Shortcut Learning in Question Answering

Department of Computer Science, Yonsei University

Seungone Kim

louisdebroglie@yonsei.ac.kr

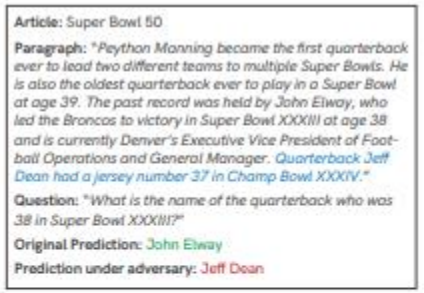
Referenced Papers

- Prerequisites / Additional Papers
  - Shortcut Learning in Deep Neural Networks [Nature Machine Intelligence 2020]
  - Gated Self-Matching Networks for Reading Comprehension and Question Answering [ACL 2017]
  - Bidirectional Attention Flow for Machine Comprehension [ICLR 2017]
  - Compositional Attention Networks for machine reasoning [ICLR 2018]
  - What makes reading comprehension questions easier? [EMNLP 2018]
  - Did the model understand the question? [ACL 2018]
  - Compositional questions do not necessitate multi-hop reasoning [ACL 2019]
  - Adversarial examples for evaluating reading comprehension systems [EMNLP 2017]
  - Assessing the benchmarking capacity of machine reading comprehension datasets [AAAI 2020]
  - A self-training method for machine reading comprehension with soft evidence extraction [ACL 2020]
  - Is attention interpretable? [ACL 2019]
  - Attention is not explainable [NAACL 2019]
  - Dynamically fused graph network for multi-hop reasoning [ACL 2019]
  - Hierarchical graph network for multi-hop question answering [EMNLP 2020]
  - Learning to retrieve reasoning paths over Wikipedia graph for question answering [ICLR 2019]
  - Revealing the importance of semantic retrieval for machine reading at scale [EMNLP 2019]
  - A simple yet strong pipeline for hotpotqa [EMNLP 2020]
  - Understanding design choices for multi-hop reasoning [NAACL 2019]
  - Finding Generalizable Evidence by Learning to Convince Q&A Models [EMNLP 2019]

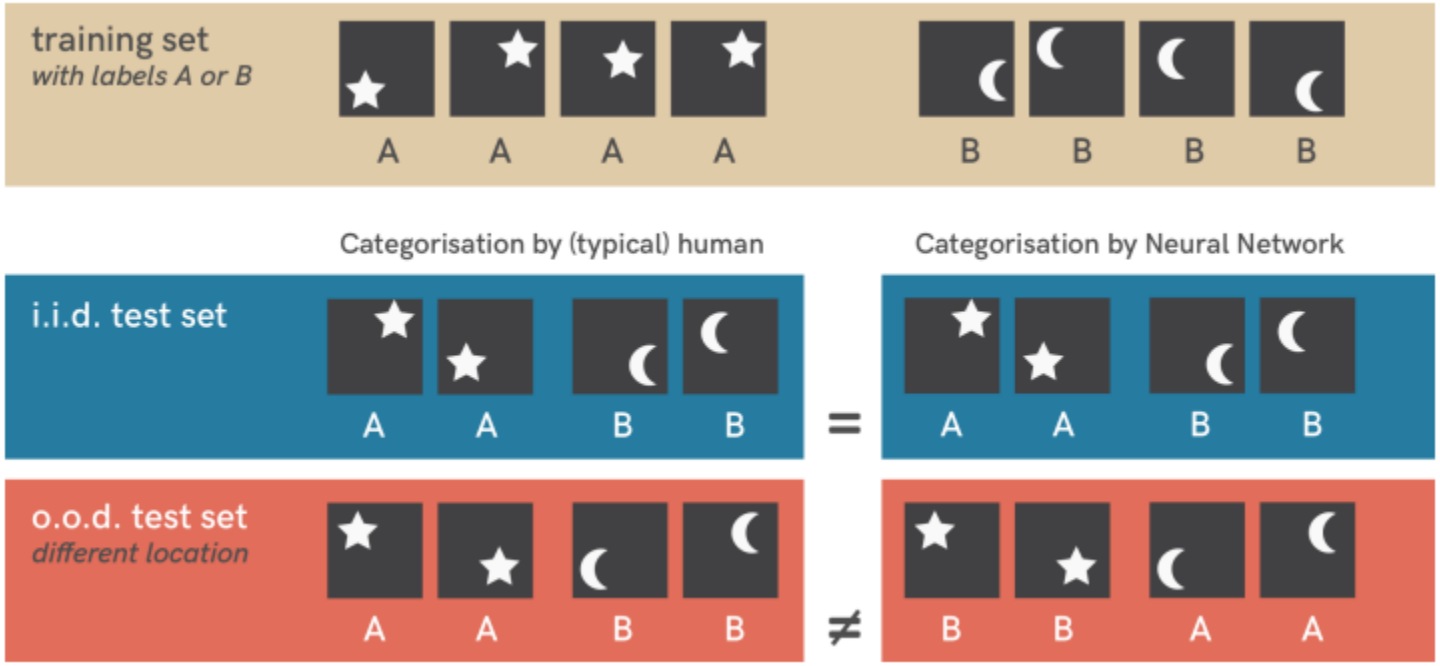
- Key Papers
  - Avoiding reasoning shortcuts : Adversarial evaluation, training, and model development for multi-hop QA [ACL 2019]
  - Why Machine Reading Comprehension Models Learn Shortcuts? [ACL 2021]
  - Robustifying multi-hop QA through pseudo-evidentiality training [ACL 2021]

What is Shortcut Learning?

- For some time, the tremendous success of deep learning has perhaps overshadowed the need to thoroughly understand the behaviour of Deep Neural Networks (DNNs).
- **Shortcuts** are *decision rules that perform well on standard benchmarks but fail to transfer to more challenging testing conditions*, such as real-world scenarios.
- When does deep learning work? When does it fail, and why?

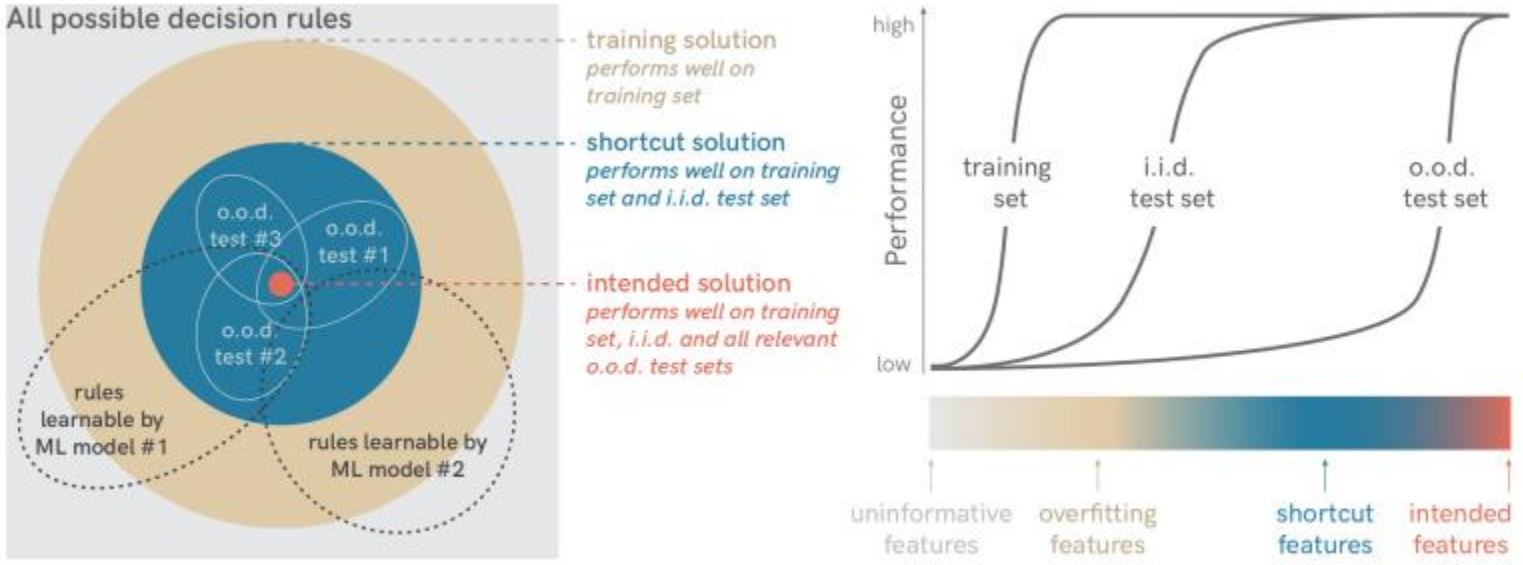
				
Task for DNN	Caption image	Recognise object	Recognise pneumonia	Answer question
Problem	Describes green hillside as grazing sheep	Hallucinates teapot if certain patterns are present	Fails on scans from new hospitals	Changes answer if irrelevant information is added
Shortcut	Uses background to recognise primary object	Uses features irrecognisable to humans	Looks at hospital token, not lung	Only looks at last sentence and ignores context

What is Shortcut Learning?

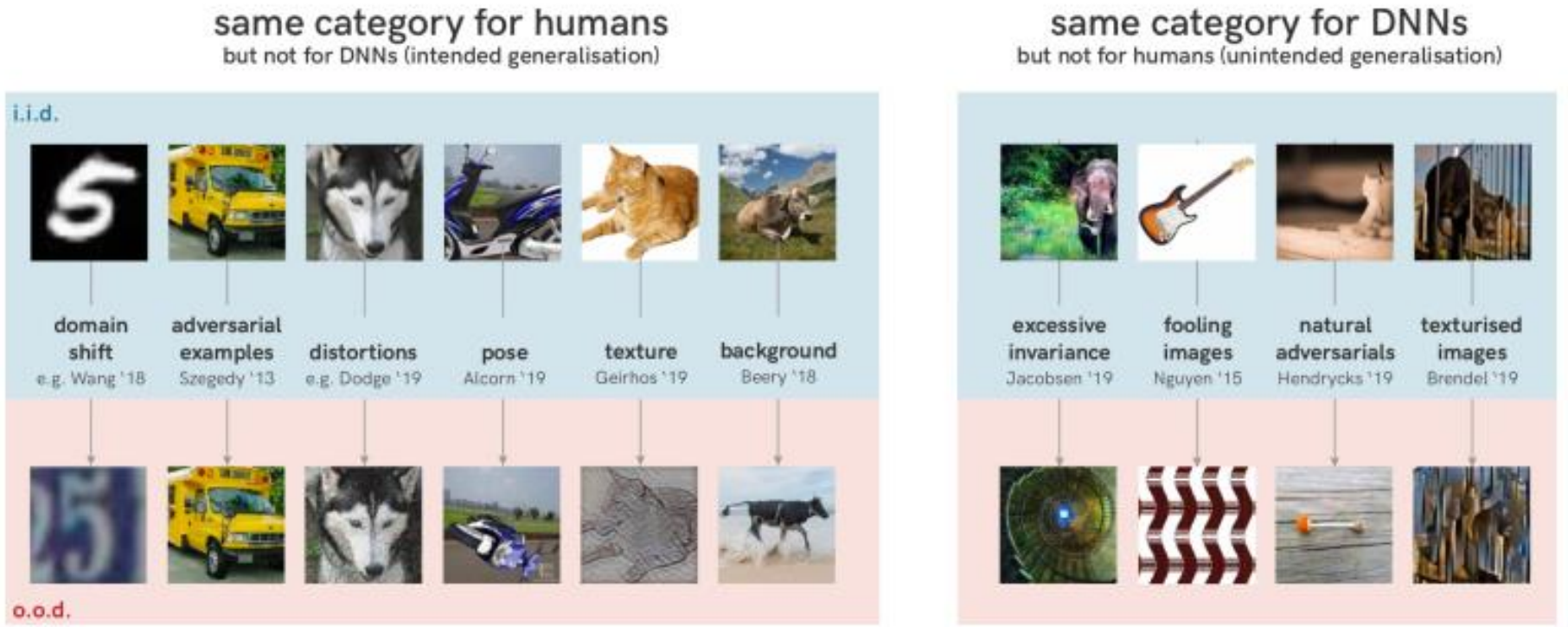


What is Shortcut Learning?

- **Decision rules** that solve both the training and i.i.d. test set typically score high on standard benchmark leaderboards.
- As long as tests are performed only on i.i.d. data, it is impossible to distinguish on which decision rules the model learned on.
- However, one can instead test models on datasets that are systematically different from i.i.d. training and test data (o.o.d. test data)
- Decision rules that use the **intended features** work well not only on an i.i.d. test set but also perform as intended on o.o.d. test data



What is Shortcut Learning?



Avoiding Reasoning Shortcuts: Adversarial Evaluation, Training, and Model Development for Multi-Hop QA

Yichen Jiang and **Mohit Bansal**
UNC Chapel Hill
{yichenj, mbansal}@cs.unc.edu

ACL 2019

Avoiding Reasoning Shortcuts : Adversarial Evaluation, Training, and Model Development for Multi-Hop QA [ACL 2019]

- **Multi-hop question answering** requires a model to *connect multiple pieces of evidence scattered in a long context to answer the question*.
- **HotpotQA** (Yang et al., 2018) dataset often contain reasoning shortcuts through which models can directly locate the answer by **word-matching** the question with a sentence in the context.
  - Therefore, a model performing well on the existing evaluation does not necessarily suggest its strong **compositional reasoning ability**
  - To truly promote and evaluate a model's ability to perform multi-hop reasoning, there should be no such reasoning shortcut where the model can locate the answer with single-hop reasoning only.
  - This is a common pitfall when collecting multi-hop examples and is difficult to address properly.

- (Contribution 1) Demonstrate the issue by constructing **adversarial documents** that create contradicting answers to the shortcut but do not affect the validity of the original answer.
  - Apply **phrase-level perturbations** to the answer span and the titles in the supporting documents to create the adversary with a new title and a fake answer to confuse the model.
  - With the adversary added to the context, it is no longer possible to locate the correct answer with the single-hop shortcut, which now leads to two possible answers.
  - E.g., "World's Best Goalkeeper" vs "World's Best Defender"
  - Evaluate with BiDAF (Seo et al., 2018) and R-Net (Wang et al., 2017) from HotpotQA on the newly constructed adversarial dev set(adv-dev) and find EM score drops significantly.

Avoiding Reasoning Shortcuts : Adversarial Evaluation, Training, and Model Development for Multi-Hop QA [ACL 2019]

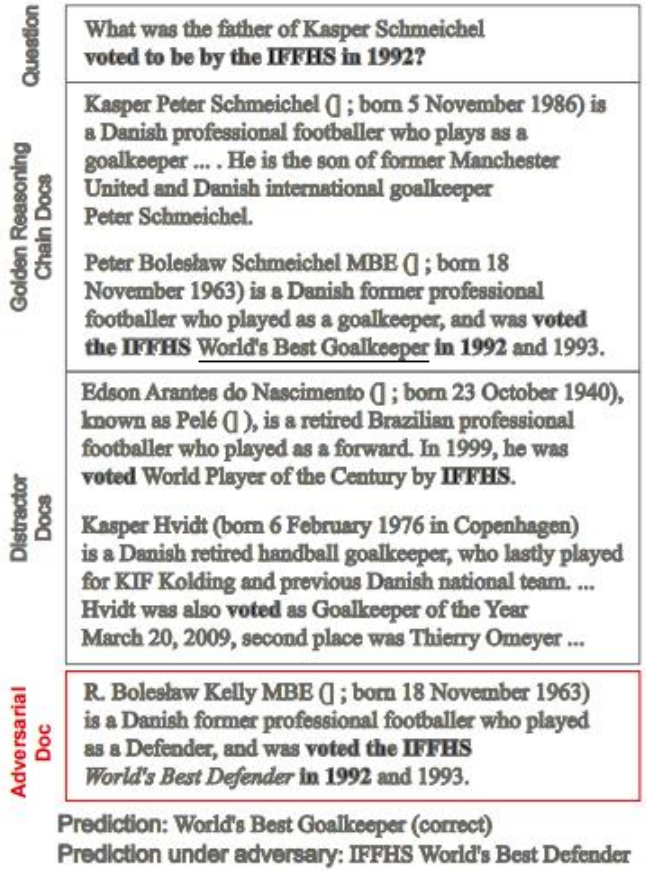


Figure 1: HotpotQA example with a reasoning shortcut, and our adversarial document that eliminates this shortcut to necessitate multi-hop reasoning.

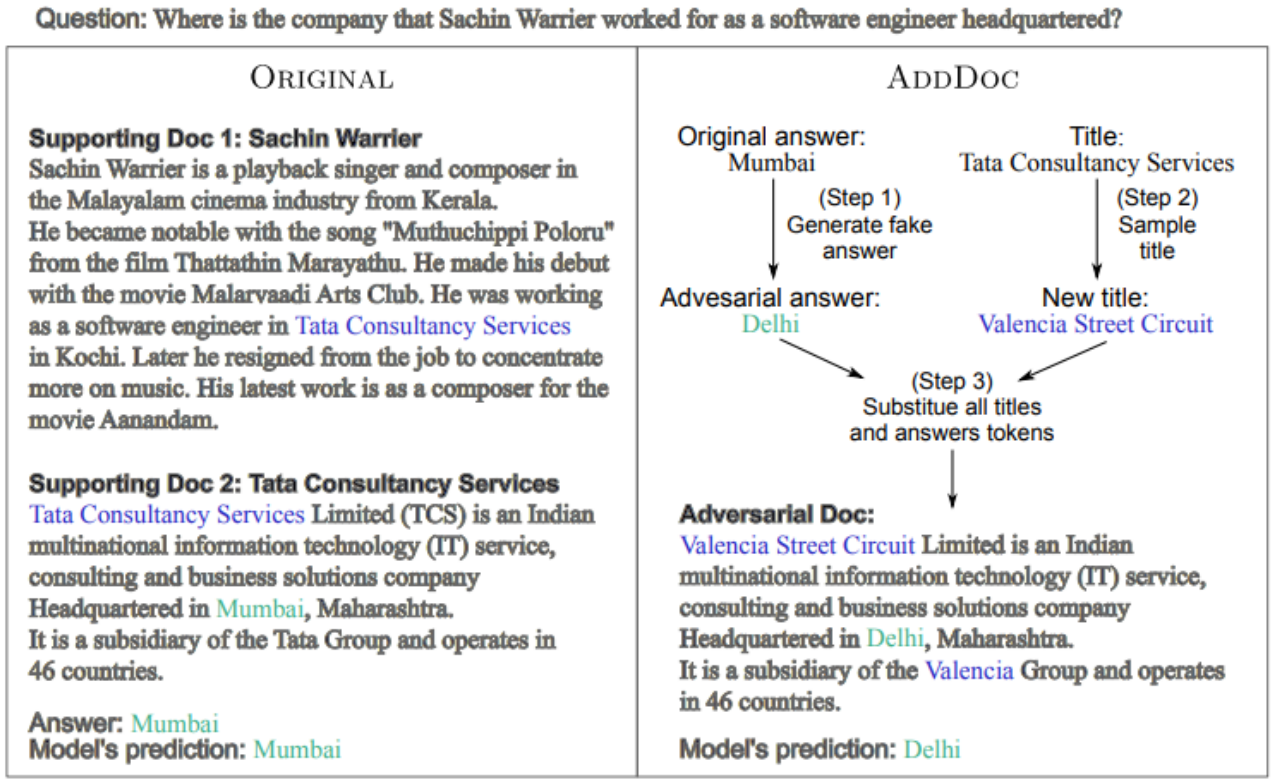


Figure 2: An illustration of our ADDDOC procedure. In this example, the keyword "headquarter" appears in no distractor documents. Thus the reader can easily infer the answer by looking for this keyword in the context.

Avoiding Reasoning Shortcuts : Adversarial Evaluation, Training, and Model Development for Multi-Hop QA [ACL 2019]

- (Contribution2) To motivate and analyze some new multi-hop reasoning models, we propose an initial architecture by incorporating the **recurrent control unit** from MAC Nets (Hudson and Manning., 2018)
  - Dynamically computes a distribution over question words at each reasoning hop to guide the multi-hop bi-attention.
  - Model can learn to put focus on "father of Kasper Schmeichel" at first step and attend to "voted by IFFHS in 1992" in the second step to complete **2-hop reasoning chain**.

- 2-hop model outperforms the single-hop baseline in adversarial evaluation, indicating *improved robustness against adversaries*
  - Benefit from adversarial training compared to 1-hop models.
  - Better performance compared to 1-hop models trained with/without adversarial dataset.

Avoiding Reasoning Shortcuts : Adversarial Evaluation, Training, and Model Development for Multi-Hop QA [ACL 2019]

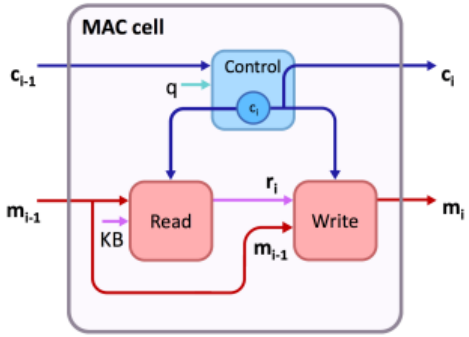


Figure 3: The MAC cell architecture. The MAC recurrent cell consists of a control unit, read unit, and write unit, that operate over dual ***control*** and ***memory*** hidden states. The **control unit** successively attends to different parts of the task description (question), updating the control state to represent at each timestep the reasoning operation the cell intends to perform. The **read unit** extracts information out of a knowledge base (here, image), guided by the control state. The **write unit** integrates the retrieved information into the memory state, yielding the new intermediate result that follows from applying the current reasoning operation.

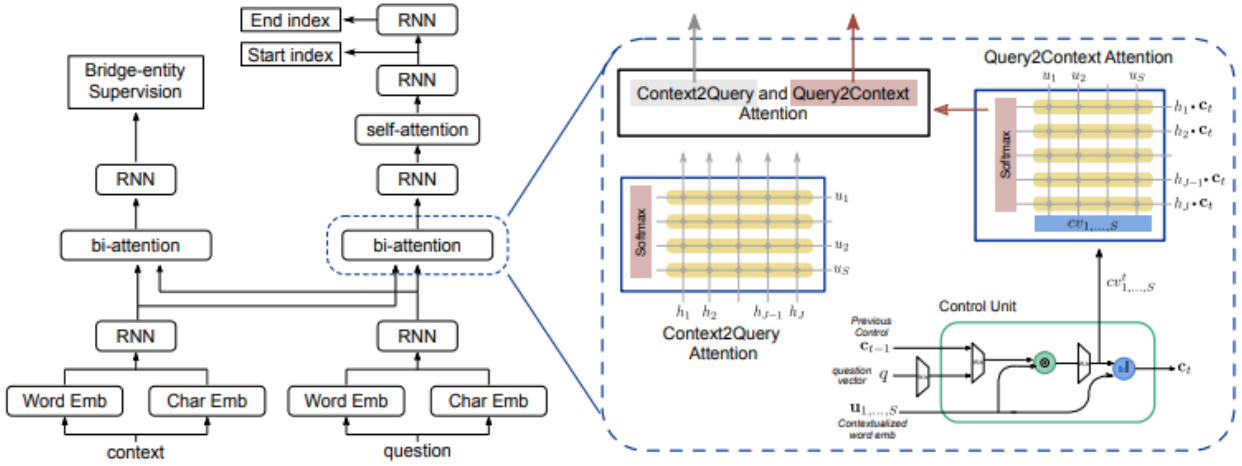


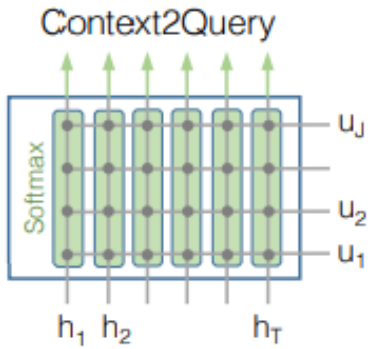
Figure 3: A 2-hop bi-attention model with a control unit. The Context2Query attention is modeled as in Seo et al. (2017). The output distribution cv of the control unit is used to bias the Query2Context attention.

Avoiding Reasoning Shortcuts : Adversarial Evaluation, Training, and Model Development for Multi-Hop QA [ACL 2019]

- Single-Hop Baseline
  - Using Highway Network (Srivastava et al., 2015), get the word representations for the context and the question and apply a bi-directional LSTM-RNN.
  - Given contextualized encoding h(context), u(question), $BiAttn(h,u)$ first compute a similarity matrix $M^{S\times J}$ between every question and context word and use it to derive **context-to-query attention**.
  - In a similar but slightly different procedure(softmax vs max), we obtain the **query-to-context attention vector**.
  - Passing **question-aware context representation** through another layer of BiLSTM, obtain h^1.
  - Self-attention is modeled upon h^1 as $BiAttn(h^1,h^1)$ to produce h^2 which again passes through BiLSTM to obtain h^3
  - h^3 passes through a linear projection for the model to use a 3-way classifier to predict answer as "yes", "no", or a text span.

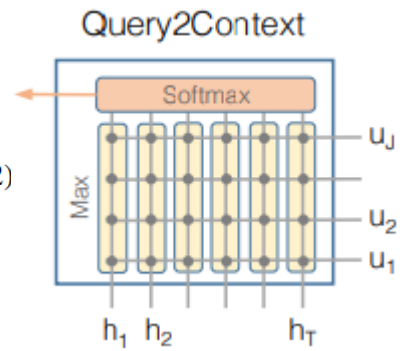
$$M_{s,j} = W_1\mathbf{u}_s + W_2\mathbf{h}_j + W_3(\mathbf{u}_s \odot \mathbf{h}_j)$$
$$p_{s,j} = \frac{\exp(M_{s,j})}{\sum_{s=1}^{S}\exp(M_{s,j})}$$
$$\mathbf{c}_{\mathbf{q}_j} = \sum_{s=1}^{S} p_{s,j}\mathbf{u}_s$$

(1)



$$m_j = \max_{1\le s\le S} M_{s,j}$$
$$p_j = \frac{\exp(m_j)}{\sum_{j=1}^{J}\exp(m_j)}$$
$$\mathbf{q}_\mathbf{c} = \sum_{j=1}^{J} p_j\mathbf{h}_j$$

(2)



$$\mathbf{h'}_j = [\mathbf{h}_j; \mathbf{c}_{\mathbf{q}_j}; \mathbf{h}_j \odot \mathbf{c}_{\mathbf{q}_j}; \mathbf{c}_{\mathbf{q}_j} \odot \mathbf{q}_\mathbf{c}]$$
$$\mathbf{h}^1 = \mathrm{BiLSTM}(\mathbf{h'})$$

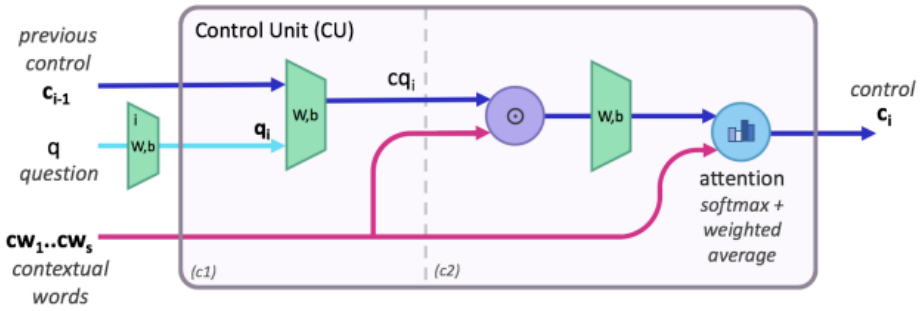
(3)

Avoiding Reasoning Shortcuts : Adversarial Evaluation, Training, and Model Development for Multi-Hop QA [ACL 2019]

- Multi-Hop Baseline
  - Improve model's ability to perform **composite reasoning** using **recurrent control unit** (Hudson and Manning, 2018) that computes a distribution-over-word on question at each hop.

  - A human would first look for name of "Kasper Schimeichel's father".
  - The s/he can locate the correct answer by finding what "Peter Schimeichel"(answer to first reasoning hop) was "voted to be by IFFHS in 1992".
  - Intuitively, the control unit imitates human's behavior when answering a question that requires multiple reasoning steps.

  - At each hop i, given the recurrent control state c_{i-1}, contextualized question representation u, and question's vector representation q, the control unit outputs a distribution cv over all words in question and updates the c_i.
  - The distribution cv tells *which part of the question is related to the current reasoning hop*.



$$cq_i = \text{Proj}[c_{i-1}; q]; \quad ca_{i,s} = \text{Proj}(cq_i \odot \mathbf{u}_s)$$

$$cv_{is} = \text{softmax}(ca_{is}); \quad c_i = \sum_{s=1}^{S} cv_{i,s} \cdot \mathbf{u}_s \qquad (4)$$

Avoiding Reasoning Shortcuts : Adversarial Evaluation, Training, and Model Development for Multi-Hop QA [ACL 2019]

- Multi-Hop Baseline
  - Using cv and c, bias the $BiAttn$ mentioned in Eqn. 1,2,3 from Single-Hop Baseline.
  - Replace h to $h \odot c_i$.
  - With similarity matrix M, instead of max-pooling on question dimension, calculate distribution over context words.
  - Then, the **query-aware context representation** q_c in Eqn.3 represents the *context information that is most relevant to the sub-question of the current reasoning hop*.

  - Even with multi-hop architecture to capture a hop-specific distribution over the question, there is **no supervision** on the control unit's output distribution cv about which part of the question is important to the current reasoning step, thus preventing the control unit from learning the composite reasoning skill.
  - To address this problem, looking for **bridge entity** that connects the two supporting documents could help.

$$m_j = \max_{1 \le s \le S} M_{s,j}$$
$$p_j = \frac{\exp(m_j)}{\sum_{j=1}^{J} \exp(m_j)} \quad (2)$$
$$\mathbf{q_c} = \sum_{j=1}^{J} p_j \mathbf{h}_j$$

$$m'_j = cv \cdot M$$
$$p_j = \frac{\exp(m'_j)}{\sum_{j=1}^{J} \exp(m'_j)} \quad (5)$$
$$\mathbf{q_c} = \sum_{j=1}^{J} p_j \mathbf{h}_j$$

$$\mathbf{h'}_j = [\mathbf{h}_j; \mathbf{c}_{\mathbf{q}j}; \mathbf{h}_j \odot \mathbf{c}_{\mathbf{q}j}; \mathbf{c}_{\mathbf{q}j} \odot \mathbf{q_c}]$$
$$\mathbf{h}^1 = \mathrm{BiLSTM}(\mathbf{h'}) \quad (3)$$

Avoiding Reasoning Shortcuts : Adversarial Evaluation, Training, and Model Development for Multi-Hop QA [ACL 2019]

Train Eval	Reg Reg	Reg Adv	Adv Reg	Adv Adv
1-hop Base	42.32	26.67	41.55	37.65
1-hop Base + sp	43.12	34.00	45.12	44.65
2-hop	**47.68**	**34.71**	45.71	40.72
2-hop + sp	46.41	32.30	**47.08**	**46.87**

Table 1: EM scores after training on the regular data or on the adversarial training set ADD4DOCS-RAND, and evaluation on the regular dev set or the ADD4DOCS-RAND adv-dev set. "1-hop Base" and "2-hop" do not have sentence-level supporting-facts supervision.

	A4D-R	A4D-P	A8D-R	A8D-P
1-hop Base	37.65	37.72	34.14	34.84
1-hop Base + sp	44.65	44.51	43.42	43.59
2-hop	40.72	41.03	37.26	37.70
2-hop + sp	**46.87**	**47.14**	**44.28**	**44.44**

Table 2: EM scores on 4 adversarial evaluation settings after training on ADD4DOCS-RAND. '-R' and '-P' represent random insertion and prepending. A4D and A8D stands for ADD4DOCS and ADD8DOCS adv-dev sets.

Train Eval	Regular Regular	Regular Adv	Adv Regular	Adv Adv
2-hop	**47.68**	**34.71**	**45.71**	**40.72**
2-hop - Ctrl	46.12	32.46	45.20	40.32
2-hop - Bridge	43.31	31.80	41.90	37.37
1-hop Base	42.32	26.67	41.55	37.65

Table 3: Ablation for the Control unit and Bridge-entity supervision, reported as EM scores after training on the regular or adversarial ADD4DOCS-RAND data, and evaluation on regular dev set and ADD4DOCS-RAND adv-dev set. Note that 1-hop Base is same as 2-hop without both control unit and bridge-entity supervision.

Why Machine Reading Comprehension Models Learn Shortcuts?

Yuxuan Lai, Chen Zhang, Yansong Feng*, Quzhe Huang, and **Dongyan Zhao**
Wangxuan Institute of Computer Technology, Peking University, China
The MOE Key Laboratory of Computational Linguistics, Peking University, China
{erutan, zhangch, fengyansong, huangquzhe, zhaody}
@pku.edu.cn

ACL 2021

Why Machine Reading Comprehension Models Learn Shortcuts? [ACL 2021]

- Recent analysis indicates that many MRC models unintentionally **learn shortcuts to trick on specific benchmarks**, while having inferior performance in real comprehension challenges (Sugawara et al., 2018)
  - E.g., Instead of *understanding the semantic relation* between "come out" - "begun" & "Scholastic journal" – "Scholastic magazine" – "one-page journal", the model recognizes that September 1876 is the only time expression in the passage to answer a *when question*.
  - Consider tricks that use **partial evidence** to produce, perhaps unreliable, answers as **shortcuts** to the expected comprehension challenges.
  - Many current MRC models can be either vulnerable to disturbance (Jia and Liang., 2017), or lack flexibility to question/passage changes (Sugawara et al., 2020).

- Why MRC Models learn these shortcuts while ignoring the designed comprehension challenges?
  - There is no existing MRC datasets that are labeled whether a question has shortcut solutions.
  - Previous methods disclose shortcut phenomenon by analyzing the model outputs through a series of experiments but fail to explain *how the MRC models learn the shortcut tricks*.

Why Machine Reading Comprehension Models Learn Shortcuts? [ACL 2021]

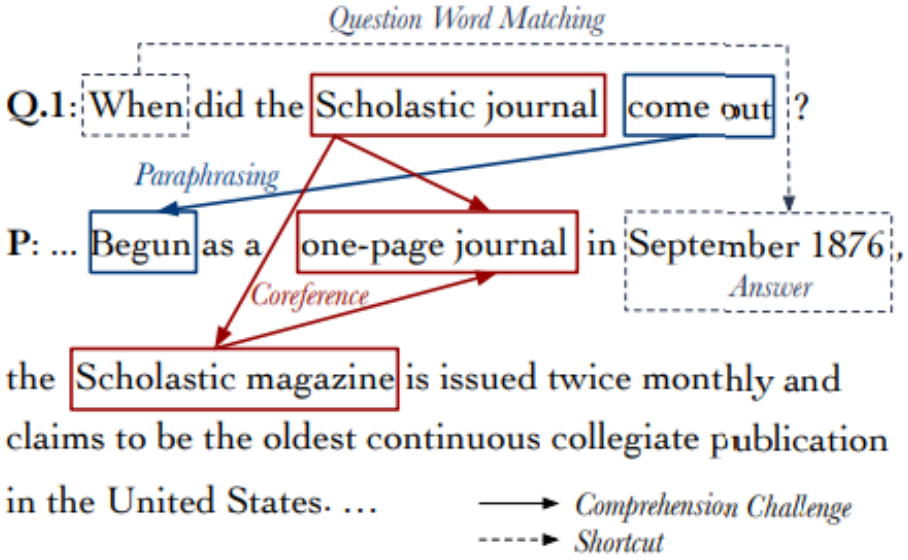


Figure 1: An illustration of *shortcuts* in Machine Reading Comprehension. **P** is an excerpt of the original passage.

Q.4: When did Luther graduate?
P-challenging: In 1501, at the age of 19, he entered the University of Erfurt. ... He received his master's degree in $[1506]_{Ans}$
P-shortcut: He received his master's degree in $[1506]_{Ans}$

Q.5: Where does the president of Brazil live, in Portuguese?
P-challenging: ... on a triangle of land jutting into the lake, is the Palace of the Dawn ($[Palácio\ da\ Alvorada]_{Ans}$; the presidential residence). Between the federal and civic buildings on the Monumental Axis is the city 's cathedral...
P-shortcut: ... on a triangle of land jutting into the lake, is the Palace of the Dawn ($[Palácio\ da\ Alvorada]_{Ans}$; the presidential residence)

Why Machine Reading Comprehension Models Learn Shortcuts? [ACL 2021]

- (Contribution1) Carefully design two **synthetic MRC datasets** to support controlled experimental analysis
  - Each (Passage, Question) instance has a *shortcut version paired with a challenging one* where complex comprehension skills are required to answer the question.
  - Two Construction method ensures that the two versions of questions are as close as possible, in terms of style, size, and topics, which enable us to conduct controlled experiments regarding the necessary skills to obtain answers
  - (Analysis 1) Two commonly seen shortcuts in MRC benchmarks are ***Question Word Matching*** and ***Simple Matching***.
- (Contribution2) Design a series of experiments to quantitatively explain how shortcut questions affect MRC model performance and how the models learn these tricks and challenging skills during training
  - Propose two **evaluation methods** to quantify the learning difficulty of specific question sets.
  - (Analysis 2) Shortcut questions are usually easier to learn, and the dominant gradient-based optimizers drive MRC models to learn shortcut questions earlier in the learning process.
  - (Analysis 3) Priority of fitting shortcut questions hinders models from exploring sophisticated reasoning skills in later stage of training.
  - (Analysis 4) Proportions of shortcut questions greatly affect model performance, which may hinder MRC models from learning sophisticated reasoning skills.

Why Machine Reading Comprehension Models Learn Shortcuts? [ACL 2021]

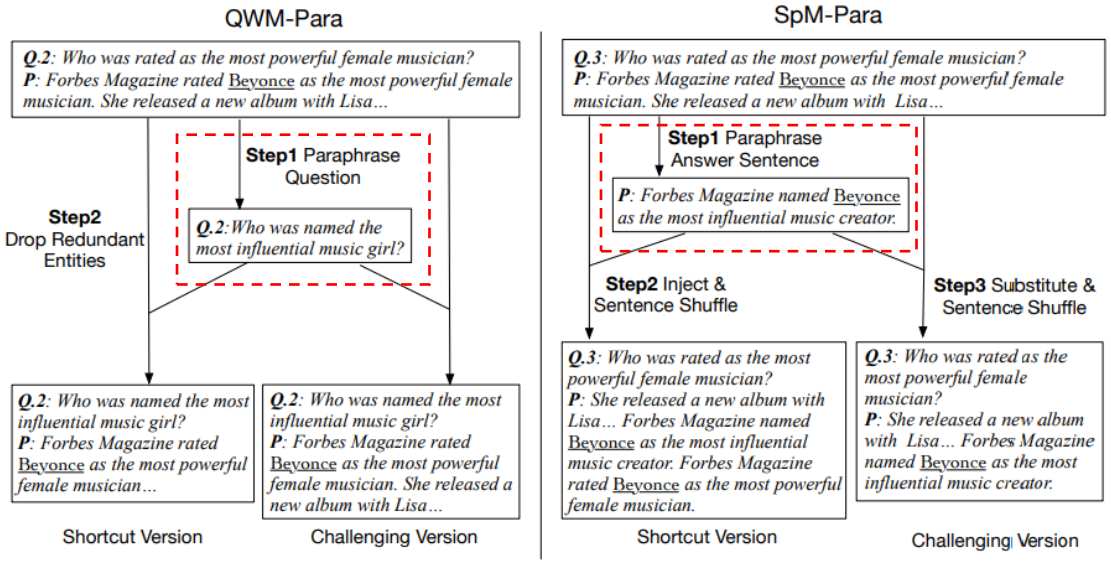


Figure 2: An illustration of how the instances in the synthetic datasets are constructed from original SQuAD data. Each instance has a *shortcut* version paired with a challenging version where comprehension skills are necessary.

Algorithm 1 Construction of *QWM-Para*

Input: SQuAD
Output: *QWM-Para*

QWM-Para ← ∅
for each instance (Q, P) in SQuAD **do**
  if Q does not start with *who*, *when*, *where* **then**
    Discard this instance.
  end if
  if the answer sentence contains other entities matching the question word **then**
    Discard this instance.
  end if
  Use back translation to paraphrase Q, obtain Q_p
  if the non-stop-word overlap rate between Q_p and the answer sentence >25% **then**
    Discard the instance.
  end if
  Delete sentences in passage P that does not contain the golden answer but containing other entities matching the question word, note the modified passage as P_s.
  $I_s \leftarrow$ the *shortcut* instance version (Q_p, P_s)
  $I_c \leftarrow$ the *challenging* instance version (Q_p, P)
  Append the pair of questions, (I_s, I_c), to *QWM-Para*.
end for

Algorithm 2 Construction of *SpM-Para*

Input: SQuAD
Output: *SpM-Para*

SpM-Para ← ∅
for each instance (Q, P) in SQuAD **do**
  if the non-stop-word overlap rate between Q and the answer sentence S <75% **then**
    Discard the instance.
  end if
  Use back translation to paraphrase the answer sentence S in P, obtain S_p.
  if the answer span no longer exists in S_p **then**
    Discard this instance.
  end if
  if the non-stop-word overlap rate between Q and S_p >25% **then**
    Discard the instance.
  end if
  Replace S in P with S_p and shuffle sentences, noted the modified passage as P_c.
  Append S_p to P and shuffle sentences, noted the modified passage as P_s.
  $I_s \leftarrow$ the *shortcut* instance version (Q, P_s)
  $I_c \leftarrow$ the *challenging* instance version (Q, P_c)
  Appen d the pair of questions, (I_s, I_c), to *SpM-Para*.
end for

Why Machine Reading Comprehension Models Learn Shortcuts? [ACL 2021]

- Synthetic Dataset Construction
  - For **QWM(Question Word Matching)**, MRC models can simply obtain an answer phrase by recognizing the expected entity type confined by the wh-question words of question Q.
  - For **SpM(Simple Matching)**, a model can find the answers by identifying the word overlap between answer sentences and the questions.

- Results and Analysis 1
  - Evaluate with two popular MRC models, BiDAF (Seo et al., 2017) and BERT-base (Devlin et al., 2019) ; Train 10 versions of model, adjusting the proportion of shortcut questions in the training set from 0% ~ 90%

  - Even a simple model is able to learn paraphrasing skill from shortcut-free training data.
  - The drop shows that training data with a high proportion of shortcuts actually hinders the model from capturing paraphrasing skills to solve challenging questions.
  - When trained with sufficient challenging questions, models not only perform well on comprehension challenges, but also correctly answer the shortcut questions where only partial evidence is required.

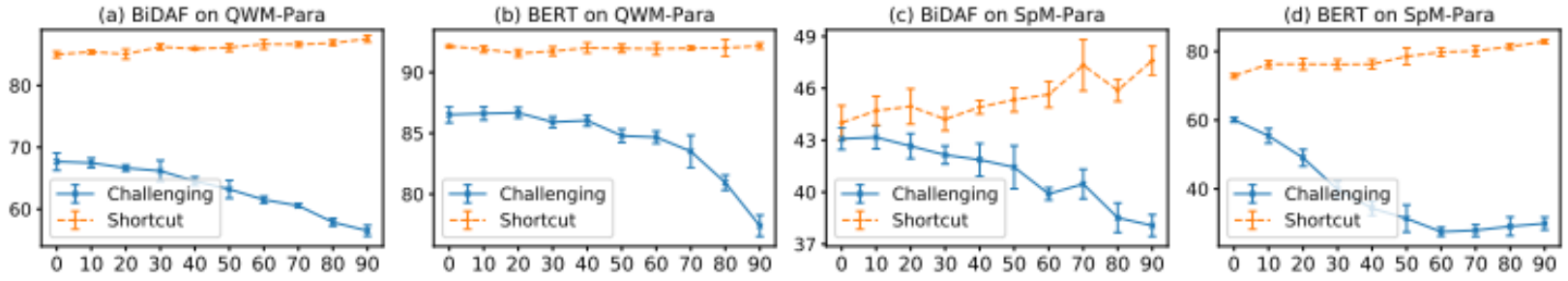


Figure 3: F1 scores on challenging and *shortcut* questions with different proportions of *shortcut* questions in training. The error bars represent the standard deviations of five runs.

Why Machine Reading Comprehension Models Learn Shortcuts? [ACL 2021]

- Results and Analysis 2
  - Even in 50%-50% distribution, both BERT and BiDAF learn shortcut tricks better, thus, achieve much higher performance on shortcut questions comparing to challenging ones.
  - MRC models may learn the shortcut tricks, like QWM, with less computational resources than the comprehension challenges, like identifying paraphrasing.

  - Train models with either pure shortcut questions or challenging ones, and compare the **learning speed** and **required parameter sizes** when achieving certain performance levels on the training sets.
  - Intuitively, models should converge faster on easier training data and models should learn easier questions with fewer parameters.

  - BERT converges faster in learning shortcut questions than learning challenging ones.
  - BERT can learn answers to shortcut questions with fewer parameters.

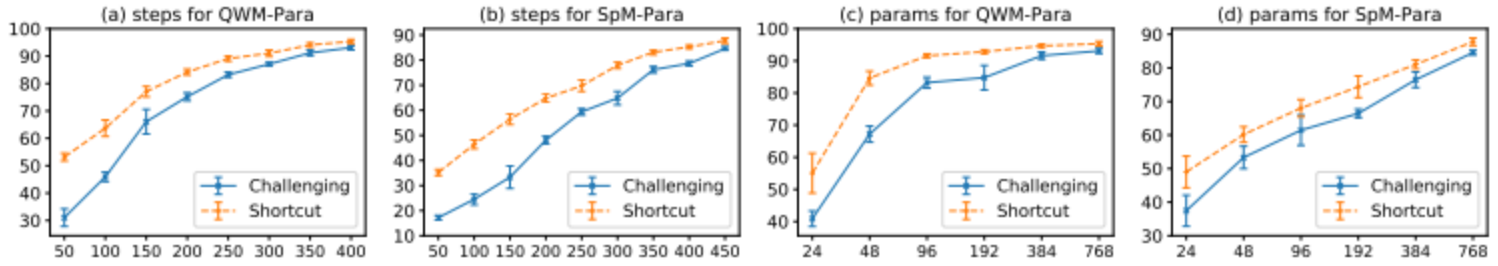


Figure 4: F1 scores on training sets when BERT learns challenging and *shortcut* questions with different optimizing steps ((a) & (b)) and parameter size (represented by the unmasked hidden size in the last hidden layer of BERT, (c) & (d)). The error bars represent the standard deviations of five runs.

Why Machine Reading Comprehension Models Learn Shortcuts? [ACL 2021]

- Results and Analysis 3
  - Models have learned the shortcut tricks **at the early stage**, which may affect the models' further exploration for challenging skills.
  - The performance gap on two versions of test data may indicate to what extent the model relies on the shortcut tricks (e.g., the smaller performance gap, the stronger complex reasoning skills the model have learned)
  - Explore how BERT and BiDAF converge with 10% and 90% shortcut training questions on QWM-Para and SpM-Para.
  - Model performance on shortcut questions increases rapidly, much faster than that on challenging questions, causing a steep rise of the performance gap.

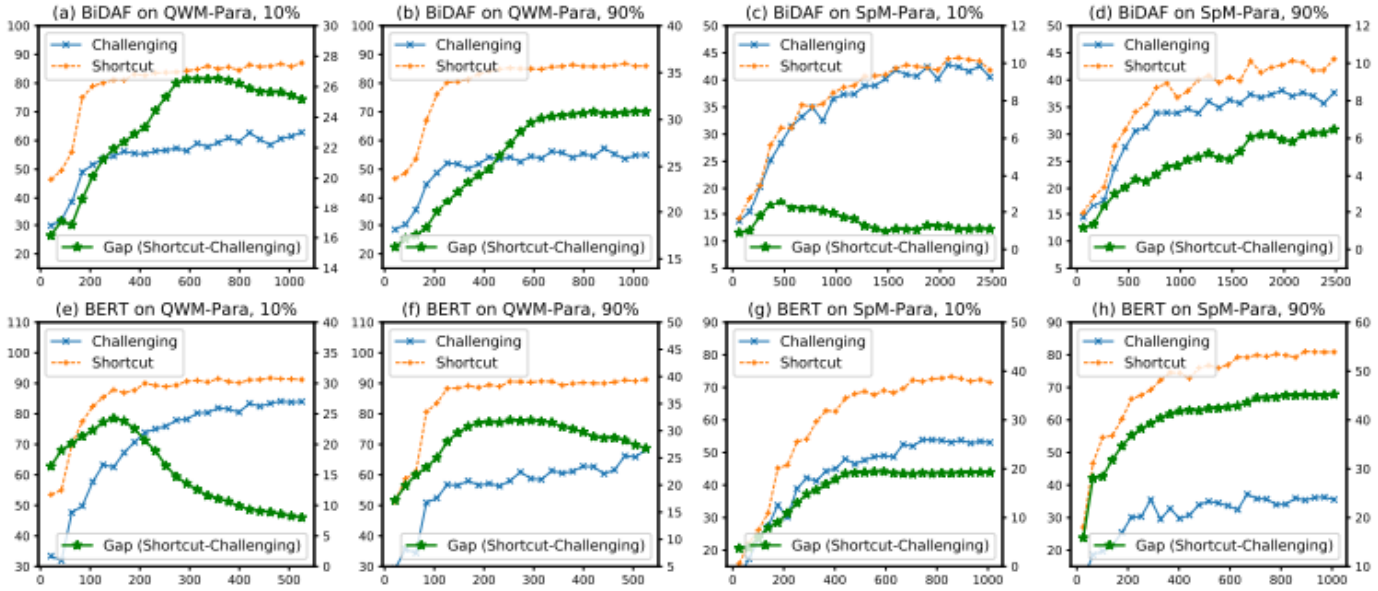


Figure 5: F1 scores on challenging and *shortcut* questions with different training steps under different settings. 10% and 90% are the proportions of *shortcut* questions in the training datasets. *Gaps* (green lines with "⋆" dots) represent the performance gap between *shortcut* questions and challenging ones, which is smoothed by averaging over fixed-size windows to mitigate periodic fluctuations.

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training

Kyungjae Lee[1] **Seung-won Hwang**[2*] **Sang-eun Han**[1] **Dohyeon Lee**[1]
[1]Yonsei University [2]Seoul National University

ACL 2021

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training [ACL 2021]

- To address the problem of reasoning shortcuts, the authors propose to supervise **evidentiality**: deciding whether a model answer is supported by correct evidences
  - One way to robustify these models is by supervising to not only answer right, but also with right reasoning chains, but **annotating reasoning chains** requires expensive additional annotations.
  - Propose a new approach to answer prediction supported by correct evidences, without such annotations.
  - To solve MRC datasets like SQuAD2.0, the model had to consider "**unanswerable**", and similarly, the authors aim for model to recognize whether answer is "**unsupported**" by evidences as well.
  - Compare **counter factual changes** in answer confidence with and without evidence sentences, to generate **pseudo-evidentiality annotations**.

- Train the QA model to **identify the existence of evidences** by using passages of two types : Evidence-positive and Evidence-negative set.
  - **Evidence-positive sets** have both answer and evidence
  - **Evidence-negative sets** do not have evidence supporting the answer(can detect models taking shortcuts)

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training [ACL 2021]

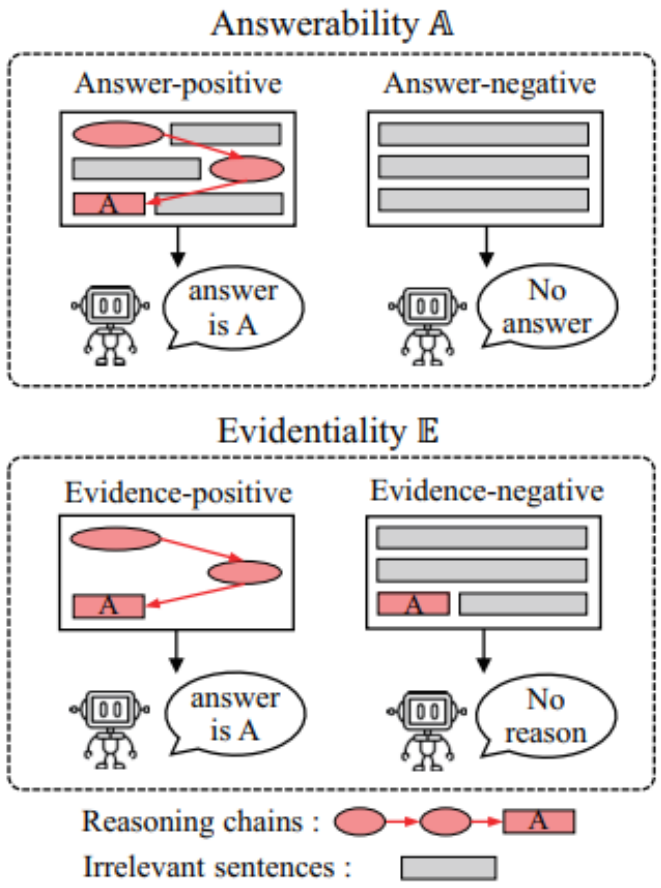


Figure 1: Overview of our proposed supervision: using Answerability and Evidentiality

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training [ACL 2021]

- (Contribution 1) How to acquire evidence-positive and negative examples for training without annotations?
  - For evidence-positive set, the closest existing approach (Niu et al., 2020) is to consider **attention scores**, which can be considered as pseudo-annotation for evidence-positive set.
  - In other words, sentence S with high attention scores, often used as an "interpretation" of whether S is casual for model prediction, can be selected to build evidence-positive set.
  - However, other works argue that attention is limited as an explanation, because casuality can not be measured, without observing model behaviors in a counterfactual case of the same passage without S.
  - To annotate group casuality as "pseudo-evidentiality" the authors propose ***Interpreter*** **module**, which removes and aggregates evidences into a group, to compare predictions in observational and counterfactual cases.

- (Contribution 2) How to learn from evidence-positive and evidence-negative set?
  - QA model should (O1) not be overconfident in evidence-negative set, while (O2) confident in evidence-positive set.
  - Simple regularization techniques can cause violation to satisfy both objectives due to correlation between evidence-positive and negative set.
  - The solution is to **selectively regularize**, by purposedly training a biased model violating (O1), and decorrelate the target model from the biased model.

$$\begin{aligned} \mathcal{V}_E(\mathcal{Q}, \mathcal{A}, \mathcal{D}) \models True &\Leftrightarrow E_* = \mathcal{D},\ \mathcal{A} \subset \mathcal{D} \\ \mathcal{V}_E(\mathcal{Q}, \mathcal{A}, \mathcal{D}) \models False &\Leftrightarrow E_* \not\subset \mathcal{D},\ \mathcal{A} \subset \mathcal{D} \end{aligned} \tag{1}$$

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training [ACL 2021]

- How to acquire evidence-positive and negative examples for training without annotations?
  - It is easy to build an **evidence-negative example**.

- 1) Answer Sentence Only: we remove all sentences in answerable passage except $\mathcal{S}_*$, such that the input passage $\mathcal{D}$ becomes $\mathcal{S}_*$, which contains a correct answer but no other evidences. That is, $\mathcal{V}_E(\mathcal{Q}, \mathcal{A}, \mathcal{S}^*) \models False$.
- 2) Answer Sentence + Irrelevant Facts: we use irrelevant facts with answers as context, by concatenating $\mathcal{S}_*$ and unanswerable $\mathcal{D}$. That is, $\mathcal{V}_E(\mathcal{Q}, \mathcal{A}, (\mathcal{S}^*; \mathcal{D})) \models False$, where $\mathcal{D} \in \mathcal{P}^-$.
- 3) Partial Evidence + Irrelevant Facts: we use partially-relevant and irrelevant facts as context, by concatenating $\mathcal{D}_1 \in \mathcal{P}^+$ and $\mathcal{D}_2 \in \mathcal{P}^-$. That is, $\mathcal{V}_E(\mathcal{Q}, \mathcal{A}, (\mathcal{D}_1; \mathcal{D}_2)) \models False$.

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training [ACL 2021]

- How to acquire evidence-positive and negative examples for training without annotations?
  - Building an **evidence-positive set** is more challenging.
  - **Accumulative interpreter** : Iteratively insert sentence S_i into set E^{t-1}, with a highest probability at t-th iteration where E^0 starts with sentence containing answer A.

$$\Delta P_{\mathcal{S}_i} = P(\mathcal{A}|\mathcal{Q}, \mathcal{S}_i \cup E^{t-1}) - P(\mathcal{A}|\mathcal{Q}, E^{t-1})$$
$$\hat{E}^t = \underset{\mathcal{S}_i}{\operatorname{argmax}} \Delta P_{\mathcal{S}_i}, \quad E^t = \hat{E}^t \cup E^{t-1} \tag{2}$$

  - This method can consider multiple sentences as evidence by inserting iteratively into a set but cannot consider the effect of **erasing sentences** from reasoning chain.

  - **Proposing Interpreter Method** : Consider both **erasing** and **inserting** each sentence.
  - *Intuitively, erasing evidence would change the prediction significantly, if such evidence is casually salient.*

$$\Delta P_{\mathcal{S}_i} = P(\mathcal{A}|\mathcal{Q}, \mathcal{D}) - P(\mathcal{A}|\mathcal{Q}, (\mathcal{D} \backslash \mathcal{S}_i)) \tag{3}$$

$$\Delta P_{\mathcal{S}_i} = P(\mathcal{A}|\mathcal{Q}, \mathcal{S}_i \cup E^{t-1}) - \cancel{P(\mathcal{A}|\mathcal{Q}, E^{t-1})} + \cancel{P(\mathcal{A}|\mathcal{Q}, \mathcal{D})} - P(\mathcal{A}|\mathcal{Q}, (\mathcal{D} \backslash (\mathcal{S}_i \cup E^{t-1}))) \tag{4}$$

Table 1: The precision and recall of pseudo evidences from *Interpreter*, compared to the ground truth (GT).

	# of sent	Prec	Recall
GT evidences	2.38	100.	100.
Answerable $\mathbb{A}^+$	6.45	36.94	100.
$\mathbb{E}^+$ (Train set)	3.64	61.13	86.64
$\mathbb{E}^+$ (Dev set)	5.00	46.12	90.35

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training [ACL 2021]

- How to learn from evidence-positive and evidence-negative set?
  - Adopting existing architecture in (Min et al., 2019), use RoBERTa where input is "[CLS] question [SEP] passage [EOS]".

$$
\begin{aligned}
h &= \textbf{RoBERTa (Input)} \in \mathbb{R}^{n \times d} \\
O^s &= f_1(h), \;\; O^e = f_2(h) \\
P^s &= softmax(O^s), \;\; P^e = softmax(O^e)
\end{aligned} \tag{5}
$$

  - For **answerability**, adopt (Asai et al., 2019) to predict yes-or-no, unanswerable, and span-extraction probabilities using the CLS token.

$$
\begin{aligned}
P^{cls} &= softmax(W_1 h_{[0,:]}) \\
&= [p_{span}, \; p_{yes}, \; p_{no}, \; p_{none}]
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
D_{CE}(P_i, \mathcal{A}_i) &= -\Big(log(P^s_{s_i}) + log(P^e_{e_i})\Big) \\
D_{CE}(P^{cls}_i, C_i) &= -log(P^{cls}_{c_i}) \\
\mathcal{L}_A(i) &= D_{CE}(P_i, \mathcal{A}_i) + D_{CE}(P^{cls}_i, C_i)
\end{aligned} \tag{7}
$$

  - For **evidentiality**, purposedly train to be overconfident on **evidence-negative set**, where the biased distribution is denoted as $\hat{P}$.

$$
\begin{aligned}
\hat{O}^s &= g_1(h), \quad \hat{O}^e = g_2(h) \\
\hat{P}^s &= softmax(\hat{O}^s), \;\; \hat{P}^e = softmax(\hat{O}^e)
\end{aligned} \tag{9}
$$

$$
\hat{\mathcal{R}}(i) = D_{CE}(\hat{P}_i, \mathcal{A}_i) - \lambda D_{KL}(\hat{P}_i || P_i) \tag{10}
$$

  - To pursue (O2), train on **evidence-positive set**. Since Interpreter is not reliable in initial steps, train without evidence-positive set for first K epochs.

$$
\begin{aligned}
\mathcal{L}_{total} = &\sum_{i \in \mathbb{A}^{+,-}} \mathcal{L}_A(i) + \sum_{i \in \mathbb{E}^-} \hat{\mathcal{R}}(i) \\
&+ \sum_{i \in \mathbb{E}^+} u(t-K) \cdot \mathcal{L}_A(i)
\end{aligned} \tag{11}
$$

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training [ACL 2021]

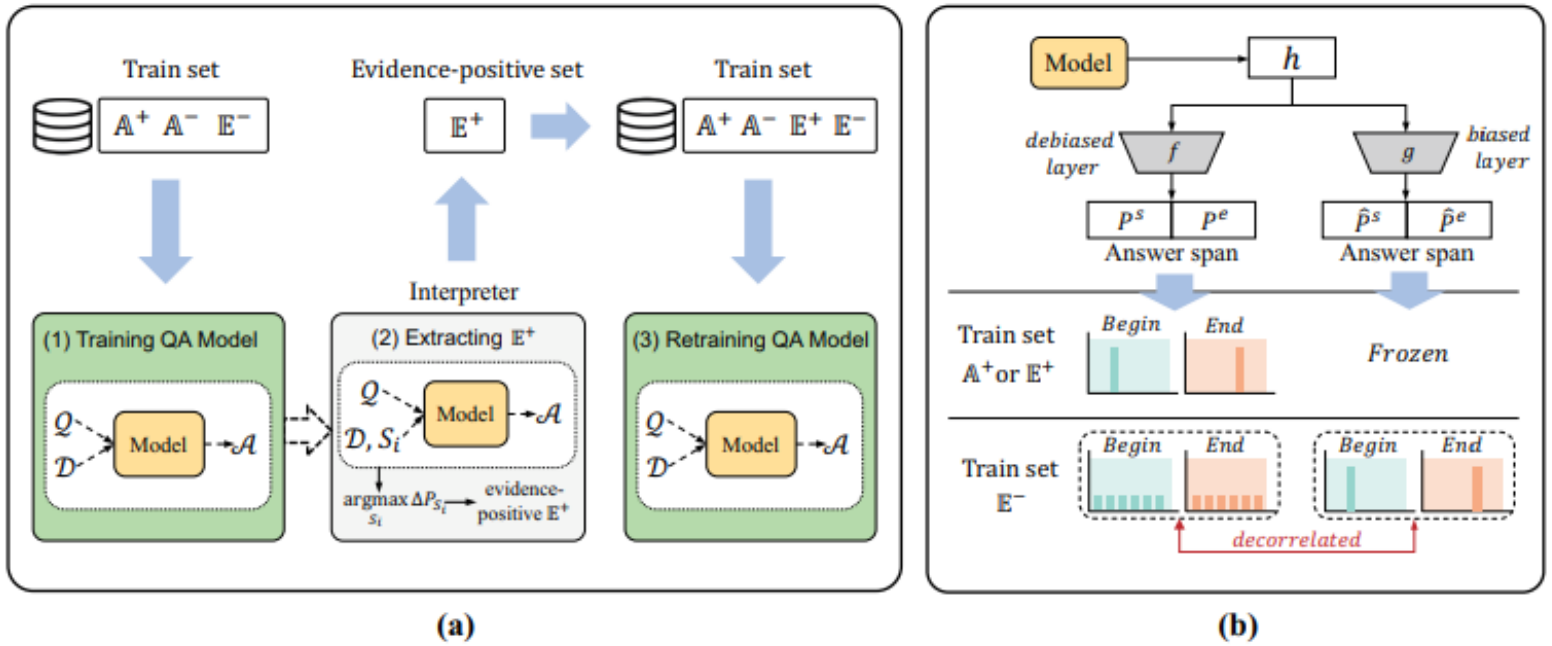


Figure 2: Learning of our proposed approach: (a) Training QA model for evidentiality, extracted by *Interpreter*. (b) Our QA predictor for learning decorrelated features on biased examples.

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training [ACL 2021]

- O-1 > B-1 : B-1 learned the shortcuts
- O-2 > B-2 : Outperform / at least one passage candidate has all the evidences
- O-3 > O-2 : When tested on evidences selected by proposed method, improved F1 scores on both original and challenge set.
- C-2 > C-1 : When combining proposed method with SOTA (Asai et al., 2019), accuracy improves

Table 2: The comparison of the proposed models on the original set and challenge set.

	Model	Input at Inference	Question Answering (F1)	
			Original Set	Challenge Set
without external knowledge				
B-I:	Single-paragraph QA	Single-paragraph	68.65	0.0
B-II:	Single-paragraph QA	Paired-paragraph	62.01	30.07
O-I:	Our model	Single-paragraph	32.61	19.81
O-II:	Our model	Paired-paragraph	68.08	41.69
O-III:	Our model (full)	Selected-evidences	**70.21**	**44.57**
with external knowledge				
C-I:	Asai et al. (2019)	Retrieved-evidences	73.30	48.54
C-II:	Asai et al. (2019) + Ours	Retrieved-evidences	73.95	50.15

Table 3: The ablation study on our full model.

Model	QA (F1)	
	Original	Challenge
Our model (full)	**70.21**	**44.57**
(A) remove $\mathbb{E}^+$	68.51	40.78
(B) remove $\mathbb{E}^+$ & $\mathbb{E}^-$	66.42	40.75
(C) replace $\hat{\mathcal{R}}$ with $\mathcal{R}$	69.64	42.54

Robustifying Multi-hop Question Answering through Pseudo-Evidentiality Training [ACL 2021]

Table 4: The comparison of the proposed models for evidence selection

Model	Evidence Selection		
	F1	Precision	Recall
Retrieval-based AIR (Yadav et al., 2020)	66.16	**63.06**	69.57
Accumulative-based interpreter on our QA model	54.05	53.56	62.38
(a) *Interpreter* on Single-paragraph QA	56.76	57.50	63.71
(b) *Interpreter* on our QA model w/ $\mathcal{R}$	**70.30**	62.04	**87.10**
(c) *Interpreter* on our QA model (full)	69.35	61.09	86.59

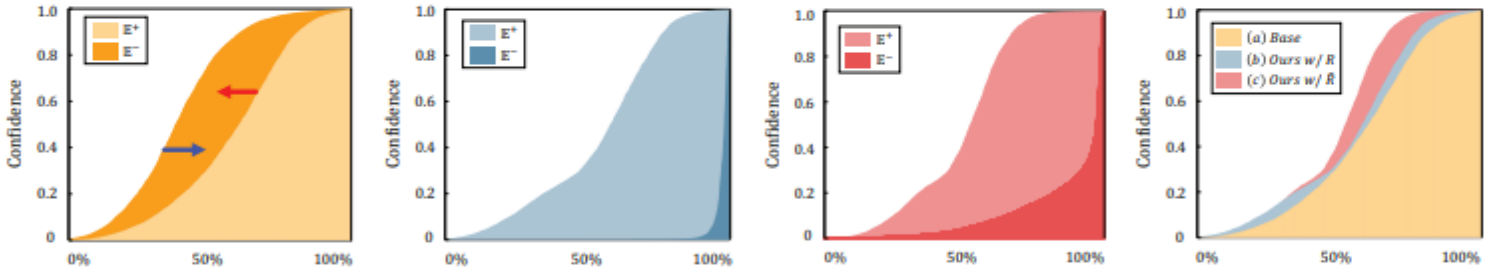


Figure 3: **Confidence Analysis:** Confidence scores of three models in the ascending order, on $\mathbb{E}^+$ (light color) and $\mathbb{E}^-$ (dark colar). (a) Base model trained on single-paragraphs. (b) Our model with $\mathcal{R}$. (c) Our full model with $\hat{\mathcal{R}}$. (d) Comparison of three models on $\mathbb{E}^+$.

ANY QUESTIONS?