicate whether the word is the answer in previous dialog, and multiply such binary vector with our learned embedding matrix to get the paragraph feature embedding e_p . We then add such paragraph feature embedding to the original BERT word embedding to incorporate the additional paragraph feature.

4.5 Experiments

To evaluate the effectiveness of the proposed FlowDelta, various tasks that contains dialogue history for understanding are performed in the following experiments.

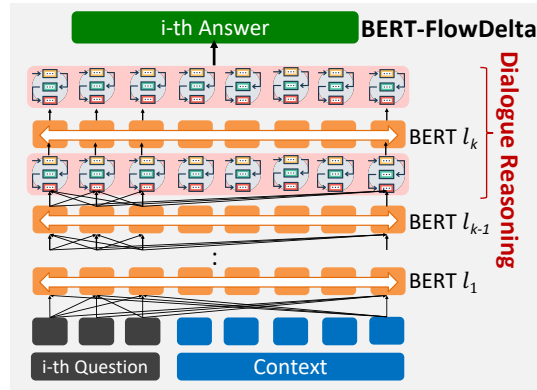
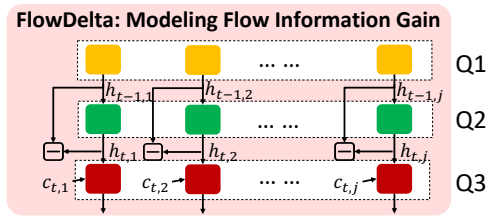


Figure 4.3: Illustration of the proposed BERT-FlowDelta model.

4.5.1 Setup

Our models are tested on two conversational QA datasets, CoQA [9] and QuAC [10], and a sequential instruction understanding dataset, SCONE [87]. For QuAC, we also report the Human Equivalence Score (HEQ). HEQ-Q and HEQ-D represent the percentage of exceeding the model performance over the human evaluation for each question and dialogue respectively. While CoQA and QuAC both follow the conversational QA setting, SCONE is the task requiring model to understand a sequence of natural language instructions and modify the word state accordingly. We follow Huang et al. [12] to reduce instruction understanding to machine comprehension and provide the example and reduction detail in section 4.5.2.

We reproduce and report the experiment results of FlowQA using the released code except SCONE part since the official released code does not contain it. Authors claim there is further performance improvement on the released version of FlowQA. All hyperparameters are kept the same as recommended one in FlowQA and BERT for CoQA and QuAC datasets. For SCONE, due to the relatively small size of dataset, to prevent overfitting we further tune the hidden size of FlowDeltaQA in three different domains. The tuned hidden sizes are 50, 60, 70 for Scene, Alchemy and Tangrams respectively.

4.5.2 Reducing SCONE to Conversational QA

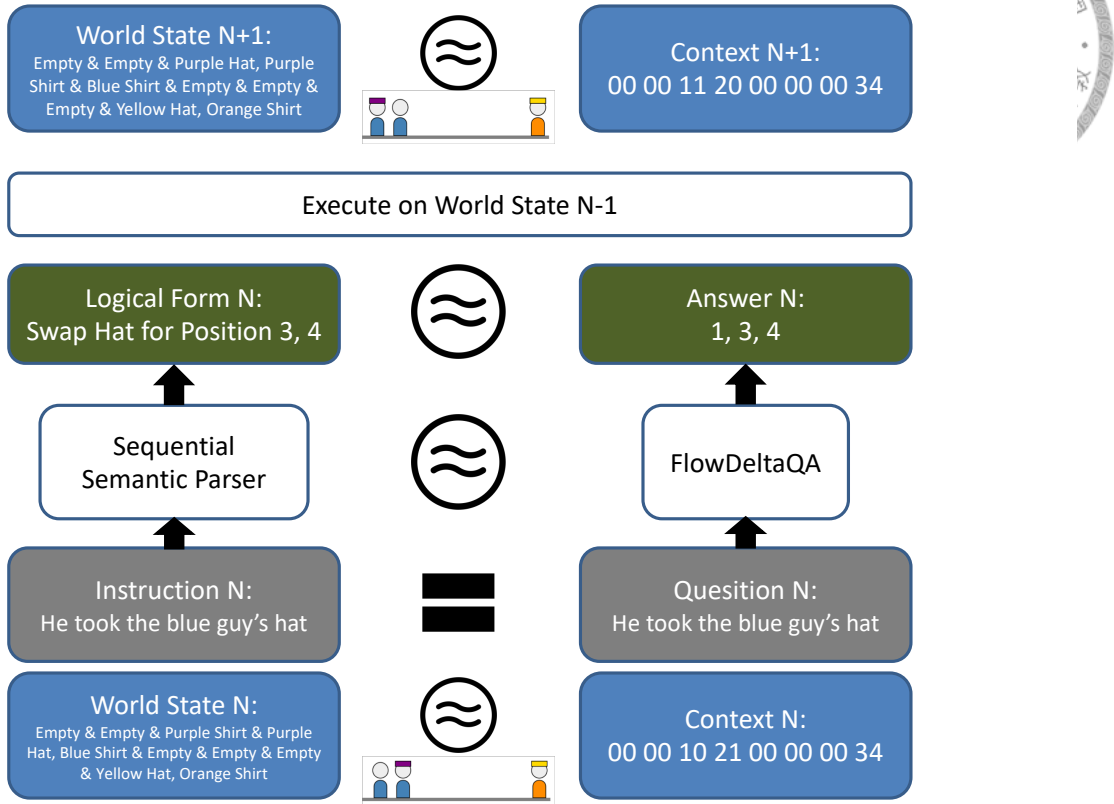


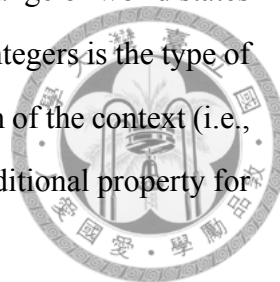
Figure 4.4: Example of the SCONE dataset and its reduction

In SCONE dataset, given the initial world state W_0 and a sequence of natural language instructions $\{I_1, \dots, I_K\}$, the model need to perform the correct sequence of actions on W_0 and obtain the correct world states $\{W_1, \dots, W_K\}$ after each instruction. An example from [87] is shown in the left-hand side of Figure 4.4.

There are three different settings SCENE, TANGRAMS and Alchemy in SCONE dataset. In SCENE, each environment has ten positions with at most one person at each position. This setting covers four actions (enter, leave, move, and trade-hats) and two properties (hat color, shirt color). In TANGRAMS, the environment is a list containing at most five shapes. This setting contains three actions (add, move, swap) and one property (shape). Lastly, in ALCHEMY, each environment is seven numbered beakers and covers three actions (pour, drain, mix) dealing with two properties (color, amount).

Following FlowQA [12], for each position in the world state, we encode it as two integers denoting the shirt and hat color in Scene, image ID and present or not in Tangrams,

and color of the liquid and number of units in Alchemy. Next, the change of world states (i.e., the logical form) is encoded as three or four integers. The first integer is the type of action performed. The second and third integers represent the position of the context (i.e., the encoded world state). Finally, the fourth integer represents the additional property for the action such as the number of units moved.



An example of encoded world states and logical form is shown in the right-hand side of Figure 4.4. In this example, action (1, 3, 4) means "swap the hat for position 3, 4" and there is no additional property for the action.

4.5.3 Main Results

Table 4.1 and 4.2 reports model performance on CoQA and QuAC. It can be found that FlowDeltaQA yields substantial improvement over FlowQA on both datasets (+ 0.9 % F1 on both CoQA and QuAC), showing the usefulness of explicitly modeling the information gain in the Flow layer. Furthermore, BERT-FlowDelta performs the best and outperforms the published models on QuAC leaderboard on Apr 24, 2019. Specifically, while BERT-FlowDelta achieves slightly worse HEQ-Q score on QuAC to the HAM model [88], we outperform HAM in HEQ-D metrics, showing the superiority of our model in modeling whole dialogue. Note that FlowDelta actually introduced few additional parameters compared to Flow, since it only augments the input dimension of GRU. The consistent improvement from both data demonstrates the generalization capability of applying the proposed mechanism to various models.

Table 4.3 shows the performance of our FlowDeltaQA on the SCONE¹. Our model outperforms FlowQA and achieves the state-of-the-art in Scene and Tangrams domains. The small performance drop in Alchemy aligns well with the statement in the ablation study. Because experiments show that removing Flow affects performance in Alchemy less when comparing between FlowQA and FusionNet [64] (same models except Flow), we claim that the previous dialogue history is less important in this domain. Thus replaying Flow with FlowDelta does not bring any improvement in the Alchemy domain but risk of

¹The results of BERT-FlowDelta are not shown, since SCONE is a relatively small and synthetic dataset.

Model	Dev	Test							
	F1	Child	Liter	Mid	News	Wiki	Reddit	Sci	F1
BiDAF++ (N-ctx)	69.2	66.5	65.7	70.2	71.6	72.6	60.8	67.1	67.8
FlowQA	76.7	73.7	71.6	76.8	79.0	80.2	67.8	76.1	75.0
SDNet [90]	78.0	75.4	73.9	77.1	80.3	83.1	69.8	76.8	76.6
FlowDeltaQA	77.6	-	-	-	-	-	-	-	-
BERT-FlowDelta	79.4	75.9	75.6	80.1	82.1	82.3	69.8	78.8	77.7
Human	89.8	90.2	88.4	89.8	88.6	89.9	86.7	88.1	88.8

Table 4.1: Conversational QA results on CoQA, where (N-ctx) refers to using previous N QA pairs (%).

Model	Dev	Test		
	F1	F1	HEQ-Q	HEQ-D
BiDAF++ (N-ctx)	60.6	60.1	54.8	4.0
FlowQA	63.9	64.1	59.6	5.8
HAM [88]	-	65.4	61.8	6.7
FlowDeltaQA	64.8	-	-	-
BERT-FlowDelta	66.1	65.5	61.0	6.9
Human	80.8	81.1	100	100

Table 4.2: Conversational QA results on QuAC, where (N-ctx) refers to using previous N QA pairs (%).

overfitting.

4.5.4 Ablation Study

Table 4.4 shows the ablation study of BERT-FlowDelta, where two proposed modules are both important for achieving such results. It is interesting that the proposed inFlowDelta and exFlowDelta boost the performance more on QuAC. As Yatskar [89] mentioned, the topics in a dialogue shift more frequently on QuAC than on CoQA, and we can see vanilla BERT also performs well on CoQA in the ablation of Flow which provides long term dialog history information. Therefore, we can conclude that while FlowDelta improves the ability to grasp information gain in the dialog, it bring less performance improvement in the setting we do not need much contexts to answer the question.

4.5.5 Flow Information Gain Variants

We test three different variants of FlowDelta on modeling the information flow in the dialog and show results in table 4.5. The three variants are:

Model	Scene	Tangrams	Alchemy
Long et al. [87]	14.7	27.6	52.3
Guu et al. [91]	46.2	37.1	52.9
Suhr and Artzi [92]	66.4	60.1	62.3
Fried et al. [93]	72.7	69.6	72.0
FusionNet	58.2	67.9	74.1
FlowQA	74.5	72.3	76.4
FlowDeltaQA	75.1	72.5	76.1

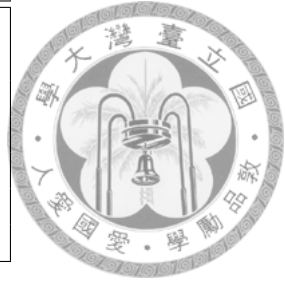


Table 4.3: Dialogue accuracy for SCONE test (%).

Model	CoQA F1	QuAC F1
BERT-FlowDelta	79.4	66.1
- inFlowDelta	79.0	64.1
- exFlowDelta	78.0	62.3
BERT-Flow	79.2	64.3

Table 4.4: The ablation study of BERT-FlowDelta (%).

- SkipDelta: $h_{t-1} - h_{t-3}$
- DoubleDelta: $[h_{t-1} - h_{t-2}; h_{t-2} - h_{t-3}]$
- Hadamard Product: $h_{t-1} * h_{t-2}$

The reason to use SkipDelta and DoubleDelta is because we want to see if there is any benefit to incorporate longer (or more) dialog history. Experiment results show while using longer dialog history (i.e., SkipDelta) helps, adding too many dialog history (i.e., DoubleDelta) does not give any improvement.

The intuition behind Hadamard product is to model the similarity of consecutive hidden states. If there are any topic shift in last turns of dialog, we expect Hadamard product can give us useful signal to detect it. Results show although the proposed FlowDelta is the best, Hadamard product outperforms SkipDelta and DoubleDelta and proves its effectiveness.

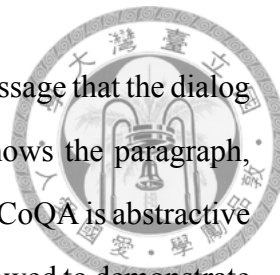
Model	F1
FlowQA	76.7
FlowDeltaQA (SkipDelta)	76.9
FlowDeltaQA (DoubleDelta)	76.7
FlowDeltaQA (Hadamard Product)	77.2
FlowDeltaQA	77.6

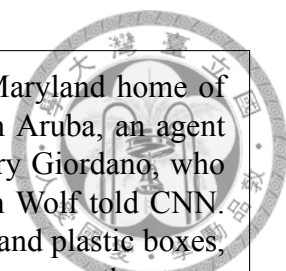
Table 4.5: CoQA results of different variants of flow interaction. All models are provided with previous 1 gold answer.

4.6 Qualitative Analysis

Here we present an example from CoQA dataset which consists of a passage that the dialog talks about, and a sequence of questions and answers. Table 4.5 shows the paragraph, questions, answers and model predictions. We note the gold answer in CoQA is abstractive and may not be a span in the passage. Only a subset of the dialog is showed to demonstrate the different behaviors of FlowQA and FlowDeltaQA.

In this example, to answer the last question "Why?", model need to understand the previous conversation correctly to know the actual question is "Why is Gary Giordano in the Aruban Jail now?". This example is particularly hard since in order to know "he" in "Where is he now?" refers to "Gary Giordano", model need to use the information from the very first question "Whose house was searched", which requires the ability to utilize full dialog history. While FlowQA fails to hook this question to the correct conversation context and respond reasonable but incorrect answer, our FlowDeltaQA successfully grasps long dialog flow and answers the correct span.





Paragraph: (CNN) – FBI agents on Friday night searched the Maryland home of the suspect in the recent disappearance of an American woman in Aruba, an agent said. The search is occurring in the Gaithersburg residence of Gary Giordano, who is currently being held in an Aruban jail, FBI Special Agent Rich Wolf told CNN. Agents, wearing vests that said FBI and carrying empty cardboard and plastic boxes, arrived about 8:40 p.m. Friday. About 15 unmarked cars could be seen on the street, as well as a Montgomery County police vehicle. Supervisory Special Agent Philip Celestini, who was at the residence, declined to comment further on the search, citing the active investigation. Aruban Solicitor General Taco Stein said earlier Friday that the suspect will appear in court Monday, where an investigating magistrate could order him held for at least eight more days, order him to remain on the island or release him outright due to a lack of evidence. Giordano was arrested by Aruban police on August 5, three days after Robyn Gardner was last seen near Baby Beach on the western tip of the Caribbean island. Giordano told authorities that he had been snorkeling with Gardner when he signaled to her to swim back, according to a statement. When he reached the beach, Gardner was nowhere to be found, Giordano allegedly said. The area that Giordano led authorities to is a rocky, unsightly location that locals say is not a popular snorkeling spot. Although prosecutors have continued to identify the 50-year-old American man by his initials, GVG, they also released a photo of a man who appears to be Giordano. His attorney, Michael Lopez, also has said that his client is being held as a suspect in Gardner’s death. Lopez has not returned telephone calls seeking comment.

Question1: Whose house was searched?

Prediction of FlowQA and FlowDeltaQA: Gary Giordano

Gold Answer: Gary Giordano

Question2: In what city?

Prediction of FlowQA and FlowDeltaQA: Gaithersburg

Gold Answer: Gaithersburg

Question3: County?

Prediction of FlowQA and FlowDeltaQA: Montgomery County

Gold Answer: Montgomery County

Question4: State?

Prediction of FlowQA and FlowDeltaQA: Maryland

Gold Answer: Maryland

Question5: Where is he now?

Prediction of FlowQA and FlowDeltaQA: Aruban jail

Gold Answer: Aruban jail

Question6: Why?

FlowQA Prediction: lack of evidence

FlowDeltaQA Prediction: 6 recent disappearance of an American woman

Gold Answer: suspect in the recent disappearance of an American woman

Figure 4.5: Qualitative analysis of FlowDeltaQA.





Chapter 5

Robust Question Answering

In this thesis, we not only want to generalize QA models to multi-turn setting but also want to improve the robustness. Thus in this section, we focus on single-turn extractive dataset SQuAD and its adversarial version AdversarialSQuAD.

5.1 Notation

Given question $Q = \{q_1, q_2, \dots, q_K\}$ and paragraph $P = \{p_1, p_2, \dots, p_N\}$, the encoded representations from the general QA model M can be formulated as:

$$\{r^q, r^p\} = \{r_1^q, \dots, r_K^q, r_1^p, \dots, r_N^p\} = M(Q, P) \quad (5.1)$$

where r^q and r^p are representations of the question and the passage respectively after the reasoning process in the QA system M .

In this chapter, we use BERT as our baseline QA models due to its state-of-the-art performance, so the encoded representation is further written as

$$BERT(Q, P) = \{r^q, r^p\} = \{r_1^q, \dots, r_K^q, r_1^p, \dots, r_N^p\} \quad (5.2)$$

As described in section 2, since the answer is a span in the paragraph, most QA models then feed the passage representation r^p to a single-layer neural network and obtain the span

start and end probabilities for each passage word. Then we can compute the cross entropy loss L_{span} , which is the negative sum of log probabilities of the predicted distributions indexed by true start and end indices illustrated in section 2.5.1, and minimize the loss



5.2 Methodology

Our QAInfomax aims at regularizing the QA system M to not simply exploit the superficial biases in the dataset for answering questions. Therefore, two constraints are introduced in order to guide the model learning.

1. **Local Constraint (LC)**: each answer word representation r_i^p in the answer representation $r^a = \{r_m^p, \dots, r_{m+M}^p\}$ should contain information about what the remaining answer words and its surrounding context are.
2. **Global Constraint (GC)**: the summarized answer representation $s = S(r^a)$ should maximize the averaged mutual information to all other question representations in r^q and passage representations in r^p , where S is a summarization function described below.

Intuitively, the model is expected to *choose the answer span after fully considering the entire question and paragraph*. However, traditional QA models suffered the over stability problem, and tended to be fooled by distractor answers, such as the one containing an unrelated human name. As Lewis and Fan [13] argued, we also believe that the main reason is that QA models are only trained to predict start and end positions of answer spans. Correlation in the dataset allows QA models to find shortcuts and ignore what the answer span looks like. A learned behavior of traditional QA models can be viewed as a simple pattern matching, such as choosing the 5-length span after the word “river” if a question is about a river and the context talks about countries in European.

Following the intuition, two constraints LC and GC are introduced to guide models to learn the desired behaviors. To prevent the model from only learning to match some specific word patterns to find the answer, LC forces the model to generate answer span representations while maximizing mutual information among words in the span and the

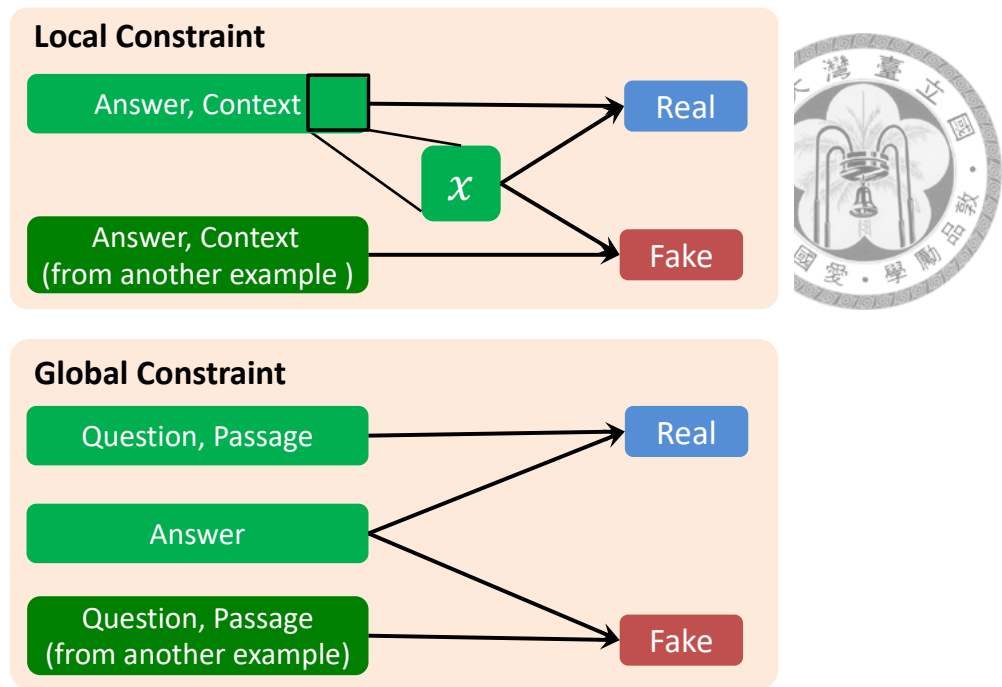


Figure 5.1: Illustration of the LC and GC.

context words surrounding the span. By maximizing the mutual information between an answer word and *all* of its context words, models need to incorporate the entire context into its decision process while choosing answers, and thus can be more robust to the adversarial sentences. Then we further require models to maximize mutual information among answer words, so models can no longer ignore any word in the chosen answer span.

On the other hand, different from LC, which only focuses on the answer span and its context, GC pushes the model to prefer answer representations carrying information that is globally shared across the whole input conditions Q and P , because shortcuts do not necessarily appear near to the answer. If the model only learns to leverage the correlation specific to the partial input, the MI of any input word without such relationship would *not* increased.

The overview about two proposed constraints is illustrated in Figure 5.1, and the example output is in 5.2. The detail of two constraints and our QAInfomax regularizer is described below.

Article: Force

Paragraph: A static equilibrium between two forces is the most usual way of measuring forces, using simple devices such as weighing scales and spring balances. For example, an object suspended on a vertical spring scale experiences the force of gravity acting on the object balanced by a force applied by the "spring reaction force", which equals the object's weight. Using such tools, some quantitative force laws were discovered: that the force of gravity is proportional to volume for objects of constant density (widely exploited for millennia to define standard weights); Archimedes' principle for buoyancy; Archimedes' analysis of the lever; Boyle's law for gas pressure; and Hooke's law for springs. These were all formulated and experimentally verified before Isaac Newton expounded his Three Laws of Motion. *Jeff Dean expounded on the Four Regulations of Action.*

Question: Who expounded the Three Laws of Motion?

Ground Truth: Isaac Newton

BERT Original Prediction: Isaac Newton

BERT Prediction under adversary: Jeff Dean

BERT + QAINfomax Prediction: Issac Newton

Figure 5.2: An example from the Adversarial-SQuAD dataset. BERT originally gets the answer correct, but is fooled by adversarial distracting sentence (in blue).

5.2.1 Local Constraint

As shown in Section 2.4, the maximization of MI needs positive samples and negative samples drawn from joint distribution and the product of marginal distribution respectively.

In LC, because all answer word representations are expected to carry the information of each other and their contexts, we choose to maximize averaged MI between the sampled answer word representations and the whole answer sequence with its context words. Specifically, a positive sample is obtained by pairing the sampled answer word representation $x \in r^a = \{r_m^p, \dots, r_{m+M}^p\}$ to all other answer and context words $r^c = \{r_{m-C}^p, \dots, r_{m+M+C}^p\} \setminus \{x\}$, where C is the hyperparameter defining how many context words for consideration. Negative samples, on the other hand, are obtained by randomly sampling answer representation $\bar{r}^a = \{\bar{r}_l^p, \dots, \bar{r}_{l+L}^p\}$ and the corresponding \bar{r}^c from other training examples. Following (2.11), the objective for sampled $x, r^c, \bar{x} \in \bar{r}^a$ and \bar{r}^c is

formulated.

$$\begin{aligned}
\text{LC}(x, r^c, \bar{x}, \bar{r}^c) &= \frac{1}{|r^c|} \sum_{r_i^c \in r^c} \log(g(x, r_i^c)) \\
&+ \frac{1}{2|\bar{r}^c|} \sum_{\bar{r}_j^c \in \bar{r}^c} \log(1 - g(x, \bar{r}_j^c)) \\
&+ \frac{1}{2|r^c|} \sum_{r_i^c \in r^c} \log(1 - g(\bar{x}, r_i^c)).
\end{aligned} \tag{5.3}$$



5.2.2 Global Constraint

Different from LC described above, GC forces the learned answer representations r^a to have information shared with all other question and passage representations. Here, we maximize the mutual information between the summarized answer vector $s = S(r^a)$ and $r_l \in r = \{r^q, r^p\} \setminus \{r^a\}$ pairs. In the experiments, we use $S(r^a) = \sigma(\frac{1}{M} \sum r_i^a)$ as our summarization function, where σ is the logistic sigmoid nonlinearity.

Specifically, a positive sample here is the pair of a answer summary vector $s = S(r^a)$ and all other word representations in r . Negative samples are provided by sampling question, passage and answer representations $\{\bar{r}^q, \bar{r}^p, \bar{r}^a\}$ from an alternative training example. Then we pair the summary s with $\bar{r} = \{\bar{r}^q, \bar{r}^p\} \setminus \{\bar{r}^a\}$, and $\bar{s} = S(\bar{r}^a)$ with r .

Similar to (5.3), the objective for the sampled s, r, \bar{s} and \bar{r} is:

$$\begin{aligned}
\text{GC}(s, r, \bar{s}, \bar{r}) &= \frac{1}{|r|} \sum_{r_i \in r} \log(g(s, r_i)) \\
&+ \frac{1}{2|\bar{r}|} \sum_{\bar{r}_j \in \bar{r}} \log((1 - g(s, \bar{r}_j))) \\
&+ \frac{1}{2|r|} \sum_{r_i \in r} \log((1 - g(\bar{s}, r_i))).
\end{aligned} \tag{5.4}$$

5.2.3 QAInfomax

In our proposed model, we combine two objectives and formulate the model as the complete QAInfomax regularizer. For each training batch consisting of training examples $\{\{Q_1, P_1, A_1\}, \dots, \{Q_B, P_B, A_B\}\}$, we pass the batch into the model M and obtain representations $\{\{r_1^q, r_1^p, r_1^a\}, \dots, \{r_B^q, r_B^p, r_B^a\}\}$. Note that we abuse the subscripts to denote

the example index in the batch for simplicity.

Then we shuffle the whole batch to obtain negative examples $\{\{\bar{r}_1^q, \bar{r}_1^p, \bar{r}_1^a\}, \dots, \{\bar{r}_B^q, \bar{r}_B^p, \bar{r}_B^a\}\}$. The complete objective L_{info} for QAInfomax becomes:

$$-\frac{1}{B} \sum_{i=1}^B (\alpha LC(x_i, r_i^c, \bar{x}_i, \bar{r}_i^c) + \beta GC(s_i, r_i, \bar{r}_i)),$$

where x_i and \bar{x}_i are the representation sampled from r_i^a and \bar{r}_i^a , r_i^c and \bar{r}_i^c are r_i^a and \bar{r}_i^a expanded with its context words respectively, s_i and \bar{s}_i are the summary vectors of r_i^a and \bar{r}_i^a , and α and β are hyperparameters.

Combined with QAInfomax as a regularizer, the final objective of the model becomes

$$L = L_{span} + \gamma L_{info}, \quad (5.5)$$

where L_{span} is the answer span prediction loss and γ is the regularize strength. The objective can be optimized through the simple gradient descent.

5.3 Experiments

To evaluate the effectiveness of the proposed QAInfomax, we conduct the experiments on a challenging dataset, Adversarial-SQuAD.

5.3.1 Setup

BERT-base [26] is employed as our QA system M in the experiments, where we set the same hyperparameters as one released in SQuAD training.¹

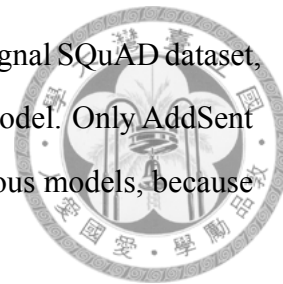
We set C , α , β and γ to be 5, 1, 0.5, 0.3 respectively in all experiments, and add the proposed QAInfomax into the BERT model as described above. The discriminator function g is the bilinear function similar to the scoring used by Oord et al.[38]:

$$g(x, y) = x^T W y, \quad (5.6)$$

¹We use PyTorch [94] reimplementation for experiments: <https://github.com/huggingface/pytorch-pretrained-BERT> [95]

where W is a learnable scoring matrix.

We train the BERT model with the proposed QAInfomax on the original SQuAD dataset, and use Adversarial-SQuAD to test the robustness of the augmented model. Only AddSent and AddOneSent metrics are reported for the comparison with previous models, because most previous models did not report their AddAny score.



5.3.2 Adversarial-SQuAD

Adversarial-SQuAD contains adversarial examples which aim to fool the QA models trained on SQuAD. As stated above, Adversarial-SQuAD has three main settings - AddSent, AddOneSent, and AddAny. In this paper, we reported scores of AddSent, AddOneSent for the comparison with previous models. AddSent and AddOneSent use a four-step procedure to generate the adversarial version of the example in SQuAD. In step 1, they first apply semantic-altering perturbation to the question by replacing nouns and adjectives with antonyms from WordNet [96], and change named entities and numbers to the nearest word in Glove [97] word embedding space. In step 2, they create a fake answer by the NER and POS tags of the original answer. For each answer, they compute its type of NER and POS tags and return the corresponding redefined fake answer. In step 3, the altered-question and fake answer will be combined into sentence in declarative form by using a set of roughly 50 manually-defined rules over constituency parses. Finally in step 4, errors in generated sentences are fixed by crowdsourcing. Each generated sentence is edited independently by five workers, resulting in up to five sentences for each raw sentence. The edited sentences are then concatenated to the original paragraph to form the adversarial example. For each example, AddSent runs the QA model M on every human-approved adversarial version of the example, picks the one that makes the model give the worst answer and returns that score. AddOneSent, on the other hand, only picks a random human-approved adversarial example and return its score for each example.

AddAny takes the search-based approach to generate adversarial examples regardless of grammaticality. For each example, it initialize a random word list and iteratively replaces each word with random candidates to minimize the expected F1 score of the QA

Model	Adversary F1	Speed (iter/s)
BERT	51.0 / 63.4	3.80
+ LC	53.6 / 64.2	3.51
+ GC	52.2 / 63.7	2.75
+ LC + GC	54.5 / 64.9	2.72



Table 5.1: Ablation study with F1 scores on AddSent / AddOneSent. The speed is measured on RTX 2080Ti.

Model	Original	AddSent	AddOneSent
BiDAF-S [98]	75.5	34.3	45.7
ReasoNet-S [99]	78.2	39.4	50.3
Reinforced Mnemonic Reader-S [61]	78.5	46.6	56.0
QANet-S [65]	83.8	45.2	55.7
GQA-S [13]	83.7	47.3	57.8
FusionNet-E [64]	83.6	51.4	60.7
BERT-S [26]	88.5	51.0	63.4
BERT-S + QAInfomax	88.6	54.5 [†]	64.9 [†]

Table 5.2: F-measure on AdversarialSquad (S: single, E: ensemble). [†] indicates the significant improvement over baselines with p-value < 0.05.

model output. This setting needs significantly more model access than the other two settings and takes a huge amount of time. As most previous other models did not report their score in this setting, in this thesis, we focus on natural adversarial sentences, and leave the arbitrary adversarial examples as our future work.

5.3.3 Results

Table 5.2 reports model performance on Adversarial-SQuAD. It can be found that QAInfomax yields substantial improvement over the vanilla BERT model, and achieves the state-of-the-art performance on both AddSent and AddOneSent metrics. Note that Wang and Bansal [100] modified distractor paragraphs and added them into training data, so we do not compare with them, because we only use the original SQuAD training data. QAInfomax obtains larger improvement on the AddSent, which picks the worst scores of the model. It shows the effectiveness of our QAInfomax in terms of forcing the model to ignore simple correlation in the data and learn the more human-like reasoning processes. It is worth to note that while QAInfomax mitigates the overstabliity problem and improves

Function	AddSent	AddOneSent
Mean	52.2	63.7
Max	52.0	63.3
Sample	52.2	63.0



Table 5.3: Different summarization functions for GC.

the robustness to adversarial examples, it does not hurt the original performance of the QA system, demonstrating the benefit for the practical usage. Some example results from the Adversarial-SQuAD dataset can be found in the Appendix, where adversarial distracting sentences are shown in italic blue fonts.

Table 5.1 shows the ablation study of our proposed QAInfomax, where two proposed constraints are both important for achieving such results. We also show the training speed of the proposed method and its limitation, where the GC objective degrades the training speed by 28%. The reason is that GC measures the averaged MI over the *whole* question and passage representations, which may include a long sequence of vectors.

Considering that the summarization function S plays an important role in GC, we explore its different variants in Table 5.3:

- Mean: $\sigma(\frac{1}{M} \sum r_i^a)$
- Max: $\sigma(\maxpool(r^a))$
- Sample: randomly sample one $r_i^a \in r^a$

According to the experimental results, Mean performs the best while Max and Sample has the competitive performance, showing the great robustness of the proposed methods to different architecture choices.

5.3.4 Qualitative Analysis

Here we present more examples from the Adversarial-SQuAD dataset in Figure 5.3, and adversarial distracting sentences are shown in blue. We can see the original BERT model is easily fooled by the distractor sentences similar to the questions. It demonstrates that QA models learn to answer questions by only exploiting superficial correlation between

questions and passages and not truly understand the questions. With our QAInfomax regularizer, the model is less likely fooled by distractor sentences. However, our proposed QAInfomax still can not defense the distractor sentences which are also hard for human and are sometimes ambiguous, showed in the second example of Figure 5.3.



Article: Islamism

Paragraph: The views of Ali Shariati, ideologue of the Iranian Revolution, had resemblance with Mohammad Iqbal, ideological father of the State of Pakistan, but Khomeini's beliefs is perceived to be placed somewhere between beliefs of Sunni Islamic thinkers like Mawdudi and Qutb. He believed that complete imitation of the Prophet Mohammad and his successors such as Ali for restoration of Sharia law was essential to Islam, that many secular, Westernizing Muslims were actually agents of the West serving Western interests, and that the acts such as "plundering" of Muslim lands was part of a long-term conspiracy against Islam by the Western governments.

The short term agenda of hamster was the acts of plundering Islamic lands by the East.

Question: What long term agenda was the acts of plundering Muslim lands by the West?

Ground Truth: conspiracy

BERT Original Prediction: conspiracy against Islam

BERT Prediction under adversary: The short term agenda of hamster

BERT + QAINfomax Prediction: conspiracy against Islam

Article: Force

Paragraph: Newton's First Law of Motion states that objects continue to move in a state of constant velocity unless acted upon by an external net force or resultant force. This law is an extension of Galileo's insight that constant velocity was associated with a lack of net force (see a more detailed description of this below). Newton proposed that every object with mass has an innate inertia that functions as the fundamental equilibrium "natural state" in place of the Aristotelian idea of the "natural state of rest". That is, the first law contradicts the intuitive Aristotelian belief that a net force is required to keep an object moving with constant velocity. By making rest physically indistinguishable from non-zero constant velocity, Newton's First Law directly connects inertia with the concept of relative velocities. Specifically, in systems where objects are moving with different velocities, it is impossible to determine which object is "in motion" and which object is "at rest". In other words, to phrase matters more technically, the laws of physics are the same in every inertial frame of reference, that is, in all frames related by a Galilean transformation *The Rosetta laws of physics refer to an object in motion and rest.*

Question: What are the laws of physics of Galileo, in reference to object in motion and rest?

Ground Truth: the laws of the physics are the same in every inertial frame of reference, that is, in all frames related by a Galilean transformation.

BERT Original Prediction: the same in every inertial frame of reference, that is, in all frames related by a Galilean transformation.

BERT Prediction under adversary: Rosetta laws

BERT + QAINfomax Prediction: Rosetta laws

Figure 5.3: Examples from the Adversarial-SQuAD dataset. BERT originally gets the answer correct, but is fooled by adversarial distracting sentence (in blue).





Chapter 6

Discussion and Conclusion

In this thesis, we focus on two problems about the generalization of QA models. The first part of this thesis presents how to generalize the single-turn QA models to conversational setting. We improve the Flow module in the previous work FlowQA by explicitly modeling the information gain in the reasoning process of the whole dialogue. Our proposed FlowDelta achieves significant improvement over FlowQA on two conversational QA datasets CoQA and QuAC with few additional parameters, which shows the superiority of our module. In the future, we want to work on generalizing QA in pure text to multi-modal QA which can deal with features in multiple modalities such image, audio and text. For example, as stated in section 3.1, AVSD is the multi-modal QA dataset which also consists of questions in the dialog manner. In our preliminary work [101], we show the model with complex attention module has small improvement over simple model which only do weighted sum of the features. Furthermore, we also find the image features are very noisy, sometimes incorporating it will even degrade the performance. Thus how to incorporate features in multiple modalities in QA still need more research, and we wish for designing a simple and effective model like FlowDelta to achieve it.

The second part of this thesis, improving the robustness of QA models is investigated. Our proposed QAInfomax is a regularizer which can improve the performance of QA models on Adversarial-SQuAD dataset, which consists of adversarial examples of the well-known SQuAD dataset. The intuition of our proposed QAInfomax is simple. We want to force the output representation of QA models to have some desired properties,

and such properties are not obtainable if models only exploit biases in the dataset. The mutual information maximization technique is used to realize our idea, and to the best of our knowledge we are the first to apply mutual information maximization as regularizer in NLP domain. By applying our QAInfomax to BERT model, we achieve state-of-the-art performance on the two settings of Adversarial-SQuAD dataset. We would like to investigate more adversarial attacks in the NLP domain and how to defend them as our future research direction. Specifically, how to automatically generate fluent adversarial examples with correct grammar is the important problem, as most existing methods does not take the naturalness of adversarial examples into consideration, or only use simple rules and crowdsourcing to deal with this problem.

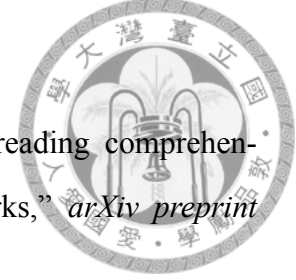




Bibliography

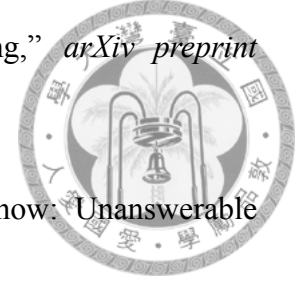
- [1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.
- [2] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," *arXiv preprint arXiv:1705.03551*, 2017.
- [3] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: a benchmark for question answering research," *Transactions of the Association of Computational Linguistics*, 2019.
- [4] H. Alamri, V. Cartillier, R. G. Lopes, A. Das, J. Wang, I. Essa, D. Batra, D. Parikh, A. Cherian, T. K. Marks, *et al.*, "Audio visual scene-aware dialog (avsd) challenge at dstc7," *arXiv preprint arXiv:1806.00525*, 2018.
- [5] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual dialog," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017.
- [6] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

- [7] R. Jia and P. Liang, “Adversarial examples for evaluating reading comprehension systems,” *arXiv preprint arXiv:1707.07328*, 2017.
- [8] D. Kaushik and Z. C. Lipton, “How much reading does reading comprehension require? a critical investigation of popular benchmarks,” *arXiv preprint arXiv:1808.04926*, 2018.
- [9] S. Reddy, D. Chen, and C. D. Manning, “Coqa: A conversational question answering challenge,” *arXiv preprint arXiv:1808.07042*, 2018.
- [10] E. Choi, H. He, M. Iyyer, M. Yatskar, W.-t. Yih, Y. Choi, P. Liang, and L. Zettlemoyer, “Quac: Question answering in context,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2174–2184, 2018.
- [11] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” *arXiv preprint arXiv:1611.01603*, 2016.
- [12] H.-Y. Huang, E. Choi, and W.-t. Yih, “Flowqa: Grasping flow in history for conversational machine comprehension,” *arXiv preprint arXiv:1810.06683*, 2018.
- [13] M. Lewis and A. Fan, “Generative question answering: Learning to answer the whole question,” 2018.
- [14] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, A. Trischler, and Y. Bengio, “Learning deep representations by mutual information estimation and maximization,” *arXiv preprint arXiv:1808.06670*, 2018.
- [15] M. Ravanelli and Y. Bengio, “Learning speaker representations with mutual information,” *arXiv preprint arXiv:1812.00271*, 2018.
- [16] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep graph infomax,” *arXiv preprint arXiv:1809.10341*, 2018.



- [17] Y.-T. Yeh and Y.-N. Chen, “FlowDelta: Modeling flow information gain in reasoning for conversational machine comprehension,” in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, (Hong Kong, China), pp. 86–90, Association for Computational Linguistics, Nov. 2019.
- [18] Y.-T. Yeh and Y.-N. Chen, “QAInfomax: Learning robust question answering system by mutual information maximization,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, (Hong Kong, China), pp. 3368–3373, Association for Computational Linguistics, Nov. 2019.
- [19] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [21] J. L. Elman, “Finding structure in time,” *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.
- [22] A. M. Dai and Q. V. Le, “Semi-supervised sequence learning,” in *Advances in neural information processing systems*, pp. 3079–3087, 2015.
- [23] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” *arXiv preprint arXiv:1802.05365*, 2018.
- [24] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding with unsupervised learning,” tech. rep., Technical report, OpenAI, 2018.
- [25] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” *arXiv preprint arXiv:1801.06146*, 2018.

- [26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [27] P. Rajpurkar, R. Jia, and P. Liang, “Know what you don’t know: Unanswerable questions for squad,” *arXiv preprint arXiv:1806.03822*, 2018.
- [28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.
- [29] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [30] J. Cheng, L. Dong, and M. Lapata, “Long short-term memory-networks for machine reading,” *arXiv preprint arXiv:1601.06733*, 2016.
- [31] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, “A decomposable attention model for natural language inference,” *arXiv preprint arXiv:1606.01933*, 2016.
- [32] R. Paulus, C. Xiong, and R. Socher, “A deep reinforced model for abstractive summarization,” *arXiv preprint arXiv:1705.04304*, 2017.
- [33] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, “A structured self-attentive sentence embedding,” *arXiv preprint arXiv:1703.03130*, 2017.
- [34] W. L. Taylor, “ “cloze procedure” : A new tool for measuring readability,” *Journalism Bulletin*, vol. 30, no. 4, pp. 415–433, 1953.
- [35] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, “Mine: mutual information neural estimation,” *arXiv preprint arXiv:1801.04062*, 2018.



- [36] M. D. Donsker and S. S. Varadhan, “Asymptotic evaluation of certain markov process expectations for large time. iv,” *Communications on Pure and Applied Mathematics*, vol. 36, no. 2, pp. 183–212, 1983.
- [37] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” *arXiv preprint arXiv:1906.05849*, 2019.
- [38] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [39] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman, “Newsqa: A machine comprehension dataset,” *arXiv preprint arXiv:1611.09830*, 2016.
- [40] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, “Hotpotqa: A dataset for diverse, explainable multi-hop question answering,” *arXiv preprint arXiv:1809.09600*, 2018.
- [41] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, *et al.*, “Natural questions: a benchmark for question answering research,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, 2019.
- [42] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, “Ms marco: A human-generated machine reading comprehension dataset,” 2016.
- [43] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, “Eli5: Long form question answering,” in *Proceedings of ACL 2019*, 2019.
- [44] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, “Race: Large-scale reading comprehension dataset from examinations,” *arXiv preprint arXiv:1704.04683*, 2017.
- [45] K. Sun, D. Yu, J. Chen, D. Yu, Y. Choi, and C. Cardie, “Dream: A challenge data set and models for dialogue-based reading comprehension,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 217–231, 2019.



- [46] A. Talmor, J. Herzig, N. Lourie, and J. Berant, “CommonsenseQA: A question answering challenge targeting commonsense knowledge,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4149–4158, Association for Computational Linguistics, June 2019.
- [47] M. Iyyer, W.-t. Yih, and M.-W. 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)*, (Vancouver, Canada), pp. 1821–1831, Association for Computational Linguistics, July 2017.
- [48] A. Talmor and J. Berant, “The web as a knowledge-base for answering complex questions,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 641–651, Association for Computational Linguistics, June 2018.
- [49] H. Alamri, V. Cartillier, R. G. Lopes, A. Das, J. Wang, I. Essa, D. Batra, D. Parikh, A. Cherian, T. K. Marks, *et al.*, “Audio visual scene-aware dialog (avsd) challenge at dstc7,” *arXiv preprint arXiv:1806.00525*, 2018.
- [50] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, “Personalizing dialogue agents: I have a dog, do you have pets too?,” *arXiv preprint arXiv:1801.07243*, 2018.
- [51] H. Fang, H. Cheng, E. Clark, A. Holtzman, M. Sap, M. Ostendorf, Y. Choi, and N. A. Smith, “Sounding board—university of washington’s alexa prize submission,” *Alexa prize proceedings*, 2017.
- [52] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, “Dailydialog: A manually labelled multi-turn dialogue dataset,” *arXiv preprint arXiv:1710.03957*, 2017.

- [53] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston, “Wizard of wikipedia: Knowledge-powered conversational agents,” *arXiv preprint arXiv:1811.01241*, 2018.
- [54] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, “Visual dialog,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 326–335, 2017.
- [55] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, pp. 3104–3112, 2014.
- [56] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI Blog*, vol. 1, no. 8, 2019.
- [57] S. Wang and J. Jiang, “Machine comprehension using match-lstm and answer pointer,” *arXiv preprint arXiv:1608.07905*, 2016.
- [58] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [59] C. Xiong, V. Zhong, and R. Socher, “Dynamic coattention networks for question answering,” *arXiv preprint arXiv:1611.01604*, 2016.
- [60] C. Xiong, V. Zhong, and R. Socher, “Dcn+: Mixed objective and deep residual coattention for question answering,” *arXiv preprint arXiv:1711.00106*, 2017.
- [61] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou, “Reinforced mnemonic reader for machine reading comprehension,” *arXiv preprint arXiv:1705.02798*, 2017.
- [62] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in *Advances in neural information processing systems*, pp. 1057–1063, 2000.

- [63] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, “Gated self-matching networks for reading comprehension and question answering,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 189–198, 2017.
- [64] H.-Y. Huang, C. Zhu, Y. Shen, and W. Chen, “Fusionnet: Fusing via fully-aware attention with application to machine comprehension,” *arXiv preprint arXiv:1711.07341*, 2017.
- [65] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, “Qanet: Combining local convolution with global self-attention for reading comprehension,” *arXiv preprint arXiv:1804.09541*, 2018.
- [66] B. McCann, J. Bradbury, C. Xiong, and R. Socher, “Learned in translation: Contextualized word vectors,” in *Advances in Neural Information Processing Systems*, pp. 6294–6305, 2017.
- [67] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [68] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” *arXiv preprint arXiv:1906.08237*, 2019.
- [69] H. Singh, N. Misra, V. Hnizdo, A. Fedorowicz, and E. Demchuk, “Nearest neighbor estimates of entropy,” *American journal of mathematical and management sciences*, vol. 23, no. 3-4, pp. 301–321, 2003.
- [70] D. Pál, B. Póczos, and C. Szepesvári, “Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graphs,” in *Advances in Neural Information Processing Systems*, pp. 1849–1857, 2010.

- [71] S. Gao, G. Ver Steeg, and A. Galstyan, “Efficient estimation of mutual information for strongly dependent variables,” in *Artificial intelligence and statistics*, pp. 277–286, 2015.
- [72] B. Poole, S. Ozair, A. v. d. Oord, A. A. Alemi, and G. Tucker, “On variational bounds of mutual information,” *arXiv preprint arXiv:1905.06922*, 2019.
- [73] L. Kong, C. d. M. d’Autume, W. Ling, L. Yu, Z. Dai, and D. Yogatama, “A mutual information maximization perspective of language representation learning,” *arXiv preprint arXiv:1910.08350*, 2019.
- [74] D. McAllester and K. Statos, “Formal limitations on the measurement of mutual information,” *arXiv preprint arXiv:1811.04251*, 2018.
- [75] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic, “On mutual information maximization for representation learning,” *arXiv preprint arXiv:1907.13625*, 2019.
- [76] S. Ozair, C. Lynch, Y. Bengio, A. v. d. Oord, S. Levine, and P. Sermanet, “Wasserstein dependency measure for representation learning,” *arXiv preprint arXiv:1903.11780*, 2019.
- [77] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, “The security of machine learning,” *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.
- [78] D. Lowd and C. Meek, “Adversarial learning,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 641–647, ACM, 2005.
- [79] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” *arXiv preprint arXiv:1611.01236*, 2016.
- [80] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” *IEEE transactions on neural networks and learning systems*, 2019.

- [81] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh, “Universal adversarial triggers for nlp,” *arXiv preprint arXiv:1908.07125*, 2019.
- [82] R. Jia, A. Raghunathan, K. Göksel, and P. Liang, “Certified robustness to adversarial word substitutions,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.
- [83] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, “Using self-supervised learning can improve model robustness and uncertainty,” in *Advances in Neural Information Processing Systems*, pp. 15637–15648, 2019.
- [84] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [85] Z. Dai, Z. Yang, Y. Yang, W. W. Cohen, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” *arXiv preprint arXiv:1901.02860*, 2019.
- [86] A. C. Stickland and I. Murray, “Bert and pals: Projected attention layers for efficient adaptation in multi-task learning,” *arXiv preprint arXiv:1902.02671*, 2019.
- [87] R. Long, P. Pasupat, and P. Liang, “Simpler context-dependent logical forms via model projections,” *arXiv preprint arXiv:1606.05378*, 2016.
- [88] C. Qu, L. Yang, M. Qiu, Y. Zhang, C. Chen, W. B. Croft, and M. Iyyer, “Attentive history selection for conversational question answering,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1391–1400, 2019.
- [89] M. Yatskar, “A qualitative comparison of coqa, squad 2.0 and quac,” *arXiv preprint arXiv:1809.10735*, 2018.



- [90] C. Zhu, M. Zeng, and X. Huang, “Sdnet: Contextualized attention-based deep network for conversational question answering,” *arXiv preprint arXiv:1812.03593*, 2018.
- [91] K. Guu, P. Pasupat, E. Z. Liu, and P. Liang, “From language to programs: Bridging reinforcement learning and maximum marginal likelihood,” *arXiv preprint arXiv:1704.07926*, 2017.
- [92] A. Suhr and Y. Artzi, “Situating mapping of sequential instructions to actions with single-step reward observation,” *arXiv preprint arXiv:1805.10209*, 2018.
- [93] D. Fried, J. Andreas, and D. Klein, “Unified pragmatic models for generating and following instructions,” *arXiv preprint arXiv:1711.04987*, 2017.
- [94] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.
- [95] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, “Huggingface’s transformers: State-of-the-art natural language processing,” *ArXiv*, vol. abs/1910.03771, 2019.
- [96] G. A. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.
- [97] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.
- [98] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” *CoRR*, vol. abs/1611.01603, 2016.
- [99] Y. Shen, P.-S. Huang, J. Gao, and W. Chen, “Reasonet: Learning to stop reading in machine comprehension,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1047–1055, ACM, 2017.

- [100] Y. Wang and M. Bansal, “Robust machine comprehension models via adversarial training,” *arXiv preprint arXiv:1804.06473*, 2018.
- [101] Y.-T. Yeh, T.-C. Lin, H.-H. Cheng, Y.-H. Deng, S.-Y. Su, and Y.-N. Chen, “Reactive multi-stage feature fusion for multimodal dialogue modeling,” *arXiv preprint arXiv:1908.05067*, 2019.

