DATA-DRIVEN NATURAL LANGUAGE INFERENCE

Yixin Nie

A thesis submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2022

Approved by:

Mohit Bansal

Snigdha Chaturvedi

Douwe Kiela

Marc Niethammer

Shahriar Nirjon

© 2022
Yixin Nie
ALL RIGHTS RESERVED

ABSTRACT

Yixin Nie: Data-Driven Natural Language Inference
(Under the direction of Mohit Bansal)

Natural Language Inference (NLI) research involves the development of models that can mimic human inference processes based on natural language and classify the inference relation between sentences. For example, given the premise that “In 2019, the Raptors won their first Eastern Conference title, and the team’s first NBA Finals”, it follows that “The Raptors beat another team in the 2019 NBA Finals”. but it does not follow that “The Golden State Warriors won the last game of the NBA Finals in 2019”. The goal of NLI is to build machines that can take pairs of premise and hypothesis as input and correctly predict the inference relation between them, that is reverse engineering the inference process of a human. NLI is a fundamental task with a simple and generic formalization such that NLI models can be practically useful in all kind of NLP applications. In recent years, there has been emerging interest and research in data-driven natural language inference.

This thesis starts with several key applications of data-driven NLI modules, including sentence-based NLI modeling, how to effectively use the NLI model as a key natural language understanding (NLU) module in both an automatic fact-checking system for claim verification and in an open-domain dialogue system for improving dialogue consistency. Empirical results not only demonstrate valuable use cases of NLI models in NLP applications but, more importantly, reveal the fact that the data is a key factor that contributes to the success of the usage of NLI models. That leads to the second part of this thesis, namely, adversarial NLI, a research endeavor that embodies dynamic human-and-model-in-the-loop learning paradigm for NLI via competitive iterations between model training and crowd-sourcing to push the limit of NLU.

To my parents and my grandparents.

ACKNOWLEDGEMENTS

First and foremost I would like to express my appreciation to my advisor Professor Mohit Bansal who have supported me throughout my PhD. Mohit has been patient and has pushed me through my difficult times. He helped teach me to be a more socially responsible person and I know that will be something more important than just doing research.

I would like to thank Hao Tan, Licheng Yu, and Qiuyu Xiao, either as lab-mates or as friends. Discussion with them have always been so enlightening and delightful. I can still remember the moments when we chatted late at night about miscellaneous things which helped me went through the journey with up-and-downs.

I would like to thank Yicheng Wang, an wonderful co-author that not only helped me finish my very first published paper but also shared with me many writing skills and helped me shaped my own writing style. I know that how isolated and unproductive I can be, therefore, I sincerely thank Xiang Zhou and Shiyue Zhang for your valuable social support.

I have had many wonderful opportunities to work with a lot of amazing people through internships and collaborations. I would like to thank Emily Dinan and Adina Williams who have been extremely supportive at my Adversarial NLI project. I would like to thank Douwe Kiela and Jason Weston who has provided the mentorship that I can not be more grateful.

I thank Linjie Li, Zhe Gan, Shuohang Wang, Chenguang Zhu, Lijuan Wang from Microsoft that helped me with my very first vision-and-language pre-training project.

I would like to thank my committee members (Mohit Bansal, Snigdha Chaturvedi, Douwe Kiela, Marc Niethammer, Shahriar Nirjon) for their constant support and valuable feedback in completing this dissertation.

I thank my parents and my grandparents for their constant support and unconditional love.

I thank my friends Deyi Liu, Hao Jiang, Qian Chen, Jie Lei, Zhenling Xu, Zhengyang Shen, and too many to list for their accompany during my difficult times, and for their willingness enduring my complaints.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Understanding entailment and contradiction is fundamental to understanding natural language, and inference about entailment and contradiction is a valuable testing ground for the development of semantic representations. Given a premise sentence and a hypothesis sentence, the task of natural language inference (NLI) is to predict the inference relation between the two sentences. The relation can be "entailment" if the premise infers the hypothesis, or "contradiction" if the premise contradicts the hypothesis, or it can also be "neutral" if the premise can neither infer nor contradict the hypothesis. The introduction of large-scale natural language inference datasets, including the PASCAL Recognising Textual Entailment Challenge (RTE), Stanford Natural Language Inference (SNLI), and Multi-Domain Natural Language Inference (MNLI) (Dagan et al., 2005; Bowman et al., 2015a; Williams et al., 2018a) provides valuable fuel to facilitate the data-driven neural approach for learning NLI. These datasets have not only attracted NLP researchers to develop models that can do general natural language understanding which results in advancement in language representation learning but also served as testing sets for general-purpose NLU evaluation which helps to benchmark our current progress (Wang et al., 2018a). During the past five years, we have witnessed significant improvement on NLI modeling (Chen et al., 2017c; Nie and Bansal, 2017; Devlin et al., 2019a; Liu et al., 2019b) and widespread applications of NLI in other NLP tasks like automatic fact verification (Thorne et al., 2018b), automatic summarization (Maynez et al., 2020), machine translation (Poliak et al., 2018a), video-caption (Pasunuru and Bansal, 2017), and conversation (Nie et al., 2021).

The main content of this thesis can roughly be categorized into efforts on answering the following two questions: (1) How can we effectively use data-driven models in NLP applications?; (2) How can we understand NLP models' general understanding ability via NLI. For the first question, we present our past work on building a NLI model that achieved the top-simple model in the second workshop on evaluating vector space representations for NLP, an automatic fact-checking system that won first place in the fact extraction and verification challenge in EMNLP 2018. We also present our work on using natural language inference models to evaluate and improve the consistency of state-of-the-art open domain chatbots. For the second question, we present adversarial NLI, a research effort on rethinking benchmarking in NLP, crowdsourcing NLP datasets and initiating dynamic NLP benchmarks.

Sentence Encoder-based NLI. For NLI modeling, practitioners often approach NLI using end-to-end neural networks. The input sentence will be first converted into a list of vectors called token embeddings and then be fed into the neural networks. The output of the neural network is a 3-element vector produced by a softmax layer, indicating model predicted probability for entailment, neutral, and contradiction, respectively. The parameters of the neural network will be optimized by minimizing the cross-entropy loss between the predicted label and the ground truth label. For the neural networks, typical choices includes RNNs (Mikolov et al., 2010), LSTMs (Hochreiter and Schmidhuber, 1997), or Transformers (Vaswani et al., 2017). All of which can easily take a sequence of token embedding as inputs. Usually, we impose no constraints on the design of the neural networks. However, there is a specific type of model called the sentence encoder-based NLI model that is focused on encoding natural language sentences into vectors with the potential to produce a vector representing the general sentence meaning. The benefit of the design is that after training, the sentence encoder will be able to learn a more generic language representations that will be useful across many NLP tasks (Conneau et al., 2017b; Reimers and Gurevych, 2019). In this thesis, I proposed an LSTM-based sentence encoder for NLI modeling by stacking multiple layers of LSTM with shortcut connections. The resultant model achieved state-of-the-art performance on the two canonical NLI datasets (SNLI (Bowman et al., 2015a) and MNLI (Williams et al., 2018a)).

Automatic Fact Checking. This is the most intuitive application of NLI. The task itself can be also described in an existing NLP framework called Machine Reading at Scale (MRS). Machine

Reading at Scale (MRS) is a term initiated in (Chen et al., 2017a) to describe the task of language understanding by utilizing external textual knowledge sources. MRS has become popular ever since its birth because it promotes the idea of combined design of information retrieval and machine comprehension. The goal of Machine Reading at Scale has also been extended to build a system that can give satisfactory responses (such as QA or fact-checking) based on user queries by accessing and retrieving information from a pre-defined set of knowledge sources. Machine Reading at Scale is vital for various NLP applications like QA, and automatic fact-checking. Existing tasks and datasets for MRS includes automatic fact checking (Thorne et al., 2018b) and multi-hop extractive QA (Yang et al., 2018). Progress on MRS has been made by improving separately the upstream retrieval sub-modules and the downstream comprehension sub-modules with recent advancements on representation learning (Devlin et al., 2019a; Liu et al., 2019b). However, partially due to the lack of annotated data for intermediate retrieval in an MRS setting, the evaluations were done mainly on the final downstream task and with much less consideration for the intermediate retrieval performance. This led to the convention that upstream retrieval modules mostly focus on getting better coverage of the downstream information such that the upper bound of the downstream score can be improved, rather than finding more exact information. This convention is misaligned with the nature of MRS where equal effort should be put into emphasizing the models' joint performance and optimizing the relationship between the semantic retrieval and the downstream comprehension subtasks. To shed light on the importance of semantic retrieval for downstream comprehension tasks, we start by establishing a simple and effective hierarchical pipeline system for MRS using Wikipedia as the external knowledge source. The system is composed of a term-based retrieval module, two neural modules for both paragraph-level retrieval and sentence-level retrieval, and a neural downstream task module. The system achieves start-of-the-art results on both FEVER (Thorne et al., 2018b) and Hotpot QA (Yang et al., 2018) with significant improvement over the previous best results.

NLI for Consistent Dialogue Modeling. One of the important aspects of NLI applications is to help improve the consistency and faithfulness of natural language generation (NLG) models.

This thesis also studied the NLI models in the context of dialogue modeling. We know that when interacting with chatbots, people carry over many of the same expectations as when interacting with humans (Nass and Moon, 2000). Self-contradictions by these bots are often jarring, immediately disrupt the conversational flow, and help support arguments about whether generative models could ever really understand what they are saying at all (Marcus, 2018). From a listener's perspective, such inconsistent bots fail to gain user trust and their long-term communication confidence. From a speaker's perspective, it violates the maxim of quality in Grice's cooperative principles (Grice, 1975) —"Do not say what you believe to be false." Hence, efforts on reducing contradicting or inconsistent conversations by open-domain chatbots are imperative. Therefore, we introduce the DialoguE COntradiction DEtection task (DECODE) and a new conversational dataset containing both human-human and human-bot contradictory dialogues. We developed a contradiction detection model within NLI paradigm and the results show that our best contradiction detection model correlates well with human judgments and further provides evidence for its usage in both automatically evaluating and improving the consistency of state-of-the-art generative chatbots.

Adversarial NLI and Benchmarking in NLP. The development of challenging large-scale benchmarks like ImageNet (Krizhevsky et al., 2012) in computer vision, and GLUE (Bowman et al., 2015a), SQuAD (Rajpurkar et al., 2016), and others in natural language processing (NLP) is crucially important for the progress in AI. In recent years, we have witnessed rapid development and advancement in deep representation learning technologies. In NLP specifically, the invention of language scale pre-trained transformers like BERT (Devlin et al., 2019a), RoBERTa (Liu et al., 2019b), GPT (Brown et al., 2020a) have achieved superhuman performance on various NLP tasks. As a result, model performance on NLP benchmarks like GLUE (Wang et al., 2018a) has saturated quickly and can no longer be able to give insightful feedback. In spite of the progress and high benchmark scores, empirical and qualitative analysis by NLP practitioners reveals that current NLP models are still far from human-level intelligence and we are not close to solving most NLP tasks (Nie et al., 2019b; Gururangan et al., 2018b; Poliak et al., 2018b; Tsuchiya, 2018; Glockner et al., 2018; Geva et al., 2019; McCoy et al., 2019). We propose an iterative, adversarial human-and-model-in-the-loop solution for NLU dataset collection that addresses both benchmark longevity and robustness issues. In the first stage, human annotators devise examples that our current best models cannot determine the correct label for. These resulting hard examples—which should expose additional model weaknesses—can be added to the training set and used to train a stronger model. We then subject the strengthened model to the same procedure and collect weaknesses over several rounds. After each round, we train a new model and set aside a new test set. The process can be iteratively repeated in a never-ending learning (Mitchell et al., 2018) setting, with the model getting stronger and the test set getting harder in each new round. Thus, not only is the resultant dataset harder than existing benchmarks, but this process also yields a "moving post" dynamic target for NLU systems, rather than a static benchmark that will eventually saturate. We used the described procedure to collect a new dataset called Adversarial NLI (Nie et al., 2020a).

1.1 Thesis Statement

The thesis reveals that rectifying the souring of NLI data is an important factor for both building applicable NLI models in fact-checking and dialogue modeling, as well as for effectively benchmarking general language understanding progress.

1.2 Overview of Chapters

The remainder of this dissertation is organized into five chapters. Chapter 2 presents our work on building a sentence encoder-based NLI model that achieved state-of-the-art performance on the multi-genre NLI task. Chapter 3 presents our work on building an MRS system that achieved state-of-the-art performance on both the extractive QA and the automatic fact-checking task. Experiments on the system reveal the importance of accurate intermediate retrieval in the success of downstream question answering and NLI models. Chapter 4 presents our work on collecting data and utilizing NLI models for automatically evaluating and improving the consistency of open-domain dialogue agents. Chapter 5 illustrates our efforts in dynamically benchmarking general natural language understanding progress via adversarial NLI data collection. Chapter 6 summarizes the contributions herein and discusses the potential opportunities for future work.

CHAPTER 2: SENTENCE ENCODER-BASED NLI

2.1 Introduction and Background

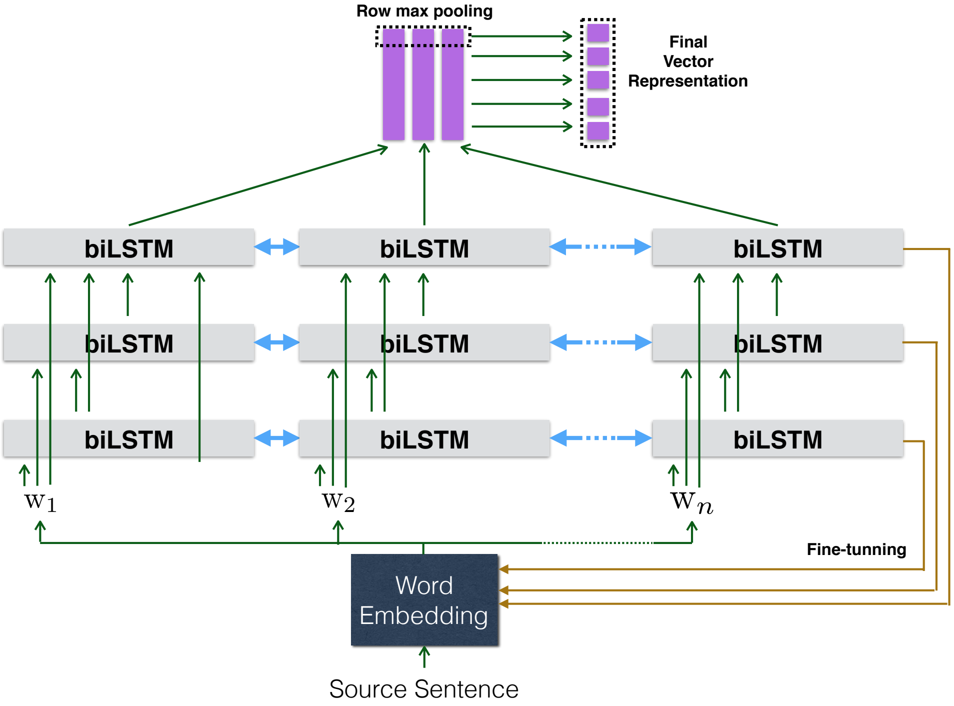


Figure 2.1: Our encoder's architecture: stacked biLSTM with shortcut connections and fine-tuning.

Natural language inference (NLI) or recognizing textual entailment (RTE) is a fundamental semantic task in the field of natural language processing. The problem is to determine whether a given hypothesis sentence can be logically inferred from a given premise sentence. Recently released datasets such as the Stanford Natural Language Inference Corpus (Bowman et al., 2015a) (SNLI) and the Multi-Genre Natural Language Inference Corpus (Williams et al., 2017) (Multi-NLI) have not only encouraged several end-to-end neural network approaches to NLI, but have also served as an evaluation resource for general representation learning of natural language.

Depending on whether a model will first encode a sentence into a fixed-length vector without any incorporating information from the other sentence, the several proposed models can be categorized into two groups: (1) encoding-based models (or sentence encoders), such as Tree-based CNN encoders (TBCNN) in Mou et al. (2016) or the Stack-augmented Parser-Interpreter Neural Network (SPINN) in Bowman et al. (2016), and (2) joint, pairwise models that use cross-features between the two sentences to encode them, such as the Enhanced Sequential Inference Model (ESIM) in Chen et al. (2017c) or the bilateral multi-perspective matching (BiMPM) model in Wang et al. (2017b). Moreover, common sentence encoders can again be classified into tree-based encoders such as SPINN in Bowman et al. (2016) which we mentioned before, or sequential encoders such as the biLSTM model by Bowman et al. (2015a).

In this work, we follow the former approach of encoding-based models, and propose a novel yet simple sequential sentence encoder for the Multi-NLI problem. Our encoder does not require any syntactic information of the sentence. It also does not contain any attention or memory structure. It is basically a stacked (multi-layered) bidirectional LSTM-RNN with shortcut connections (feeding all previous layers' outputs and word embeddings to each layer) and word embedding fine-tuning. The overall supervised model uses these shortcut-stacked encoders to encode two input sentences into two vectors, and then we use a classifier over the vector combination to label the relationship between these two sentences as that of entailment, contradiction, or neutral (similar to the classifier setup of Bowman et al. (2015a) and Conneau et al. (2017b)). Our simple shortcut-stacked encoder achieves strong improvements over existing encoders due to its multi-layered and shortcut-connected properties, on both matched and mismatched evaluation settings for multi-domain natural language inference, as well as on the original SNLI dataset. It is the top single-model (non-ensemble) result in the EMNLP RepEval 2017 Multi-NLI Shared Task (Nangia et al., 2017), and the new state-of-the-art for encoding-based results on the SNLI dataset (Bowman et al., 2015a).

Github Code Link: `https://github.com/easonnie/multiNLI_encoder`

2.2 Model

Our model mainly consists of two separate components, a sentence encoder and an entailment classifier. The sentence encoder compresses each source sentence into a vector representation and the classifier makes a three-way classification based on the two vectors of the two source sentences. The model follows the ‘encoding-based rule’, i.e., the encoder will encode each source sentence into a fixed length vector without any information or function based on the other sentence (e.g., cross-attention or memory comparing the two sentences). In order to fully explore the generalization of the sentence encoder, the same encoder is applied to both the premise and the hypothesis with shared parameters projecting them into the same space. This setting follows the idea of Siamese Networks in Bromley et al. (1993). Figure 2.1 shows the overview of our encoding model (the standard classifier setup is not shown here; see Bowman et al. (2015a) and Conneau et al. (2017b) for that).

2.2.1 Sentence Encoder

Our sentence encoder is simply composed of multiple stacked bidirectional LSTM (biLSTM) layers with shortcut connections followed by a max pooling layer. Let bilstm^i represent the ith biLSTM layer, which is defined as:

$$h_t^i = \text{bilstm}^i(x_t^i, t), \forall t \in [1, 2, ..., n] \tag{2.1}$$

where h_t^i is the output of the ith biLSTM at time t over input sequence $(x_1^i, x_2^i, ..., x_n^i)$.

In a typical **stacked biLSTM** structure, the input of the next LSTM-RNN layer is simply the output sequence of the previous LSTM-RNN layer. In our settings, the input sequences for the ith biLSTM layer are the concatenated outputs of *all the previous layers*, plus the original word embedding sequence. This gives a **shortcut connection** style setup, related to the widely used idea of residual connections in CNNs for computer vision (He et al., 2016a), highway networks for RNNs in speech processing (Zhang et al., 2016), and shortcut connections in hierarchical multitasking learning (Hashimoto et al., 2017); but in our case we feed in all the previous layers' output sequences as well as the word embedding sequence to every layer.

Let $W = (w_1, w_2, ..., w_n)$ represent words in the source sentence. We assume $w_i \in \mathbb{R}^d$ is a word embedding vector which is initialized using some pre-trained vector embeddings (and is then fine-tuned end-to-end via the NLI supervision). Then, the input of ith biLSTM layer at time t is defined as:

$$x_t^1 = w_t \tag{2.2}$$

$$x_t^i = [w_t, h_t^{i-1}, h_t^{i-2}, ...h_t^1] \quad (\text{for } \ i > 1) \tag{2.3}$$

where [] represents vector concatenation.

Then, assuming we have m layers of biLSTM, the final vector representation will be obtained by applying row-max-pool over the output of the last biLSTM layer, similar to Conneau et al. (2017b). The final layer is defined as:

$$H^m = (h_1^m, h_2^m, ..., h_n^m) \tag{2.4}$$

$$v = max(H^m) \tag{2.5}$$

where $h_i^m, v \in \mathbb{R}^{2d_m}$, $H^m \in \mathbb{R}^{2d_m \times n}$, d_m is the dimension of the hidden state of the last forward and backward LSTM layers, and v is the final vector representation for the source sentence (which is later fed to the NLI classifier).

The closest encoder architecture to ours is that of Conneau et al. (2017b), whose model consists of a single-layer biLSTM with a max-pooling layer, which we treat as our starting point. Our experiments (Section 2.4) demonstrate that our enhancements of the stacked-biRNN with shortcut connections provide significant gains on top of this baseline (for both SNLI and Multi-NLI).

2.2.2 Entailment Classifier

After we obtain the vector representation for the premise and hypothesis sentence, we apply three matching methods to the two vectors (i) concatenation (ii) element-wise distance and (iii) element-wise product for these two vectors and then concatenate these three match vectors (based on the heuristic matching presented in Mou et al. (2016)). Let v_p and v_h be the vector representations for premise and hypothesis, respectively. The matching vector is then defined as:

$$m = [v_p, v_h, |v_p - v_h|, v_p \otimes v_h] \tag{2.6}$$

At last, we feed this final concatenated result m into a MLP layer and use a softmax layer to make final classification.

Layers and Dimensions		Accuracy	
#layers	bilstm-dim	Matched	Mismatched
1	512	72.5	72.9
2	512 + 512	73.4	73.6
1	1024	72.9	72.9
2	512 + 1024	73.7	74.2
1	2048	73.0	73.5
2	512 + 2048	73.7	74.2
2	1024 + 2048	73.8	74.4
2	2048 + 2048	74.0	74.6
3	512 + 1024 + 2048	**74.2**	**74.7**

Table 2.1: Analysis of results for models with different # of biLSTM layers and their hidden state dimensions.

	Matched	Mismatched
without any shortcut connection	72.6	73.4
only word shortcut connection	74.2	74.6
full shortcut connection	**74.2**	**74.7**

Table 2.2: Ablation results with and without shortcut connections.

Word-Embedding	Matched	Mismatched
fixed	71.8	72.6
fine-tuned	**72.7**	**72.8**

Table 2.3: Ablation results with and without fine-tuning of word embeddings.

# of MLPs	Activation	Matched	Mismatched
1	tanh	73.7	74.1
2	tanh	73.5	73.6
1	relu	74.1	74.7
2	relu	**74.2**	**74.7**

Table 2.4: Ablation results for different MLP classifiers.

2.3 Experimental Setup

2.3.1 Datasets

As instructed in the RepEval Multi-NLI shared task, we use all of the training data in Multi-NLI combined with 15% randomly selected samples from the SNLI training set resampled at each epoch) as our final training set for all models; and we use both the cross-domain ('mismatched') and in-domain ('matched') Multi-NLI development sets for model selection. For the SNLI test results in Table 2.5, we train on only the SNLI training set (and we also verify that the tuning decisions hold true on the SNLI dev set).

Model	**Accuracy**		
	SNLI	**Multi-NLI Matched**	**Multi-NLI Mismatched**
CBOW (Williams et al., 2017)	80.6	65.2	64.6
biLSTM Encoder (Williams et al., 2017)	81.5	67.5	67.1
300D Tree-CNN Encoder (Mou et al., 2016)	82.1	–	–
300D SPINN-PI Encoder (Bowman et al., 2016)	83.2	–	–
300D NSE Encoder (Munkhdalai and Yu, 2017)	84.6	–	–
biLSTM-Max Encoder (Conneau et al., 2017b)	84.5	–	–
Our biLSTM-Max Encoder	85.2	71.7	71.2
Our Shortcut-Stacked Encoder	**86.1**	**74.6**	**73.6**

Table 2.5: Final Test Results on SNLI and Multi-NLI datasets.

2.3.2 Parameter Settings

We use cross-entropy loss as the training objective with Adam-based (Kingma and Ba, 2015) optimization with a batch size of 32. The starting learning rate is 0.0002 with half decay every two epochs. The number of hidden units for MLP classifier is 1600. A dropout layer is also applied on the output of each MLP layer, with dropout rate set to 0.1. We used pre-trained 300D Glove 840B vectors (Pennington et al., 2014) to initialize the word embeddings. Tuning decisions for word embedding training strategy, the hyperparameters of dimension and number of layers for biLSTM, and the activation type and number of layers for MLP, are all explained in Section 2.4.

2.4 Results and Analysis

2.4.1 Ablation Analysis Results

We now investigate the effectiveness of each of the enhancement components in our overall model. These ablation results are shown in Tables 2.1, 2.2, 2.3 and 2.4, all based on the Multi-NLI development sets. Finally, Table 2.5 shows results for different encoders on SNLI and Multi-NLI test sets.

First, Table 2.1 shows the performance changes for different number of biLSTM layers and their varying dimension size. The dimension size of a biLSTM layer is referring to the dimension of the hidden state for both the forward and backward LSTM-RNNs. As shown, each added layer model improves the accuracy and we achieve a substantial improvement in accuracy (around 2%) on both matched and mismatched settings, compared to the single-layer biLSTM in Conneau et al. (2017b). We only experimented with up to 3 layers with 512, 1024, 2048 dimensions each, so the model still has potential to improve the result further with a larger dimension and more layers.

Next, in Table 2.2, we show that the shortcut connections among the biLSTM layers are also an important contributor to accuracy improvement (around 1.5% on top of the full 3-layered stacked-RNN model). This demonstrates that simply stacking the biLSTM layers is not sufficient to handle a complex task like Multi-NLI and it is significantly better to have the higher layer connected to both the output and the original input of all the previous layers (note that Table 2.1 results are based on multi-layered models with shortcut connections).

Next, in Table 2.3, we show that fine-tuning the word embeddings also improves results, again for both the in-domain task and cross-domain tasks (the ablation results are based on a smaller model with a 128+256 2-layer biLSTM). Hence, all our models were trained with word embeddings being fine-tuned. The last ablation in Table 2.4 shows that a classifier with two layers of ReLU is preferable than other options. Thus, we use that setting for our strongest encoder.

2.4.2 Multi-NLI and SNLI Test Results

Finally, in Table 2.5, we report the test results for MNLI and SNLI. First for Multi-NLI, we improve substantially over the CBOW and biLSTM Encoder baselines reported in the dataset paper (Williams et al., 2017). We also show that our final shortcut-based stacked encoder achieves around 3% improvement as compared to the 1-layer biLSTM-Max Encoder in the second last row (using the exact same classifier and optimizer settings). Our shortcut-encoder was also the top singe-model (non-ensemble) result on the EMNLP RepEval Shared Task leaderboard.

Next, for SNLI, we compare our shortcut-stacked encoder with the current state-of-the-art encoders from the SNLI leaderboard (`https://nlp.stanford.edu/projects/snli/`). We also compare to the recent biLSTM-Max Encoder of Conneau et al. (2017b), which served as our model's 1-layer starting point.[1] The results indicate that 'Our Shortcut-Stacked Encoder' surpasses all the previous state-of-the-art encoders, and achieves the new best encoding-based result on SNLI, suggesting the general effectiveness of simple shortcut-connected stacked layers in sentence encoders.

[1] Note that the 'Our biLSTM-Max Encoder' results in the second-last row are obtained using our reimplementation of the Conneau et al. (2017b) model; our version is 0.7% better, likely due to our classifier and optimizer settings.

2.5 Conclusion

We explored various simple combinations and connections of biLSTM-RNN layered architectures and developed a Shortcut-Stacked Sentence Encoder for natural language inference. Our model is the top single result in the EMNLP RepEval 2017 Multi-NLI Shared Task, and it also surpasses the state-of-the-art encoders for the SNLI dataset. In future work, we are also evaluating the effectiveness of shortcut-stacked sentence encoders on several other semantic tasks.

2.6 Addendum: Shortcut vs. Residual

In later experiments, we found that a residual connection can achieve similar accuracies with fewer number of parameters, compared to a shortcut connection. Therefore, in order to reduce the model size and to also follow the SNLI leader-board settings (e.g., 300D and 600D embeddings), we performed some additional SNLI experiments with the shortcut connections replaced with residual connections, where the input to each next biLSTM layer is the concatenation of the word embedding and the summation of outputs of all previous layers (related to ResNet in computer vision (He et al., 2016a)). Table 2.6 shows these residual-connection SNLI test results and the parameter comparison to shortcut-connection models (using 3 stacked-biLSTM layers, and one 800-unit MLP layer, based on SNLI dev set tuning).

Model	#param	Dev	Test
300D Residual-Stacked-Encoder	9.7M	86.4	85.7
600D Residual-Stacked-Encoder	28.9M	**87.0**	**86.0**
600D Shortcut-Stacked-Encoder	34.7M	86.8	85.9

Table 2.6: Results on SNLI for the fewer-parameter Residual-Stacked Encoder models. Each model has 3 biLSTM-stacked layers and 1 MLP layer. The #param column denotes the number of parameters in millions.

CHAPTER 3: FACT CHECKING, EXTRACTIVE QA, AND SEMANTIC RETRIEVAL IN MACHINE READING AT SCALE

3.1 Introduction

Extracting external textual knowledge for machine comprehensive systems such as fact checking and question answering has long been an important yet challenging problem. Success requires not only precise retrieval of the relevant information sparsely restored in a large knowledge source but also a deep understanding of both the selected knowledge and the input query to give the corresponding output. Initiated by Chen et al. (2017b), the task was termed as Machine Reading at Scale (MRS), seeking to provide a challenging situation where machines are required to do both semantic retrieval and comprehension at different levels of granularity for the final downstream task.

Progress on MRS has been made by separately improving the information retrieval (IR) submodules and the machine comprehension sub-modules with recent advancements on representation learning (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019b). However, partially due to the lack of annotated data for intermediate retrieval in an MRS setting, the evaluations were done mainly on the final downstream task and with much less consideration on the intermediate retrieval performance. This led to the convention that upstream retrieval modules mostly focus on getting better coverage of the downstream information such that the upper-bound of the downstream score can be improved, rather than finding more exact information. This convention is misaligned with the nature of MRS where equal effort should be put in emphasizing the models' joint performance and optimizing the relationship between the semantic retrieval and the downstream comprehension sub-tasks.

Hence, to shed light on the importance of semantic retrieval for downstream comprehension tasks, we start by establishing a simple yet effective hierarchical pipeline system for MRS using Wikipedia as the external knowledge source. The system is composed of a term-based retrieval module, two neural modules for both the paragraph-level retrieval and sentence-level retrieval, and a neural downstream task module. We evaluated the system on two recent large-scale open domain benchmarks for fact verification and multi-hop QA, namely FEVER (Thorne et al., 2018a) and HOTPOTQA (Yang et al., 2018), in which retrieval performance can also be evaluated accurately since intermediate annotations on evidences are provided. Our system achieves the state-of-the-art results with 45.32% for answer EM and 25.14% joint EM on HOTPOTQA (8% absolute improvement on answer EM and doubling the joint EM over the previous best results) and with 67.26% on FEVER score (3% absolute improvement over previously published systems).

We then provide empirical studies to validate design decisions. Specifically, we show the necessity of both the paragraph-level retrieval and sentence-level retrieval for maintaining good performance, and further illustrate that a better semantic retrieval module not only is beneficial to achieving high recall and keeping a high upper bound for downstream tasks, but also plays an important role in shaping the downstream data distribution and providing more relevant and high-quality data for downstream sub-module training and inference. These mechanisms are vital for a good MRS system on both QA and fact verification.

3.2 Related Work

Machine Reading at Scale First proposed and formalized in Chen et al. (2017b), MRS has gained popularity with increasing amount of work on both dataset collection (Joshi et al., 2017; Welbl et al., 2018) and MRS model developments (Wang et al., 2018b; Clark and Gardner, 2017; Htut et al., 2018). In some previous work (Lee et al., 2018), paragraph-level retrieval modules were mainly for improving the recall of required information, while in some other works (Yang et al., 2018), sentence-level retrieval modules were merely for solving the auxiliary sentence selection task. In our work, we focus on revealing the relationship between semantic retrieval at different granularity levels and the downstream comprehension task. To the best of our knowledge, we are the first to apply and optimize neural semantic retrieval at both the paragraph and sentence levels for MRS.

Automatic Fact Checking: Recent work (Thorne and Vlachos, 2018) formalized the task of automatic fact checking from the viewpoint of machine learning and NLP. The release of FEVER (Thorne et al., 2018a) stimulates many recent developments (Nie et al., 2019a; Yoneda et al., 2018; Hanselowski et al., 2018) on data-driven neural networks for automatic fact checking. We consider the task also as MRS because they share almost the same setup except that the downstream task is verification or natural language inference (NLI) rather than QA.

Information Retrieval Success in deep neural networks inspires their application to information retrieval (IR) tasks (Huang et al., 2013; Guo et al., 2016; Mitra et al., 2017; Dehghani et al., 2017). In typical IR settings, systems are required to retrieve and rank (Nguyen et al., 2016) elements from a collection of documents based on their relevance to the query. This setting might be very different from the retrieval in MRS where systems are asked to select facts needed to answer a question or verify a statement. We refer to the retrieval in MRS as *Semantic Retrieval* since it emphasizes on semantic understanding.

3.3 Method

In previous works, an MRS system can be complicated with different sub-components processing different retrieval and comprehension sub-tasks at different levels of granularity, and with some sub-components intertwined. For interpretability considerations, we used a unified pipeline setup. The overview of the system is in Fig. 3.1.

To be specific, we formulate the MRS system as a function that maps an input tuple $(q, \mathbf{K})$ to an output tuple $(\hat{y}, \mathbf{S})$ where q indicates the input query, $\mathbf{K}$ is the textual KB (Knowledge Base), $\hat{y}$ is the output prediction, and $\mathbf{S}$ are selected supporting sentences from Wikipedia. Let $\mathbf{E}$ denotes a set of necessary evidences or facts selected from $\mathbf{K}$ for the prediction. For a QA task, q is the

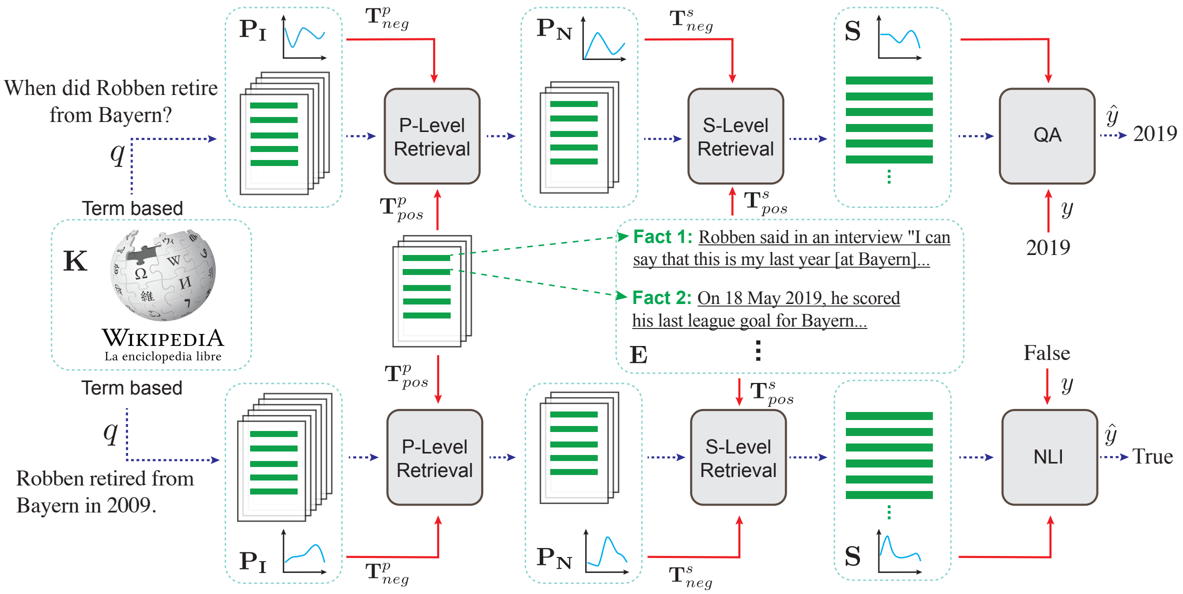


Figure 3.1: System Overview: blue dotted arrows indicate the inference flow and the red solid arrows indicate the training flow. Grey rounded rectangles are neural modules with different functionality. The two retrieval modules were trained with all positive examples from annotated ground truth set and negative examples sampled from the direct upstream modules. Thus, the distribution of negative examples is subjective to the quality of the upstream module.

input question and $\hat{y}$ is the predicted answer. For a verification task, q is the input claim and $\hat{y}$ is the predicted truthfulness of the input claim. For all tasks, $\mathbf{K}$ is Wikipedia.

The system procedure is listed below:

(1) Term-Based Retrieval: To begin with, we used a combination of the TF-IDF method and a rule-based keyword matching method[1] to narrow the scope from the entire Wikipedia down to a set of related paragraphs; this is a standard procedure in MRS (Chen et al., 2017b; Lee et al., 2018; Nie et al., 2019a). The focus of this step is to efficiently select a candidate set $\mathbf{P_I}$ that can cover the information as much as possible ($\mathbf{P_I} \subset \mathbf{K}$) while keeping the size of the set acceptable enough for downstream processing.

(2) Paragraph-Level Neural Retrieval: After obtaining the initial set, we compare each paragraph in $\mathbf{P_I}$ with the input query q using a neural model (which will be explained later in Sec 3.3.1). The outputs of the neural model are treated as the relatedness score between the input query and the paragraphs. The scores will be used to sort all the upstream paragraphs. Then, $\mathbf{P_I}$ will be

[1] Details of term-based retrieval are in Appendix.

narrowed to a new set $\mathbf{P_N}$ ($\mathbf{P_N} \subset \mathbf{P_I}$) by selecting the top k_p paragraphs having relatedness score higher than some threshold value h_p (going out from the P-Level grey box in Fig. 3.1). k_p and h_p would be chosen by keeping a good balance between the recall and precision of the paragraph retrieval.

(3) Sentence-Level Neural Retrieval: Next, we select the evidence at the sentence-level by decomposing all the paragraphs in $\mathbf{P_N}$ into sentences. Similarly, each sentence is compared with the query using a neural model (see details in Sec 3.3.1) and we obtain a set of sentences $\mathbf{S} \subset \mathbf{P_N}$ for the downstream task by choosing the top k_s sentences with output scores higher than some threshold h_s (S-Level grey box in Fig. 3.1). During evaluation, $\mathbf{S}$ is often evaluated against some ground truth sentence set denoted as $\mathbf{E}$.

(4) Downstream Modeling: At the final step, we simply applied task-specific neural models (e.g., QA and NLI) on the concatenation of all the sentences in $\mathbf{S}$ and the query, obtaining the final output $\hat{y}$.

In some experiments, we modified the setup for certain analysis or ablation purposes which will be explained individually in Sec 3.6.

3.3.1 Modeling and Training

Throughout all our experiments, we used **BERT-Base** (Devlin et al., 2019b) to provide the state-of-the-art contextualized modeling of the input text.[2]

Semantic Retrieval: We treated the neural semantic retrieval at both the paragraph and sentence level as binary classification problems with the models' parameters updated by minimizing binary cross entropy loss. To be specific, we fed the query and context into BERT as:

$$[CLS]\ Query\ [SEP]\ Context\ [SEP]$$

[2] We used the pytorch BERT implementation in `https://github.com/huggingface/pytorch-pretrained-BERT`.

We applied an affine layer and sigmoid activation on the last layer output of the $[CLS]$ token which is a scalar value. The parameters were updated with the objective function:

$$\mathcal{J}_{retri} = -\sum_{i \in \mathbf{T}^{p/s}_{pos}} \log(\hat{p}_i) - \sum_{i \in \mathbf{T}^{p/s}_{neg}} \log(1 - \hat{p}_i)$$

where $\hat{p}_i$ is the output of the model, $\mathbf{T}^{p/s}_{pos}$ is the positive set and $\mathbf{T}^{p/s}_{neg}$ is the negative set. As shown in Fig. 3.1, at sentence level, ground-truth sentences were served as positive examples while other sentences from the upstream retrieved set were served as negative examples. Similarly at the paragraph-level, paragraphs having any ground-truth sentence were used as positive examples and other paragraphs from the upstream term-based retrieval processes were used as negative examples.

QA: We followed Devlin et al. (2019b) for QA span prediction modeling. To correctly handle yes-or-no questions in HOTPOTQA, we fed the two additional "yes" and "no" tokens between $[CLS]$ and the $Query$ as:

$$[CLS]\ yes\ no\ Query\ [SEP]\ Context\ [SEP]$$

where the supervision was given to the second or the third token when the answer is "yes" or "no", such that they can compete with all other predicted spans. The parameters of the neural QA model were trained to maximize the log probabilities of the true start and end indexes as:

$$\mathcal{J}_{qa} = -\sum_{i} \big[\log(\hat{y}^s_i) + \log(\hat{y}^e_i) \big]$$

where $\hat{y}^s_i$ and $\hat{y}^e_i$ are the predicted probability on the ground-truth start and end position for the ith example, respectively. It is worth noting that we used ground truth supporting sentences plus some other sentences sampled from the upstream retrieved set as the context for training the QA module such that it will adapt to the upstream data distribution during inference.

Fact Verification: Following Thorne et al. (2018a), we formulate downstream fact verification as a 3-way natural language inference (NLI) classification problem (MacCartney and Manning, 2009; Bowman et al., 2015a) and train the model with 3-way cross entropy loss. The input format is the same as that of semantic retrieval and the objective is $\mathcal{J}_{ver} = -\sum_i \mathbf{y}_i \cdot \log(\hat{\mathbf{y}}_i)$, where $\hat{\mathbf{y}}_i \in \mathbf{R}^3$ denotes the model's output for the three verification labels, and $\mathbf{y}_i$ is a one-hot embedding for the ground-truth label. For verifiable queries, we used ground truth evidential sentences plus some other sentences sampled from the upstream retrieved set as new evidential context for NLI. For non-verifiable queries, we only used sentences sampled from the upstream retrieved set as context because those queries are not associated with ground truth evidential sentences. This detail is important for the model to identify non-verifiable queries and will be explained more in Sec 3.6.

It is worth noting that each sub-module in the system relies on its preceding sub-module to provide data both for training and inference. This means that there will be upstream data distribution misalignment if we trained the sub-module in isolation without considering the properties of its precedent upstream module. The problem is similar to the concept of internal covariate shift (Ioffe and Szegedy, 2015), where the distribution of each layer's inputs changes inside a neural network. Therefore, it makes sense to study this issue in a joint MRS setting rather than a typical supervised learning setting where training and test data tend to be fixed and modules being isolated. We release our code and the organized data both for reproducibility and providing an off-the-shelf testbed to facilitate future research on MRS.

3.4 Experimental Setup

MRS requires a system not only to retrieve relevant content from textual KBs but also to poccess enough understanding ability to solve the downstream task. To understand the impact or importance of semantic retrieval on the downstream comprehension, we established a unified experimental setup that involves two different downstream tasks, i.e., multi-hop QA and fact verification.

3.4.1 Tasks and Datasets

HOTPOTQA: This dataset is a recent large-scale QA dataset that brings in new features: (1) the questions require finding and reasoning over multiple documents; (2) the questions are diverse and not limited to pre-existing KBs; (3) it offers a new comparison question type (Yang et al., 2018). We experimented with our system on HOTPOTQA in the fullwiki setting, where a system must find the answer to a question in the scope of the entire Wikipedia, an ideal MRS setup. The sizes of the train, dev and test split are 90,564, 7,405, and 7,405. More importantly, HOTPOTQA also provides human-annotated sentence-level supporting facts that are needed to answer each question. Those intermediate annotations enable evaluation on models' joint ability on both fact retrieval and answer span prediction, facilitating our direct analysis on the explainable predictions and their relations with the upstream retrieval.

FEVER: The Fact Extraction and VERification dataset (Thorne et al., 2018a) is a recent dataset collected to facilitate the automatic fact checking. The work also proposes a benchmark task in which given an arbitrary input claim, candidate systems are asked to select evidential sentences from Wikipedia and label the claim as either SUPPORT, REFUTE, or NOT ENOUGH INFO, if the claim can be verified to be true, false, or non-verifiable, respectively, based on the evidence. The sizes of the train, dev and test split are 145,449, 19,998, and 9,998. Similar to HOTPOTQA, the dataset provides annotated sentence-level facts needed for the verification. These intermediate annotations could provide an accurate evaluation on the results of semantic retrieval and thus is well suited for the analysis on the effects of the retrieval module on downstream verification.

As in Chen et al. (2017b), we use Wikipedia as our unique knowledge base because it is a comprehensive and self-evolving information source often used to facilitate intelligent systems. Moreover, as Wikipedia is the source for both HOTPOTQA and FEVER, it helps standardize any further analysis of the effects of semantic retrieval on the two different downstream tasks.

3.4.2 Metrics

Following Thorne et al. (2018a); Yang et al. (2018), we used annotated sentence-level facts to calculate the F1, Precision and Recall scores for evaluating sentence-level retrieval. Similarly, we labeled all the paragraphs that contain any ground truth fact as ground truth paragraphs and used the same three metrics for paragraph-level retrieval evaluation. For HOTPOTQA, following Yang et al. (2018), we used exact match (EM) and F1 metrics for QA span prediction evaluation, and used the joint EM and F1 to evaluate models' joint performance on both retrieval and QA. The joint EM and F1 are calculated as: $P_j = P_a \cdot P_s; R_j = R_a \cdot R_s; F_j = \frac{2P_j \cdot R_j}{P_j + R_j}; \text{EM}_j = \text{EM}_a \cdot \text{EM}_s$, where P, R, and EM denote precision, recall and EM; the subscript a and s indicate that the scores are for answer span and supporting facts.

For the FEVER task, following Thorne et al. (2018a), we used the Label Accuracy for evaluating downstream verification and the Fever Score for joint performance. Fever score will award one point for each example with the correct predicted label only if all ground truth facts were contained in the predicted facts set with at most 5 elements. We also used Oracle Score for the two retrieval modules. The scores were proposed in Nie et al. (2019a) and indicate the upperbound of the final FEVER Score at one intermediate layer assuming all downstream modules are perfect. All scores are averaged over the examples in the whole evaluation set.

3.5 Results on Benchmarks

We chose the best system based on the dev set, and used that for submitting private test predictions on both FEVER and HOTPOTQA[3].

As can be seen in Table 3.1, with the proposed hierarchical system design, the whole pipeline system achieves a start-of-the-art on HOTPOTQA with large-margin improvements on all the metrics. More specifically, the biggest improvement comes from the EM for the supporting fact which in turn leads to doubling of the joint EM on previous best results. The scores for answer

[3] Results can also be found at the leaderboard websites for the two datasets: `https://hotpotqa.github.io` and `https://competitions.codalab.org/competitions/18814`

Method	Ans		Sup		Joint	
	EM	F1	EM	F1	EM	F1
Yang Yang et al. (2018)	24.7	34.4	5.3	41.0	2.5	17.7
Ding Ding et al. (2019)	37.6	49.4	23.1	58.5	12.2	35.3
whole pip.	**46.5**	**58.8**	**39.9**	**71.5**	**26.6**	**49.2**
Dev set						
Yang Yang et al. (2018)	24.0	32.9	3.9	37.7	1.9	16.2
MUPPET	30.6	40.3	16.7	47.3	10.9	27.0
Ding Ding et al. (2019)	37.1	48.9	22.8	57.7	12.4	34.9
whole pip.	**45.3**	**57.3**	**38.7**	**70.8**	**25.1**	**47.6**
Test set						

Table 3.1: Results of systems on HOTPOTQA.

Model	F1	LA	FS
Hanselowski Hanselowski et al. (2018)	-	68.49	64.74
Yoneda Yoneda et al. (2018)	35.84	69.66	65.41
Nie Nie et al. (2019a)	51.37	69.64	66.15
Full system (single)	**76.87**	**75.12**	**70.18**
Dev set			
Hanselowski Hanselowski et al. (2018)	37.33	65.22	61.32
Yoneda Yoneda et al. (2018)	35.21	67.44	62.34
Nie Nie et al. (2019a)	52.81	68.16	64.23
Full system (single)	**74.62**	**72.56**	**67.26**
Test set			

Table 3.2: Performance of systems on FEVER. “F1” indicates the sentence-level evidence F1 score. “LA” indicates Label Acc. without considering the evidence prediction. “FS”=FEVER Score (Thorne et al., 2018a)

predictions are also higher than all previous best results with ∼8 absolute points increase on EM and ∼9 absolute points on F1. All the improvements are consistent between test and dev set evaluation.

Similarly for FEVER, we showed F1 for evidence, the Label Accuracy, and the FEVER Score (same as benchmark evaluation) for models in Table 3.2. Our system obtained substantially higher scores than all previously published results with a ∼4 and ∼3 points absolute improvement on Label Accuracy and FEVER Score. In particular, the system gains 74.62 on the evidence F1, 22 points greater that of the second system, demonstrating its ability on semantic retrieval.

Previous systems (Ding et al., 2019; Yang et al., 2018) on HOTPOTQA treat supporting fact retrieval (sentence-level retrieval) just as an auxiliary task for providing extra model explainability. In Nie et al. (2019a), although they used a similar three-stage system for FEVER, they only applied one neural retrieval module at sentence-level which potentially weakens its retrieval ability. Both of these previous best systems are different from our fully hierarchical pipeline approach. These observations lead to the assumption that the performance gain comes mainly from the hierarchical retrieval and its positive effects on downstream tasks. Therefore, to validate the system design decisions in Sec 3.3 and reveal the importance of semantic retrieval towards downstream performance, we conducted a series of ablation and analysis experiments on all the modules. We started by examining the necessity of both the paragraph and sentence retrieval and give insights on why both of them matter.

3.6 Analysis and Ablations

Intuitively, both the paragraph-level and sentence-level retrieval sub-module help speeding up the downstream processing. More importantly, since downstream modules were trained by sampled data from upstream modules, both of neural retrieval sub-modules also play an implicit but important role in controlling the immediate retrieval distribution i.e. the distribution of set $\mathbf{P_N}$ and set S (as shown in Fig. 3.1), and providing better inference data and training data for downstream modules.

3.6.1 Ablation Studies

Setups: To reveal the importance of neural retrieval modules at both paragraph and sentence level for maintaining the performance of the overall system, we removed either of them and examine the consequences. Because the removal of a module in the pipeline might change the distribution of the input of the downstream modules, we re-trained all the downstream modules accordingly. To be specific, in the system without the paragraph-level neural retrieval module, we re-trained the sentence-level retrieval module with negative sentences directly sampled from the

Method	P-Level Retrieval			S-Level Retrieval				Answer		Joint	
	Prec.	Rec.	F1	EM	Prec.	Rec.	F1	EM	F1	EM	F1
Whole Pip.	35.17	87.93	50.25	**39.86**	**75.60**	**71.15**	**71.54**	**46.50**	**58.81**	**26.60**	**49.16**
Pip. w/o p-level	6.02	89.53	11.19	0.58	29.57	60.71	38.84	31.23	41.30	0.34	19.71
Pip. w/o s-level	35.17	87.92	50.25	-	-	-	-	44.77	56.71	-	-

Table 3.3: Ablation over the paragraph-level and sentence-level neural retrieval sub-modules on HotpotQA.

Method	P-Level Retrieval				S-Level Retrieval				Verification		
	Orcl.	Prec.	Rec.	F1	Orcl.	Prec.	Rec.	F1	LA	FS	L-F1 (S/R/N)
Whole Pip.	94.15	48.84	91.23	63.62	88.92	**71.29**	83.38	76.87	**70.18**	**75.01**	81.7/75.7/**67.1**
Pip. w/o p-level	**94.69**	18.11	**92.03**	30.27	**91.07**	44.47	**86.60**	58.77	61.55	67.01	76.5/72.7/40.8
Pip. w/o s-level	94.15	48.84	91.23	63.62	-	-	-	-	55.92	61.04	72.1/67.6/27.7

Table 3.4: Ablation over the paragraph-level and sentence-level neural retrieval sub-modules on FEVER. "LA"=Label Accuracy; "FS"=FEVER Score; "Orcl." is the oracle upperbound of FEVER Score assuming all downstream modules are perfect. "L-F1 (S/R/N)" means the classification f1 scores on the three verification labels: Support, Refute, and Not Enough Info.

term-based retrieval set and then also re-trained the downstream QA or verification module. In the system without the sentence-level neural retrieval module, we re-train the downstream QA or verification module by sampling data from both ground truth set and retrieved set directly from the paragraph-level module. We tested the simplified systems on both FEVER and HotpotQA.

Results: Tables 3.3 and 3.4 shows the ablation results for the two neural retrieval modules at both paragraph and sentence level on HotpotQA and FEVER. To begin with, we can see that removing the paragraph-level retrieval module significantly reduces the precision for sentence-level retrieval and the corresponding F1 on both tasks. More importantly, this loss of retrieval precision also led to substantial decreases for all the downstream scores on both the QA and verification task in spite of their higher upper-bound and recall scores. This indicates that the negative effects on the downstream module induced by the omission of paragraph-level retrieval can not be amended by the sentence-level retrieval module, and focusing semantic retrieval merely on improving the recall or the upper-bound of the final score will risk jeopardizing the performance of the overall system.

Next, the removal of the sentence-level retrieval module induces a ∼2 point drop on EM and F1 score in the QA task, and a ∼15 point drop on the FEVER Score in the verification task. This

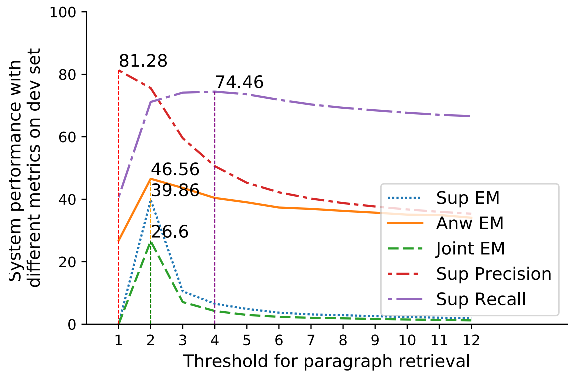


Figure 3.2: The results of EM for supporting fact, answer prediction and joint score, and the results of supporting fact precision and recall with different values of k_p at paragraph-level retrieval on HotpotQA.

suggests that rather than just enhance explainability for QA, the sentence-level retrieval module can also help pinpoint relevant information and reduce the noise in the evidence that might otherwise distract the downstream comprehension module. Another interesting finding is that without sentence-level retrieval module, the QA module suffered much less than the verification module; conversely, the removal of the paragraph-level retrieval neural module induces a 11 point drop on answer EM comparing to a ~9 point drop on Label Accuracy in the verification task. This seems to indicate that the downstream QA module relies more on the upstream paragraph-level retrieval whereas the verification module relies more on the upstream sentence-level retrieval. Finally, we also evaluate the F1 score on FEVER for each classification label and we observe a significant drop of F1 on Not Enough Info category without retrieval module, meaning that semantic retrieval is vital for the downstream verification module's discriminative ability on Not Enough Info label.

3.6.2 Sub-Module Change Analysis

To further study the effects of upstream semantic retrieval towards downstream tasks, we change training or inference data between intermediate layers and then examine how this modification will affect the downstream performance.

Effects of Paragraph-level Retrieval We fixed $h_p = 0$ (the value achieving the best performance) and re-trained all the downstream parameters and track their performance as k_p (the

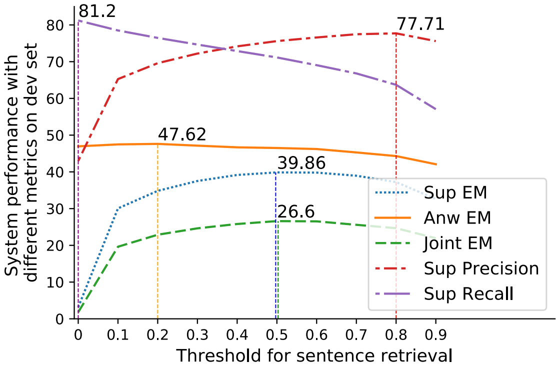


Figure 3.3: The results of EM for supporting fact, answer prediction and joint score, and the results of supporting fact precision and recall with different values of h_s at sentence-level retrieval on HOTPOTQA.

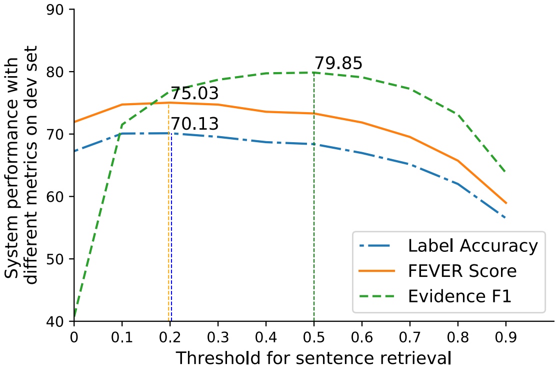


Figure 3.4: The results of Label Accuracy, FEVER Score, and Evidence F1 with different values of h_s at sentence-level retrieval on FEVER.

number of selected paragraphs) being changed from 1 to 12. The increasing of k_p means a potential higher coverage of the answer but more noise in the retrieved facts. Fig. 3.2 shows the results. As can be seen that the EM scores for supporting fact retrieval, answer prediction, and joint performance increase sharply when k_p is changed from 1 to 2. This is consistent with the fact that at least two paragraphs are required to answer each question in HOTPOTQA. Then, after the peak, every score decreases as k_p becomes larger except the recall of supporting fact which peaks when $k_p = 4$. This indicates that even though the neural sentence-level retrieval module possesses a certain level of ability to select correct facts from noisier upstream information, the final QA module is more sensitive to upstream data and fails to maintain the overall system performance. Moreover, the reduction on answer EM and joint EM suggests that it might be risky to give too much information for downstream modules with a unit of a paragraph.

Effects of Sentence-level Retrieval Similarly, to study the effects of the neural sentence-level retrieval module towards downstream QA and verification modules, we fixed k_s to be 5 and set h_s ranging from 0.1 to 0.9 with a 0.1 interval. Then, we re-trained the downstream QA and verification modules with different h_s value and experimented on both HOTPOTQA and FEVER.

Question Answering: Fig. 3.3 shows the performance trend. Intuitively, the precision increases while the recall decreases as the system becomes more strict about the retrieved sentences. The EM scores for supporting fact retrieval and joint performance reaches their highest values when $h_s = 0.5$, a natural balancing point between precision and recall. More interestingly, the EM score for answer prediction peaks when $h_s = 0.2$ and where the recall is higher than the precision. This misalignment between answer prediction performance and retrieval performance indicates that unlike the observation at the paragraph-level, the downstream QA module is able to withstand a certain amount of noise at the sentence-level and benefits from a higher recall.

Fact Verification: Fig. 3.4 shows the trends for Label Accuracy, FEVER Score, and Evidence F1 by modifying the upstream sentence-level threshold h_s. We observed that the general trend is similar to that of the QA task where both the label accuracy and FEVER score peak at $h_s = 0.2$ whereas the retrieval F1 peaks at $h_s = 0.5$. Note that, although the downstream verification could take advantage of a higher recall, the module is more sensitive to sentence-level retrieval compared to the QA module in HOTPOTQA.

3.6.3 Answer Breakdown

We further sample 200 examples from HOTPOTQA and manually tag them according to several common answer types (Yang et al., 2018). The proportion of different answer types is shown in Figure 3.5. The performance of the system on each answer type is shown in Table 3.5. The most frequent answer type is 'Person' (24%) and the least frequent answer type is 'Event' (2%). It is also interesting to note that the model performs the best in Yes/No questions as shown in Table 3.5, reaching an accuracy of 70.6%.

Answer Type	Total	Correct	Acc. (%)
Person	50	28	56.0
Location	31	14	45.2
Date	26	13	50.0
Number	14	4	28.6
Artwork	19	7	36.8
Yes/No	17	12	**70.6**
Event	5	2	40.0
Common noun	11	3	27.3
Group/Org	17	6	35.3
Other PN	20	9	45.0
Total	200	98	49.0

Table 3.5: System performance on different answer types. “PN”= Proper Noun

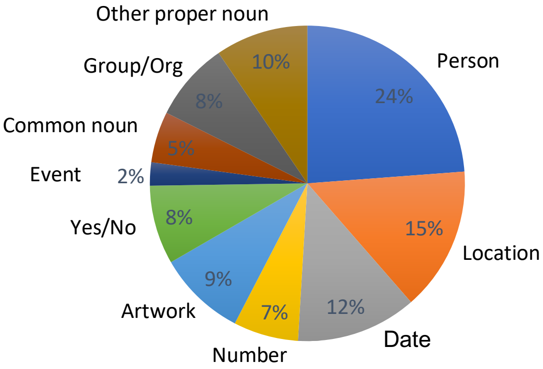


Figure 3.5: Proportion of answer types.

3.6.4 Examples

Fig. 3.6 shows an example that is correctly handled by the full pipeline system but not by the system without paragraph-level retrieval module. We can see that it is very difficult to filter the distracting sentence after the sentence-level either by the sentence retrieval module or the QA module.

Above findings in both FEVER and HOTPOTQA bring us some important guidelines for MRS: (1) A paragraph-level retrieval module is imperative; (2) A downstream task module is able to undertake a certain amount of noise from sentence-level retrieval; (3) Cascade effects on downstream tasks might be caused by a modification at the paragraph-level retrieval.

Question: Wojtek Wolski played for what team based in the Miami metropolitan area?
GT Answer: Florida Panthers

GT Facts:
[**Florida Panthers,0**]: The Florida Panthers are a professional ice hockey team based in the Miami metropolitan area. $(\textit{P-Score} : 0.99; \textit{S-Score} : 0.98)$
[**Wojtek Wolski,1**]: In the NHL, he has played for the Colorado Avalanche, Phoenix Coyotes, New York Rangers, Florida Panthers, and the Washington Capitals. $(\textit{P-Score} : 0.98; \textit{S-Score} : 0.95)$

Distracting Fact:
[**History of the Miami Dolphins,0**]: The Miami Dolphins are a professional American football franchise based in the Miami metropolitan area. $(\textit{P-Score} : 0.56; \textit{S-Score} : 0.97)$

Wrong Answer : The Miami Dolphins

Figure 3.6: An example with a distracting fact. P-Score and S-Score are the retrieval score at paragraph and sentence level respectively. The full pipeline was able to filter the distracting fact and give the correct answer. The wrong answer in the figure was produced by the system without paragraph-level retrieval module.

3.7 Conclusion

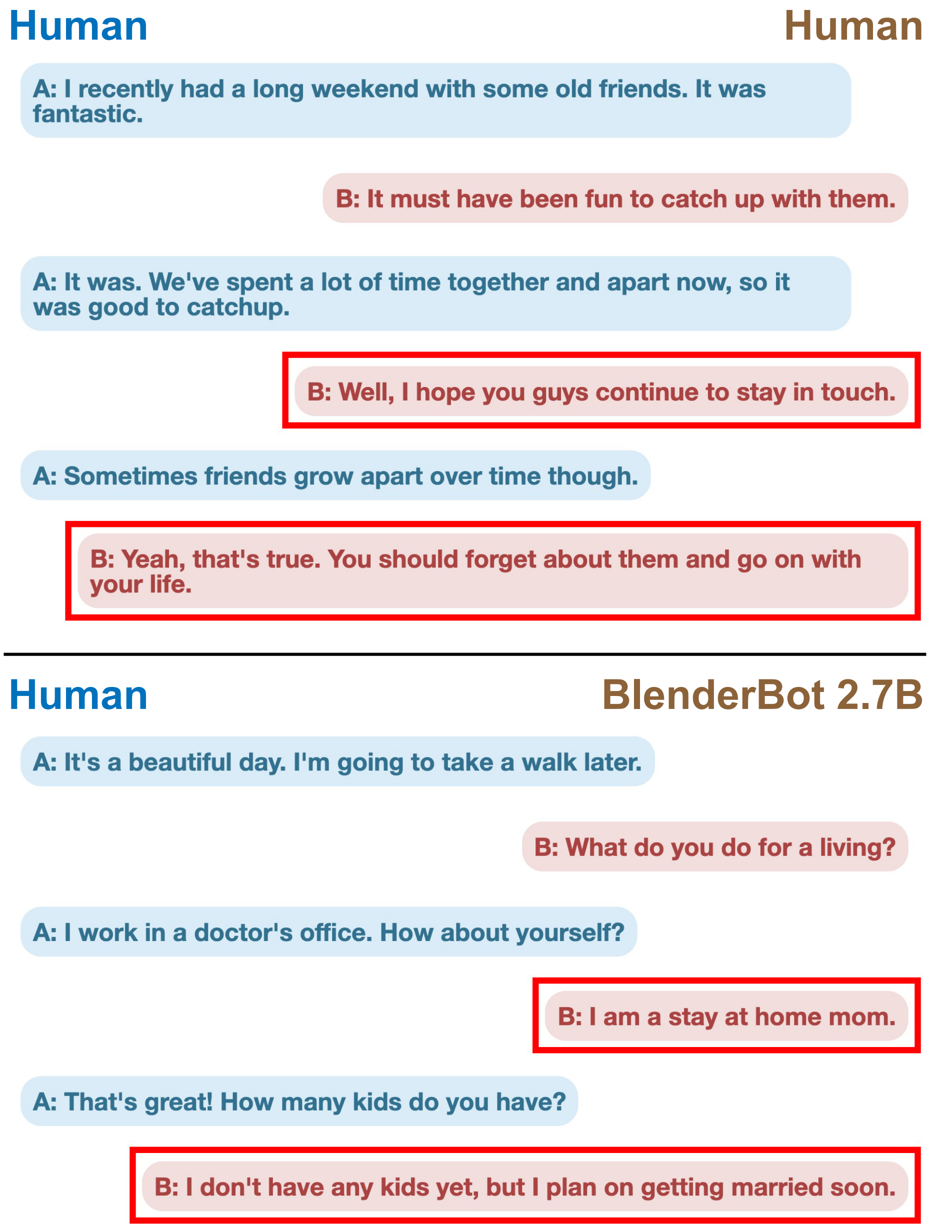
We proposed a simple yet effective hierarchical pipeline system that achieves state-of-the-art results on two MRS tasks. Ablation studies demonstrate the importance of semantic retrieval at both the paragraph and sentence levels in the MRS system. The work can give general guidelines on MRS modeling and inspire future research on the relationship between semantic retrieval and downstream comprehension in a joint setting.

CHAPTER 4: CONSISTENT DIALOGUE MODELING VIA NLI

4.1 Introduction

Recent progress on neural approaches to natural language processing (Devlin et al., 2019a; Brown et al., 2020b), and the availability of large amounts of conversational data (Lowe et al., 2015; Smith et al., 2020) have triggered a resurgent interest on building intelligent open-domain chatbots. Newly developed end-to-end neural bots (Zhang et al., 2020; Adiwardana et al., 2020; Roller et al., 2020) are claimed to be superior to their predecessors (Worsnick, 2018; Zhou et al., 2020) using various human evaluation techniques (See et al., 2019; Li et al., 2019; Adiwardana et al., 2020) that aim to give a more accurate measure of what makes a good conversation. While the success is indisputable, there is still a long way to go before we arrive at human-like open-domain chatbots. For example, it has been shown that open-domain chatbots frequently generate annoying errors (Adiwardana et al., 2020; Roller et al., 2020) and a notorious one among these is the class of contradiction, or consistency errors.

When interacting with chatbots, people carry over many of the same expectations as when interacting with humans (Nass and Moon, 2000). Self-contradictions by these bots (see Fig.4.1, bottom) are often jarring, immediately disrupt the conversational flow, and help support arguments about whether generative models could ever really understand what they are saying at all (Marcus, 2018). From a listener's perspective, such inconsistent bots fail to gain user trust and their long-term communication confidence. From a speaker's perspective, it violates the maxim of quality in Grice's cooperative principles (Grice, 1975) —”Do not say what you believe to be false.” Hence, efforts on reducing contradicting or inconsistent conversations by open-domain chatbots are imperative.



A: I recently had a long weekend with some old friends. It was fantastic.

B: It must have been fun to catch up with them.

A: It was. We've spent a lot of time together and apart now, so it was good to catchup.

B: Well, I hope you guys continue to stay in touch.

A: Sometimes friends grow apart over time though.

B: Yeah, that's true. You should forget about them and go on with your life.

A: It's a beautiful day. I'm going to take a walk later.

B: What do you do for a living?

A: I work in a doctor's office. How about yourself?

B: I am a stay at home mom.

A: That's great! How many kids do you have?

B: I don't have any kids yet, but I plan on getting married soon.

Figure 4.1: Contradictory dialogues contained in our new DECODE dataset. The main train, valid and test sets contain human-written dialogues with deliberate contradictions (example at top), and an out-of-domain test set consists of labeled human-bot dialogues (bottom), involving state-of-the-art bots (Roller et al., 2020).

Prior works (Welleck et al., 2019) characterized the modeling of persona-related consistency as a natural language inference (NLI) problem (Dagan et al., 2005; Bowman et al., 2015b), and constructed a dialog NLI dataset based on Persona-Chat (Zhang et al., 2018), but so far state-of-the-art chatbots (Roller et al., 2020) have not been able to make use of such techniques. Overall, the challenge remains that we are still unable to answer the simple yet important question—"*how good are we at modeling consistency (including persona, logic, causality, etc.) in a general conversation?*". The inability to measure this obscures to what degree building new modules or techniques can in turn help prevent contradicting responses during generation.

Seeking to answer this question, we introduce the DialoguE COntradiction DEtection task (DECODE)[1] and collect a new conversational dataset containing human written dialogues where one of the speakers deliberately contradicts what they have previously said at a certain point during the conversation. We also collect an out-of-distribution (OOD) set of dialogues in human-bot interactive settings which contain human-labeled self-contradictions made by different chatbots.

We then compare a set of state-of-the-art systems, including a standard unstructured approach and a proposed structured approach for utilizing NLI models to detect contradictions. In the unstructured approach, a Transformer NLI model directly takes in the concatenation of all utterances of the input dialogue for prediction, following the paradigm of NLU modeling. In the structured approach, utterances are paired separately before being fed into Transformer NLI models, explicitly taking account of the natural dialogue structure.

Results reveal that: (1) our newly collected dataset is notably more effective at providing supervision for the contradiction detection task than existing NLI data including those aimed at covering the dialogue domain; (2) the structured utterance-based approach for dialogue consistency modeling is more robust in our analysis and more transferable to OOD human-bot conversation than the unstructured approach. This finding challenges the mainstream unstructured approach of simply applying pre-trained Transformer models and expecting them to learn the structure, especially for OOD scenarios which are often the case when incorporating NLU modules into NLG systems, since intermediate in-domain data are scarce.

Finally, with such improvements on the contradiction detection task, we show that our best resulting detector correlates well with human judgments and can be suitable for use as an automatic

[1] DECODE dataset and code are publicly available at XXX.

metric for checking dialogue consistency. We further provide evidence for its usage in improving the consistency of state-of-the-art generative chatbots.

4.2 Related Work

Several prior works on improving dialogue consistency have explored using direct modeling of the dialogue context in generation algorithms. The modeling can be implicit where the dialogue consistency-related information like style (Wang et al., 2017a), topics, or personal facts are maintained in distributed embeddings (Li et al., 2016; Zhang et al., 2019a), neural long-term memories (Bang et al., 2015), hierarchical neural architecture (Serban et al., 2016), latent variables (Serban et al., 2017), topical attention (Dziri et al., 2019b), or even self-learned feature vectors (Zhang et al., 2019b). Some works have grounded generation models on explicit user input (Qian et al., 2018), or designated personas (Zhang et al., 2018). Although, improvements on automatic generation metrics were often shown on guided response generation based on the consistency modeling, the issue of contradiction has never been resolved, nor have generally applicable methods to gauge the consistency improvements been developed. Further, simply scaling models has not made the problem go away, as is evident in the largest chatbots trained such as BlenderBot with up to 9.4B parameter Transformers (Roller et al., 2020).

More similar to our work is utilizing NLI models in dialogue consistency. Dziri et al. (2019a) attempted to use entailment models trained on synthetic datasets for dialogue topic coherence evaluation. Particularly, Welleck et al. (2019) constructed the dialogue NLI dataset and (Li et al., 2020) utilized it to try to reduce inconsistency in generative models via unlikelihood training in a preliminary study that reports perplexity results, but did not measure actual generations or contradiction rates. We note that the dialogue NLI dataset is only semi-automatically generated, with limited coverage of only Persona-chat data (Zhang et al., 2018), whereas our DECODE is human-written and across multiple domains. Our task also involves logical and context-related reasoning beyond personal facts. We show that transfer of DECODE is subsequently more robust than dialogue NLI on both human-human and human-bot chats.

4.3 Task and Data

4.3.1 Dialogue Contradiction Detection

We formalize dialogue contradiction detection as a supervised classification task. The input of the task is a list of utterances $x = \{u_0, u_1, u_2, ..., u_n\}$ representing a dialogue or a dialogue snippet. The output is y, indicating whether the last utterance u_n contradicts any previously conversed information contained in the dialogue $\{u_0, u_1, ..., u_{n-1}\}$, where y can be 0 or 1 corresponding to the non-contradiction and the contradiction label respectively. Preferably, the output should also include a set of indices $\mathbf{I} \subseteq \{0, 1, ..., n-1\}$ representing a subset of $\{u_0, u_1, ..., u_{n-1}\}$ which contain information that is actually contradicted by the last utterance u_n. The extra indices $\mathbf{I}$ output require models to pinpoint the evidence for the contradiction, providing an extra layer of explainability.

4.3.2 Data Collection

Our goal is first to collect training and evaluation data for this task. We thus collect dialogues in which the last utterance contradicts some previous utterances in the dialogue history. To obtain such dialogues, we give annotators dialogue snippets from pre-selected dialogue corpora, and then ask them to continue the conversation by writing one or two utterances such that the last utterance by the last speaker contradicts the dialogue history. We also ask annotators to mark all the utterances in the dialogue history that are involved in the contradiction as supporting evidence. We ask annotators to write contradicting utterances based partly on existing dialogues rather than collecting new dialogue from scratch because the provided dialogues can often convey semantic-rich contexts from different domains and inspire annotators to write more diverse examples. We crowdsource the continuation and annotation data with Amazon Mechanical Turk via ParlAI (Miller et al., 2017).

To ensure data quality, we apply three techniques: (i) an onboarding test every annotator has to pass to contribute examples; (ii) each annotator can only create up to 20 examples; and (iii) all examples in the validation and test set are verified by asking 3 additional workers. More details about annotation are provided in Appendix.

4.3.3 Dataset

We collected 17,713 human-written contradicting dialogues in which 4,121 are verified by 3 annotators. The pre-selected dialogue source corpora are Wizard of Wikipedia (Dinan et al., 2018), EMPATHETICDIALOGUES (Rashkin et al., 2019), Blended Skill Talk (Smith et al., 2020), and ConvAI2 (Dinan et al., 2020), covering various conversational topics. To facilitate the evaluation of consistency modeling on the dialogue contradiction detection classification task, we sample an equal number of non-contradicting dialogues according to the same dialogue length distribution as the contradicting ones from the same dialogue corpus. Then, we make the splits such that the train split contains unverified examples, and dev and test splits only contain verified examples. Each split has balanced labels between contradiction and non-contradiction. The breakdown of each of the dataset sources is shown in Appendix.

Auxiliary (Checklist) Test Sets We further create two auxiliary checklist evaluation sets by transforming the contradiction examples in the original test in two ways such that the ground truth label is either invariant or expected to flip. The two resultant sets serve as diagnostic tests on the behavior, generalization and transferability of our models.

The transformations are: (1) **Add Two Turns (A2T)** We insert a pair of randomly sampled utterances into the dialogue such that the inserted utterances are between the two original contradicting utterances. This gives a new contradicting dialogue with a longer dialogue history; and (2) **Remove Contradicting Turns (RCT)** We remove all the turns (all pairs of utterances) marked as supporting evidence for the contradiction in the dialogue except the last utterance. This results in a new non-contradiction dialogue. The dataset dialogues involve two speakers taking turns speaking. To maintain this structure, for each marked utterance, we remove a pair of utterances that represents a turn.

	Count	**Label**
Main (Train)	27,184	balanced
Main (Dev)	4,026	balanced
Main (Test)	4,216	balanced
Human-Bot (Test)	764	balanced
A2T (Test)	2,079	contradiction
RCT (Test)	2,011	non-contradiction

Table 4.1: DECODE Dataset summary. The first column presents the different dataset types. "Main" is the collected human-written dialogues. "balanced" indicates that the contradiction and non-contradiction labels in that part of the dataset are balanced. A2T and RCT are the auxiliary test sets described in subsection 4.3.3.

Human-Bot Test Set Our main dataset involves human-written dialogues containing contradicting utterances based on human-human dialogues from existing corpora. In practice, to evaluate the response quality of a machine rather than a human in terms of its consistent responses, we care about how well a contradiction detector can perform in human-bot interactive conversations. To that end, we further collect human-bot dialogue data by employing crowdworkers to interact with a diverse set of open-domain bots. These include Poly-encoder (Humeau et al., 2019) based retrieval models, generative models (Roller et al., 2020), unlikelihood trained models (Li et al., 2020), retrieve-and-refine models (Weston et al., 2018; Roller et al., 2020), models either pre-trained on a previously existing Reddit dataset extracted and obtained by a third party that was hosted by pushshift.io (Baumgartner et al., 2020) or fine-tuned on the Blended Skill Talk (BST) dialogue tasks (Smith et al., 2020) – that is, all the dialogue models that are compared in the study in Roller et al. (2020). During the collection, if the bot generates an utterance that contradicts itself, we ask the worker to mark the utterance. In some of the dialogues, workers are explicitly instructed to goad the bots into making contradicting utterances. The final human-bot test set we derive contains 764 dialogues, half of which end with a contradicting utterance by the bot. All the dialogues in the set, with either contradiction or non-contradiction labels, are verified by 3 additional annotators, beside the human who actually talked to the bot.

The auxiliary and human-bot test sets aim to test models' robustness and generalizability beyond the collected human-written test set (Ribeiro et al., 2020; Gardner et al., 2020), and give a more comprehensive analysis of the task. Table 5.2 summarizes the final overall dataset. Examples are provided for each dataset type in Fig. 4.1 and Appendix Table 5.1.

4.4 Models

To model the dialogue consistency task, we first employ some of the techniques used in NLI sequence-to-label modeling, where the input is a pair of textual sequences and the output is a label. The benefit of such modeling is that we can directly make use of existing NLI datasets during training. However, unlike previous work (Welleck et al., 2019) that directly utilized NLI models giving a 3-way output among "entailment", "contradiction", and "neutral", we modify the model with a 2-way output between "contradiction" and "non-contradiction" (either "entailment" or "neutral") labels, as our task is centered around the detection of inconsistency.

More formally, we denote the model as $\hat{y}_{pred} = f_\theta(\mathbf{C}, u)$, where $\hat{y}_{pred}$ is the prediction of the label y, i.e. whether the textual response u contradicts some textual context $\mathbf{C} = \{u_0, u_1, ..., u_{n-1}\}$, and θ are the parameters of the model.

4.4.1 Dialogue Contradiction Detectors

We explore two distinct approaches that propose differing f_θ for the detection prediction problem.

Unstructured Approach. In this approach, we simply concatenate all the previous utterances in the dialogue history to form a single textual context. Then, we apply f_θ to the context and the last utterance to infer the probability of contradiction:

$$\hat{y}_{pred} = f_\theta([u_0, u_1, u_2, ..., u_{n-1}], u_n) \tag{4.1}$$

When concatenating the utterances, we insert special tokens before each utterance to indicate the speaker of that utterance. This is aimed to provide a signal of the dialogue structure to the models. Still, this approach assumes that the model can use these features adequately to learn the underlying structure of the dialogue implicitly during training.

Structured Utterance-based Approach. Since the reasoning crucially depends on the last utterance, in this method we first choose all the utterances by the last speaker to form a set $\mathbf{S}$. We then pair every utterance in the set with the last utterance and feed them one by one into f_θ^{UB}. The final contradiction probability is the maximum over all the outputs:

$$\hat{y}_{pred} = \max\left\{f_\theta^{UB}(u_i, u_n) : u_i \in \mathbf{S}\right\} \tag{4.2}$$

Additionally, the utterance-based approach is able to give a set of utterances as supporting evidence for a contradiction decision by choosing the pairs having contradiction probability higher than a threshold η_e:

$$\mathbf{I} = \left\{i : f_\theta^{UB}(u_i, u_n) > \eta_e\right\} \tag{4.3}$$

This not only gives explanations for its prediction but can also help diagnose the model itself, e.g. we can measure metrics of the model's ability to provide these explanations by comparing them against gold supporting evidence annotations.

One downside of this modeling approach is that it will not be able to capture reasoning between speakers. A case for that would be a pronoun by one speaker might refer to something initiated by the other speaker. Nevertheless, the utterance-based approach explicitly adds an inductive structure bias to learning and inference which we will see can aid its generalization capability.

Thresholding. For both the unstructured and utterance-based approaches, the detection of contradiction is made by comparing $\hat{y}_{pred}$ with a threshold τ and by default τ is 0.5.

4.4.2 Experimental Setup

We study four base pre-trained models variants for f_θ: BERT (Devlin et al., 2019a), Electra (Clark et al., 2019), RoBERTa (Liu et al., 2019b), and BART (Lewis et al., 2020). They repre-

Model	Training Data	MT	MT (Strict)	HB	SE F1
Unstructured Approach					
RoBERTa	All	**97.46**	-	77.09	-
	All - DNLI	97.44	-	73.17	-
	All - ANLI-R3	98.04	-	73.56	-
	All - DECODE	84.42	-	61.91	-
	DNLI	57.19	-	60.34	-
	ANLI-R3	82.21	-	59.69	-
	DECODE	96.85	-	70.03	-
Utterance-based Approach					
RoBERTa	SNLI + MNLI	77.40	47.70	73.17	72.4
	All	94.19	80.08	83.64	**88.5**
	All - DNLI	94.38	**80.93**	81.68	88.4
	All - ANLI-R3	94.07	79.32	82.85	88.4
	All - DECODE	86.67	66.95	77.36	80.6
	DNLI	76.54	63.09	75.26	71.2
	ANLI-R3	81.59	69.11	70.52	74.3
	DECODE	93.19	80.86	**84.69**	87.5
BERT	DECODE	88.88	74.14	75.52	84.3
Electra	DECODE	93.17	81.19	80.76	87.5
BART	DECODE	94.47	80.10	79.19	88.2
Majority					
-	-	50.00	50.00	50.00	48.7

Table 4.2: Test performance on DECODE for various methods. "MT" and "HB" columns show model accuracy on the Main Human-Human Test set and the Human-Bot set, respectively. The "MT (Strict)" column indicates the percentage when both the 2-way contradiction detection and the supporting evidence retrieval exactly match with the ground truth. "SE F1" is F1 score for supporting evidence retrieval. "All" in the "Training Data" column stands for a combination of SNLI, MNLI, DNLI, ANLI-R3, DECODE. "All - DNLI" denotes all the datasets with DNLI removed.

sent the start-of-the-art language representation models and have yielded successes in many NLU tasks. The input format of f_θ follows how these models handle sequence-pairs ($\mathbf{C}$ and u) for classification tasks with padding, separator and other special tokens such as position embeddings and segment features inserted at designated locations accordingly.

We fine-tune f_θ on different combinations of NLI training data including SNLI (Bowman et al., 2015b), MNLI (Williams et al., 2018b), ANLI-R3 (Nie et al., 2020b)[2], DNLI (Welleck et al., 2019), as well as our DECODE Main training set. We convert the 3-way labels of the examples in existing NLI datasets to 2-way, as described before, and θ is optimized using cross-entropy loss. When training f_θ^{UB} in the utterance-based approach using the DECODE training set,

[2] ANLI data is collected in three rounds resulting in three subsets (R1, R2, R3). We only used training data in R3 since it contains some dialogue-related examples.

the input sequences are sampled utterance pairs from the DECODE dialogue. In other scenarios, f_θ or f_θ^{UB} are trained with data treated as in normal NLI training.

The models are evaluated on the test sets described in subsection 4.3.3. For the utterance-based approach, which provides supporting evidence utterances (Equation 4.3), we report F1 on evidence retrieval. We also report a stricter score which evaluates whether both 2-way contradiction detection and supporting evidence retrieval *exactly match* with the ground truth on DECODE Main test.

4.5 Results and Analysis

4.5.1 Performance on Constructed Dataset

Our main results comparing various detectors on DECODE are shown in Table 4.2. We now describe our key observations.

DECODE is notably more effective than other existing NLI data in providing supervision for contradiction detection in dialogue. We found that models trained on DECODE achieve higher accuracy than that of those trained on DNLI or ANLI-R3, on all evaluation sets, with large improvements, e.g. a 12-point jump from the same model training on ANLI-R3 and a 16-point jump from training on DNLI using utterance-based RoBERTa on the DECODE Main test set. Moreover, while training on “All” datasets (SNLI, MNLI, ANLI-R3, DNLI & DECODE) is effective, the removal of DECODE from the training data induces a consequential downgrade on the performance. Training on NLI data which does not cover the dialogue domain, e.g., SNLI+MNLI is even worse, only achieving 77.4% on DECODE Main (Test) vs. 93.19% for DECODE and cannot even reach the majority baseline on the “Main (Test-Strict)”. Further, training on DECODE is also more helpful than DNLI or ANLI-R3 for supporting evidence retrieval. These findings indicate that existing NLI data has limited transferability to the dialogue contradiction detection task despite their coverage of the dialogue domain in addition to other domains and that our DECODE data provides a valuable resource for modeling dialogue consistency and developing data-driven approaches for contradiction detection.

Different pre-training models that perform similarly on the in-domain test set can have very different performance on OOD human-bot dialogue. The last four rows of the table show the results of utterance-based RoBERTa, BERT, Electra, and BART trained on DECODE. We can see that RoBERTa, Electra, and BART got similar in-domain accuracy on DECODE, around 93%-94%. RoBERTa stands out when comparing their performance on the human-bot test set with the highest score of 84.69% across the column and with better performance on supporting evidence retrieval as well. We speculate that this is due to the fact that RoBERTa pre-training data has a broader coverage than Electra and BART. We hope future work on dialogue contradiction detection could explore pre-training models on more dialogue-focused corpora.

The unstructured approach gets higher accuracy on the in-domain test set. A direct comparison between unstructured RoBERTa and utterance-based RoBERTa trained on DECODE reveals that the unstructured approach more often than not gets a higher accuracy than its corresponding utterance-based approach when other experimental setups are kept identical. Noticeably, unstructured RoBERTa trained on all NLI data got a 97.46% score, whereas utterance-based yielded 94.19%. This seemingly indicates that training an unstructured model is able to yield a good representation of the consistency of the dialogue. However, analysis on the human-bot and auxiliary test sets shows that such high accuracy is an over-amplification of the model's real understanding ability, as we discuss next.

The structured utterance-based approach is more robust, and more transferable. Figure 4.2 gives a comparison between utterance-based and unstructured RoBERTa on each of the evaluation sets. We can see that the utterance-based model is able to maintain satisfactory performance across all the sets whereas the unstructured model underperforms at the human-bot and RCT auxiliary test sets with a 34.4% accuracy on RCT compared to 78.4% for utterance-based, in stark contrast to the high performance of the unstructured method on the in-domain DECODE

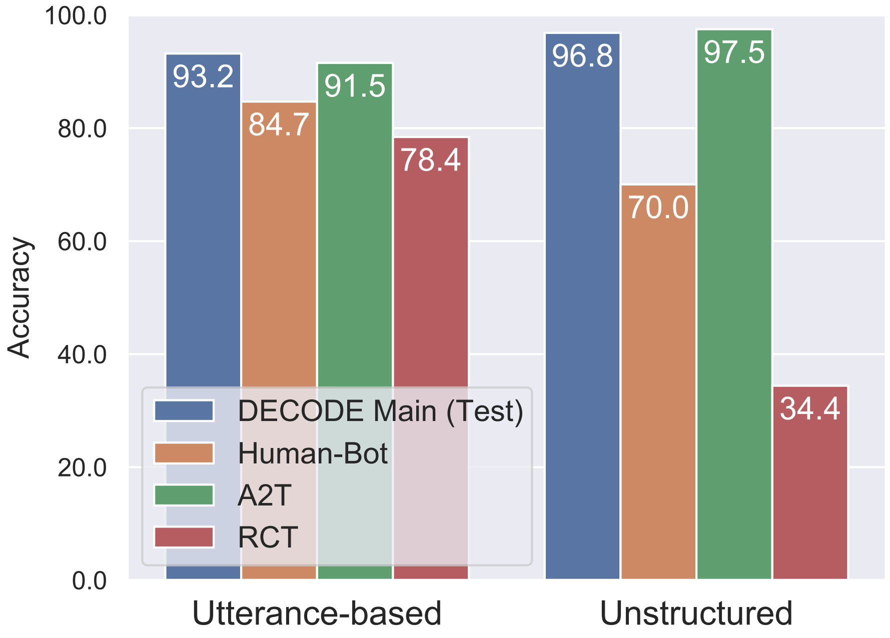


Figure 4.2: Comparison between utterance-based and unstructured approaches of RoBERTa pre-trained, DECODE fine-tuned models on DECODE Main (Test), Human-bot, and auxiliary test sets.

Main test set. This result indicates the unstructured approach overfits on superficial patterns in the DECODE Main training data which are still present due to RCT's construction process.[3] The fact that the utterance-based approach has good transferability to the OOD human-bot test set indicates that injecting the correct inductive structure bias is beneficial for modeling dialogue consistency. We believe this is an interesting result generally for research using Transformers, where there is currently a belief amongst some practitioners that they can just use a standard Transformer and it will learn all the structure correctly on its own. In our setting that is not the case, and we provide a method that can rectify that failing.

In general, there is still much room for improvement. The results in Table 4.2 also demonstrate that the modeling of dialogue consistency is a demanding task. On the contradiction detection task, the best score achieved by the state-of-the-art pre-trained language models on DECODE (Test-Strict) is 80.86% and the best human-bot test score is 84.69%. Considering all the

[3] Overfitting on superficial patterns is a typical issue and open problem in NLU modeling (Nie et al., 2020b).

examples in the test sets are verified by at least 3 annotators, humans are able to swiftly identify such contradictions. This suggests there is a large ability gap between our best automatic detectors and humans. Closing this gap is an important challenge for the community.

4.5.2 Performance in an Interactive Setting

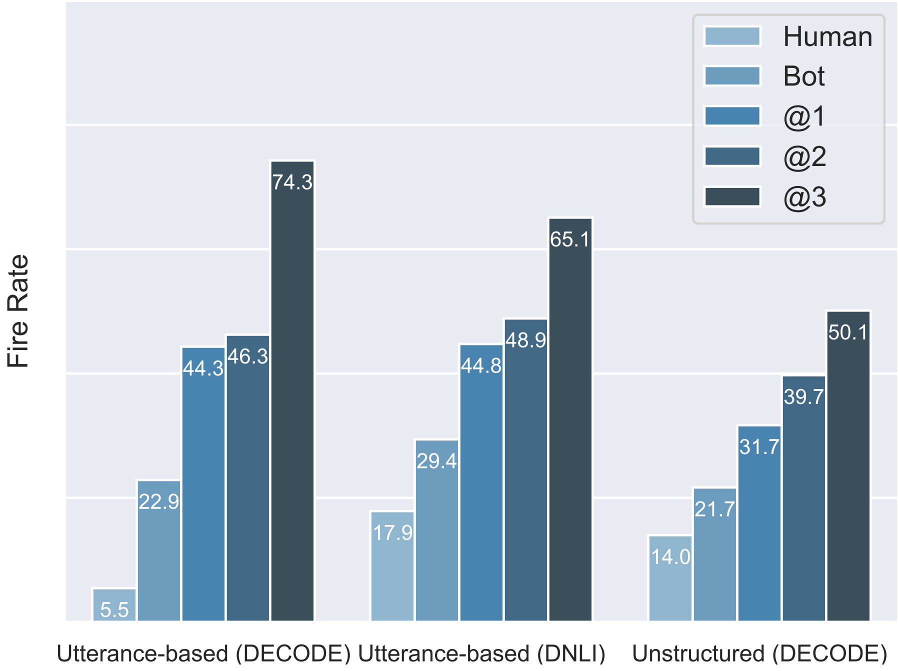


Figure 4.3: The fire rate of RoBERTa models with different setups on utterances belonging to different categories. “Human” and “Bot” stand for utterances by the human or the bot prospectively. “@N” indicates the category where N annotators agreed on the contradiction label. The x-axis indicates different approaches and the text in parentheses denotes the training data.

Model vs. Human Judgment To further understand the detector predictions and how well they might align with human judgments, we consider the Human-Bot data again. We first divide all the utterances into two categories based on whether they are generated by a human or a bot. Then, the bot-generated utterances that have been marked by annotators as contradicting utterances are categorized into three sets based on the number of annotators that agree on the contradiction label. By design, the more annotators that agree on the contradiction label, the more plausible that it is a contradiction. We examine detector model fire rate on the utterances in the 5 different categories and results are shown in Figure 4.3. The fire rate of utterance-based RoBERTa trained on DECODE on human utterances is 5.5% contrasting to the 74.3% on 3-agreed contradicting utterances, whereas the fire rates of unstructured RoBERTa on different categories are more clustered together. This finding demonstrates that our models can discriminate between utterances with a distinct nature, and the model predictions are aligned with human judgments. Moreover, a strong discriminative detector could be a useful tool to stratify utterances.

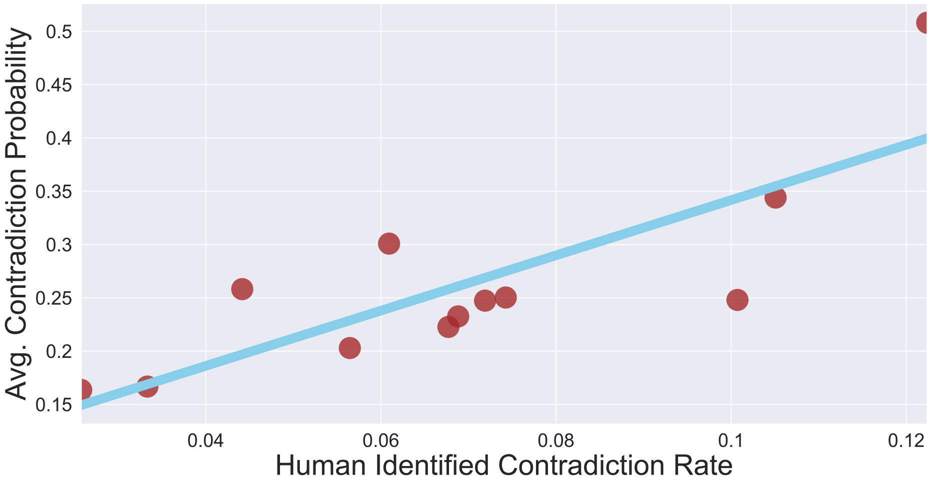


Figure 4.4: The comparison between the average contradiction score by the detector (y-axis) and the human identified contradiction rate (x-axis) on the utterances by different bots, averaged by type of bot. Each point in the plot is a bot which has conversed with humans and produced at least 180 utterances (with some identified as contradictions) in our interactive settings.

Using DECODE as an Automatic Metric The results presented above indicate that the prediction of the detector can easily differentiate between the quality of utterances by humans and the utterances by bots. We further investigate whether it can differentiate the quality of the utterances by different bots and be used as an automatic metric checking generation consistency. We compare the average contradiction score of the detector with the contradiction rate by human judgments on the utterances generated by different classes of model (bots). The bots are the same set of models described in subsection 4.5.2 from which we collected our human-bot dialogue examples. The trend in Figure 4.4 reveals that the scores are positively correlated with human judgments, with a Pearson correlation coefficient of 0.81. We would expect that improvement on the DECODE task will directly increase the correlation between the automatically produced

Model + Decoding Strategy	**DECODE Contradict%**	**Human Contradict%**
Standard generation		
Beam Search	69.7%	84.2%
Top-k ($k = 40$)	42.1%	69.7%
Sample-and-Rank	39.5%	55.3%
DECODE Re-ranking		
Beam Search	46.1%	55.3%
Top-k ($k = 40$)	2.6%	39.5%

Table 4.3: Generation Re-ranking using DECODE vs. standard methods, reporting the contradiction % as flagged by our contradiction detection classifier (i.e., an automatic metric, "DECODE Contradict%") in addition to human judgments ("Human Contradict%").

detection score and human judgments, where use of such an automatic metric can ease the burden on laborious human evaluation of consistency.

4.5.3 Generation Re-ranking

Given a contradiction detector, an obvious question other than using it as an automatic metric, is: can it be used to improve the consistency of dialogue generation models? We consider a very simple way to do that in the state-of-the-art generative model, BlenderBot (BST 2.7B) (Roller et al., 2020). During the decoding phase, for decoding methods that can output multiple hypotheses, we simply rerank the top scoring hypotheses using the contradiction detection classifier. We use our best performing classifier, our utterance-based RoBERTa model with DECODE fine-tuning, and consider three methods of decoding: beam search, top-k sampling (Fan et al., 2018) and sample-and-rank (Adiwardana et al., 2020), and compare the standard and DECODE-reranked decoding methods to each other. For beam search we use the best found parameters from (Roller et al., 2020) which are beam size 10, minimum beam length 20 and beam blocking of 3-grams. For top-k we use $k = 40$. For Sample-and-Rank we use k=40 and 20 samples. We consider the same human-bot dialogue logs as before, but only between Blenderbot BST 2.7B and humans, selecting only contradicting utterances. Table 4.3 presents the results.

Automatic metric using DECODE Using our same DECODE contradiction classifier as the automatic metric, as in subsection 4.5.2, we observe that by re-ranking the beam of beam search (size 10) we can improve the metric. Still, 46.1% of the time the detector flags generations as contradictions (vs. 69.7% without re-ranking). Upon observation of the outputs, this seems to be due to the beam of beam decoding not being diverse enough (Vijayakumar et al., 2016): when the top scoring utterance is flagged as contradicting, many of the other utterances in the beam are similar responses with slight rephrases, and are flagged contradicting as well. Top-k sampling fares much better, where reranking in our test can very often find at least one from the $k = 40$ samples that does not flag the classifier, leaving only a 2.6% contradiction firing rate. We note we expect these numbers are over-optimisticly low because the metric itself is being used to search (re-rank) and evaluate in this case.

Human Judgments The last column of Table 4.3 presents human judgments of the various model generations, judged using the same approach as before with human verifiers, and reporting the percentage of contradictions. We observe similar results to the automatic metric findings. DECODE re-ranking reduces the number of contradictions, particularly for Top-k re-ranking vs. Top-k: testing for significance with a Wilcoxon signed-rank test, we get $p = 0.051$ using two human verifiers and $p = 0.023$ for three verifiers.

4.6 Conclusion

We introduce the DialoguE COntradiction DEtection task (DECODE) and a new conversational dataset containing both human-human and human-bot contradictory dialogues. Training models on DECODE achieves better performance than other existing NLI data by a large margin. We further propose a structured utterance-based approach where utterances are paired before being fed into Transformer NLI models to tackle the dialogue contradiction detection task. We show the superiority of such an approach when transferring to out-of-distribution dialogues compared to a standard unstructured approach representative of mainstream NLU modeling. We further show that our best contradiction detector correlates with human judgments, and provide evidence for its usage in both automatic checking and improving the consistency of state-of-the-art generative chatbots.

CHAPTER 5: ADVERSARIAL NLI

5.1 Introduction

Progress in AI has been driven by, among other things, the development of challenging large-scale benchmarks like ImageNet (Russakovsky et al., 2015) in computer vision, and SNLI (Bowman et al., 2015a), SQuAD (Rajpurkar et al., 2016), and others in natural language processing (NLP). Recently, for natural language understanding (NLU) in particular, the focus has shifted to combined benchmarks like SentEval (Conneau and Kiela, 2018) and GLUE (Wang et al., 2018a), which track model performance on multiple tasks and provide a unified platform for analysis.

With the rapid pace of advancement in AI, however, NLU benchmarks struggle to keep up with model improvement. Whereas it took around 15 years to achieve "near-human performance" on MNIST (LeCun et al., 1998; Cireşan et al., 2012; Wan et al., 2013) and approximately 7 years to surpass humans on ImageNet (Deng et al., 2009; Russakovsky et al., 2015; He et al., 2016b), the GLUE benchmark did not last as long as we would have hoped after the advent of BERT (Devlin et al., 2019b), and rapidly had to be extended into SuperGLUE (Wang et al., 2019). This raises an important question: Can we collect a large benchmark dataset that can last longer?

The speed with which benchmarks become obsolete raises another important question: are current NLU models genuinely as good as their high performance on benchmarks suggests? A growing body of evidence shows that state-of-the-art models learn to exploit spurious statistical patterns in datasets (Gururangan et al., 2018a; Poliak et al., 2018c), instead of learning *meaning* in the flexible and generalizable way that humans do. Given this, human annotators—be they seasoned NLP researchers or non-experts—might easily be able to construct examples that expose model brittleness.

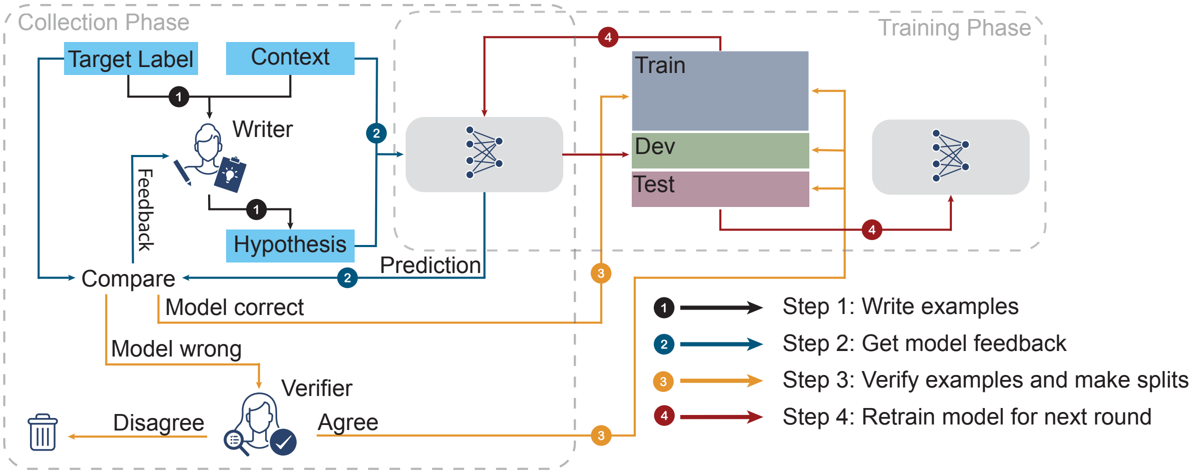


Figure 5.1: Adversarial NLI data collection via human-and-model-in-the-loop enabled training (HAMLET). The four steps make up one round of data collection. In step 3, model-correct examples are included in the training set; development and test sets are constructed solely from model-wrong verified-correct examples.

We propose an iterative, adversarial human-and-model-in-the-loop solution for NLU dataset collection that addresses both benchmark longevity and robustness issues. In the first stage, human annotators devise examples that our current best models cannot determine the correct label for. These resulting hard examples—which should expose additional model weaknesses—can be added to the training set and used to train a stronger model. We then subject the strengthened model to the same procedure and collect weaknesses over several rounds. After each round, we train a new model and set aside a new test set. The process can be iteratively repeated in a never-ending learning (Mitchell et al., 2018) setting, with the model getting stronger and the test set getting harder in each new round. Thus, not only is the resultant dataset harder than existing benchmarks, but this process also yields a “moving post” dynamic target for NLU systems, rather than a static benchmark that will eventually saturate.

Our approach draws inspiration from recent efforts that gamify collaborative training of machine learning agents over multiple rounds (Yang et al., 2017) and pit “builders” against “breakers” to learn better models (Ettinger et al., 2017). Recently, Dinan et al. (2019) showed that such an approach can be used to make dialogue safety classifiers more robust. Here, we focus on natural language inference (NLI), arguably the most canonical task in NLU. We collected three rounds of data, and call our new dataset Adversarial NLI (ANLI).

Our contributions are as follows: 1) We introduce a novel human-and-model-in-the-loop dataset, consisting of three rounds that progressively increase in difficulty and complexity, that includes annotator-provided explanations. 2) We show that training models on this new dataset leads to state-of-the-art performance on a variety of popular NLI benchmarks. 3) We provide a detailed analysis of the collected data that sheds light on the shortcomings of current models, categorizes the data by inference type to examine weaknesses, and demonstrates good performance on NLI stress tests. The ANLI dataset is available at github.com/facebookresearch/anli/. A demo is available at adversarialnli.com.

Context	Hypothesis	Reason	Round	Labels			Annotations
				orig.	pred.	valid.	
Roberto Javier Mora García (c. 1962 – 16 March 2004) was a Mexican journalist and editorial director of "El Mañana", a newspaper based in Nuevo Laredo, Tamaulipas, Mexico. He worked for a number of media outlets in Mexico, including the "El Norte" and "El Diario de Monterrey", prior to his assassination.	Another individual laid waste to Roberto Javier Mora Garcia.	The context states that Roberto Javier Mora Garcia was assassinated, so another person had to have "laid waste to him." The system most likely had a hard time figuring this out due to it not recognizing the phrase "laid waste."	A1 (Wiki)	E	N	E E	Lexical (assassination, laid waste), Tricky (Presupposition), Standard (Idiom)
A melee weapon is any weapon used in direct hand-to-hand combat; by contrast with ranged weapons which act at a distance. The term "melee" originates in the 1640s from the French word "mělée", which refers to hand-to-hand combat, a close quarters battle, a brawl, a confused fight, etc. Melee weapons can be broadly divided into three categories	Melee weapons are good for ranged and hand-to-hand combat.	Melee weapons are good for hand to hand combat, but NOT ranged.	A2 (Wiki)	C	E	C N C	Standard (Conjunction), Tricky (Exhaustification), Reasoning (Facts)
If you can dream it, you can achieve it—unless you're a goose trying to play a very human game of rugby. In the video above, one bold bird took a chance when it ran onto a rugby field mid-play. Things got dicey when it got into a tussle with another player, but it shook it off and kept right on running. After the play ended, the players escorted the feisty goose off the pitch. It was a risky move, but the crowd chanting its name was well worth it.	The crowd believed they knew the name of the goose running on the field.	Because the crowd was chanting its name, the crowd must have believed they knew the goose's name. The word "believe" may have made the system think this was an ambiguous statement.	A3 (News)	E	N	E E	Reasoning (Facts), Reference (Coreference)

Table 5.1: Examples from development set. 'An' refers to round number, 'orig.' is the original annotator's gold label, 'pred.' is the model prediction, 'valid.' are the validator labels, 'reason' was provided by the original annotator, 'Annotations' are the tags determined by an linguist expert annotator.

5.2 Dataset collection

The primary aim of this work is to create a new large-scale NLI benchmark on which current state-of-the-art models fail. This constitutes a new target for the field to work towards, and can elucidate model capabilities and limitations. As noted, however, static benchmarks do not last very long these days. If continuously deployed, the data collection procedure we introduce here can pose a dynamic challenge that allows for never-ending learning.

5.2.1 HAMLET

To paraphrase the great bard (Shakespeare, 1603), *there is something rotten in the state of the art*. We propose *Human-And-Model-in-the-Loop Enabled Training* (HAMLET), a training procedure to automatically mitigate problems with current dataset collection procedures (see Figure 5.1).

In our setup, our starting point is a *base model*, trained on NLI data. Rather than employing automated adversarial methods, here the model's "adversary" is a human annotator. Given a *context* (also often called a "premise" in NLI), and a desired *target label*, we ask the human *writer* to provide a *hypothesis* that fools the model into misclassifying the label. One can think of the writer as a "white hat" hacker, trying to identify vulnerabilities in the system. For each human-generated example that is misclassified, we also ask the writer to provide a *reason* why they believe it was misclassified.

For examples that the model misclassified, it is necessary to verify that they are actually correct —i.e., that the given context-hypothesis pairs genuinely have their specified target label. The best way to do this is to have them checked by another human. Hence, we provide the example to human *verifiers*. If two human verifiers agree with the writer, the example is considered a good example. If they disagree, we ask a third human verifier to break the tie. If there is still disagreement between the writer and the verifiers, the example is discarded. If the verifiers disagree, they can overrule the original target label of the writer.

Once data collection for the current round is finished, we construct a new training set from the collected data, with accompanying development and test sets, which are constructed solely from verified correct examples. The test set was further restricted so as to: 1) include pairs from "exclusive" annotators who are never included in the training data; and 2) be balanced by label classes (and genres, where applicable). We subsequently train a *new model* on this and other existing data, and repeat the procedure.

5.2.2 Annotation details

We employed Mechanical Turk workers with qualifications and collected hypotheses via the ParlAI[1] framework. Annotators are presented with a context and a target label—either 'entailment', 'contradiction', or 'neutral'—and asked to write a hypothesis that corresponds to the label. We phrase the label classes as "definitely correct", "definitely incorrect", or "neither definitely correct nor definitely incorrect" given the context, to make the task easier to grasp. Model predictions are obtained for the context and submitted hypothesis pair. The probability of each label is shown to the worker as feedback. If the model prediction was incorrect, the job is complete. If not, the worker continues to write hypotheses for the given (context, target-label) pair until the model predicts the label incorrectly or the number of tries exceeds a threshold (5 tries in the first round, 10 tries thereafter).

To encourage workers, payments increased as rounds became harder. For hypotheses that the model predicted incorrectly, and that were verified by other humans, we paid an additional bonus on top of the standard rate.

5.2.3 Three Rounds of Collection

Round 1 For the first round, we used a BERT-Large model (Devlin et al., 2019b) trained on a concatenation of SNLI (Bowman et al., 2015a) and MNLI (Williams et al., 2018c), and selected the best-performing model we could train as the starting point for our dataset collection procedure. For Round 1 contexts, we randomly sampled short multi-sentence passages from Wikipedia (of 250-600 characters) from the manually curated HotpotQA training set (Yang et al., 2018). Contexts are either ground-truth contexts from that dataset, or they are Wikipedia passages retrieved using TF-IDF (Chen et al., 2017b) based on a HotpotQA question.

[1] `https://parl.ai/`

Dataset	Genre	Context	Train / Dev / Test	Model error rate		Tries	Time (sec.)
				Unverified	Verified	mean/median per verified ex.	
A1	Wiki	2,080	16,946 / 1,000 / 1,000	29.68%	18.33%	3.4 / 2.0	199.2 / 125.2
A2	Wiki	2,694	45,460 / 1,000 / 1,000	16.59%	8.07%	6.4 / 4.0	355.3 / 189.1
A3	Various	6,002	100,459 / 1,200 / 1,200	17.47%	8.60%	6.4 / 4.0	284.0 / 157.0
	(Wiki subset)	1,000	19,920 / 200 / 200	14.79%	6.92%	7.4 / 5.0	337.3 / 189.6
ANLI	Various	10,776	162,865 / 3,200 / 3,200	18.54%	9.52%	5.7 / 3.0	282.9 / 156.3

Table 5.2: Dataset statistics: 'Model error rate' is the percentage of examples that the model got wrong; 'unverified' is the overall percentage, while 'verified' is the percentage that was verified by at least 2 human annotators.

Round 2 For the second round, we used a more powerful RoBERTa model (Liu et al., 2019b) trained on SNLI, MNLI, an NLI-version[2] of FEVER (Thorne et al., 2018b), and the training data from the previous round (A1). After a hyperparameter search, we selected the model with the best performance on the A1 development set. Then, using the hyperparameters selected from this search, we created a final set of models by training several models with different random seeds. During annotation, we constructed an ensemble by randomly picking a model from the model set as the adversary each turn. This helps us avoid annotators exploiting vulnerabilities in one single model. A new non-overlapping set of contexts was again constructed from Wikipedia via HotpotQA using the same method as Round 1.

Round 3 For the third round, we selected a more diverse set of contexts, in order to explore robustness under domain transfer. In addition to contexts from Wikipedia for Round 3, we also included contexts from the following domains: News (extracted from Common Crawl), fiction (extracted from StoryCloze (Mostafazadeh et al., 2016) and CBT (Hill et al., 2015)), formal spoken text (excerpted from court and presidential debate transcripts in the Manually Annotated Sub-Corpus (MASC) of the Open American National Corpus[3]), and causal or procedural text, which describes sequences of events or actions, extracted from WikiHow. Finally, we also collected annotations using the longer contexts present in the GLUE RTE training data, which came

[2] The NLI version of FEVER pairs claims with evidence retrieved by Nie et al. (2019a) as (context, hypothesis) inputs.

[3] `anc.org/data/masc/corpus/`

from the RTE5 dataset (Bentivogli et al., 2009). We trained an even stronger RoBERTa ensemble by adding the training set from the second round (A2) to the training data.

5.2.4 Comparing with other datasets

The ANLI dataset, comprising three rounds, improves upon previous work in several ways. First, and most obviously, the dataset is collected to be more difficult than previous datasets, by design. Second, it remedies a problem with SNLI, namely that its contexts (or premises) are very short, because they were selected from the image captioning domain. We believe longer contexts should naturally lead to harder examples, and so we constructed ANLI contexts from longer, multi-sentence source material.

Following previous observations that models might exploit spurious biases in NLI hypotheses, (Gururangan et al., 2018a; Poliak et al., 2018c), we conduct a study of the performance of hypothesis-only models on our dataset. We show that such models perform poorly on our test sets.

With respect to data generation with naïve annotators, Geva et al. (2019) noted that models can pick up on annotator bias, modelling annotator artefacts rather than the intended reasoning phenomenon. To counter this, we selected a subset of annotators (i.e., the "exclusive" workers) whose data would only be included in the test set. This enables us to avoid overfitting to the writing style biases of particular annotators, and also to determine how much individual annotator bias is present for the main portion of the data. Examples from each round of dataset collection are provided in Table 5.1.

Furthermore, our dataset poses new challenges to the community that were less relevant for previous work, such as: can we improve performance online without having to train a new model from scratch every round, how can we overcome catastrophic forgetting, how do we deal with mixed model biases, etc. Because the training set includes examples that the model got right but were not verified, learning from noisy and potentially unverified data becomes an additional interesting challenge.

Model	**Training Data**	A1	A2	A3	ANLI	ANLI-E	SNLI	MNLI-m/-mm
BERT	S,M$^{\star 1}$	00.0	28.9	28.8	19.8	19.9	91.3	86.7 / 86.4
	+A1	44.2	32.6	29.3	35.0	34.2	91.3	86.3 / 86.5
	+A1+A2	57.3	45.2	33.4	44.6	43.2	90.9	86.3 / 86.3
	+A1+A2+A3	57.2	49.0	46.1	50.5	46.3	90.9	85.6 / 85.4
	S,M,F,ANLI	57.4	48.3	43.5	49.3	44.2	90.4	86.0 / 85.8
XLNet	S,M,F,ANLI	67.6	50.7	48.3	55.1	52.0	91.8	89.6 / 89.4
RoBERTa	S,M	47.6	25.4	22.1	31.1	31.4	92.6	90.8 / 90.6
	+F	54.0	24.2	22.4	32.8	33.7	92.7	90.6 / 90.5
	+F+A1$^{\star 2}$	68.7	19.3	22.0	35.8	36.8	92.8	90.9 / 90.7
	+F+A1+A2$^{\star 3}$	71.2	44.3	20.4	43.7	41.4	92.9	91.0 / 90.7
	S,M,F,ANLI	73.8	48.9	44.4	53.7	49.7	92.6	91.0 / 90.6

Table 5.3: Model Performance. 'S' refers to SNLI, 'M' to MNLI dev (-m=matched, -mm=mismatched), and 'F' to FEVER; 'A1–A3' refer to the rounds respectively and 'ANLI' refers to A1+A2+A3, '-E' refers to test set examples written by annotators exclusive to the test set. Datasets marked '$\star^n$' were used to train the base model for round n, and their performance on that round is underlined (A2 and A3 used ensembles, and hence have non-zero scores).

5.3 Dataset statistics

The dataset statistics can be found in Table 5.2. The number of examples we collected increases per round, starting with approximately 19k examples for Round 1, to around 47k examples for Round 2, to over 103k examples for Round 3. We collected more data for later rounds not only because that data is likely to be more interesting, but also simply because the base model is better and so annotation took longer to collect good, verified correct examples of model vulnerabilities.

For each round, we report the model error rate, both on verified and unverified examples. The unverified model error rate captures the percentage of examples where the model disagreed with the writer's target label, but where we are not (yet) sure if the example is correct. The verified model error rate is the percentage of model errors from example pairs that other annotators confirmed the correct label for. Note that error rate is a useful way to evaluate model quality: the lower the model error rate—assuming constant annotator quality and context-difficulty—the better the model.

We observe that model error rates decrease as we progress through rounds. In Round 3, where we included a more diverse range of contexts from various domains, the overall error rate went slightly up compared to the preceding round, but for Wikipedia contexts the error rate decreased substantially. While for the first round roughly 1 in every 5 examples were verified model errors, this quickly dropped over consecutive rounds, and the overall model error rate is less than 1 in 10. On the one hand, this is impressive, and shows how far we have come with just three rounds. On the other hand, it shows that we still have a long way to go if even untrained annotators can fool ensembles of state-of-the-art models with relative ease.

Table 5.2 also reports the average number of "tries", i.e., attempts made for each context until a model error was found (or the number of possible tries is exceeded), and the average time this took (in seconds). Again, these metrics are useful for evaluating model quality: observe that the average number of tries and average time per verified error both go up with later rounds. This demonstrates that the rounds are getting increasingly more difficult.

The dataset statistics can be found in Table 5.2. The number of examples we collected increases per round, starting with approximately 19k examples for Round 1, to around 47k examples for Round 2, to over 103k examples for Round 3. We collected more data for later rounds not only because that data is likely to be more interesting, but also simply because the base model is better and so annotation took longer to collect good, verified correct examples of model vulnerabilities.

For each round, we report the model error rate, both on verified and unverified examples. The unverified model error rate captures the percentage of examples where the model disagreed with the writer's target label, but where we are not (yet) sure if the example is correct. The verified model error rate is the percentage of model errors from example pairs that other annotators confirmed the correct label for. Note that error rate is a useful way to evaluate model quality: the lower the model error rate—assuming constant annotator quality and context-difficulty—the better the model.

We observe that model error rates decrease as we progress through rounds. In Round 3, where we included a more diverse range of contexts from various domains, the overall error rate went slightly up compared to the preceding round, but for Wikipedia contexts the error rate decreased substantially. While for the first round roughly 1 in every 5 examples were verified model errors, this quickly dropped over consecutive rounds, and the overall model error rate is less than 1 in 10. On the one hand, this is impressive, and shows how far we have come with just three rounds. On the other hand, it shows that we still have a long way to go if even untrained annotators can fool ensembles of state-of-the-art models with relative ease.

Table 5.2 also reports the average number of “tries”, i.e., attempts made for each context until a model error was found (or the number of possible tries is exceeded), and the average time this took (in seconds). Again, these metrics are useful for evaluating model quality: observe that the average number of tries and average time per verified error both go up with later rounds. This demonstrates that the rounds are getting increasingly more difficult.

5.4 Results

Table 5.3 reports the main results. In addition to BERT (Devlin et al., 2019b) and RoBERTa (Liu et al., 2019b), we also include XLNet (Yang et al., 2019) as an example of a strong, but different, model architecture. We show test set performance on the ANLI test sets per round, the total ANLI test set, and the exclusive test subset (examples from test-set-exclusive workers). We also show accuracy on the SNLI test set and the MNLI development set (for the purpose of comparing between different model configurations across table rows). In what follows, we discuss our observations.

Base model performance is low. Notice that the base model for each round performs very poorly on that round’s test set. This is the expected outcome: For round 1, the base model gets the entire test set wrong, by design. For rounds 2 and 3, we used an ensemble, so performance is not necessarily zero. However, as it turns out, performance still falls well below chance[4], indicating that workers did not find vulnerabilities specific to a single model, but generally applicable ones for that model class.

Rounds become increasingly more difficult. As already foreshadowed by the dataset statistics, round 3 is more difficult (yields lower performance) than round 2, and round 2 is more difficult than round 1. This is true for all model architectures.

Training on more rounds improves robustness. Generally, our results indicate that training on more rounds improves model performance. This is true for all model architectures. Simply training on more "normal NLI" data would not help a model be robust to adversarial attacks, but our data actively helps mitigate these.

RoBERTa achieves state-of-the-art performance... We obtain state of the art performance on both SNLI and MNLI with the RoBERTa model finetuned on our new data. The RoBERTa paper (Liu et al., 2019b) reports a score of 90.2 for both MNLI-matched and -mismatched dev, while we obtain 91.0 and 90.7. The state of the art on SNLI is currently held by MT-DNN (Liu et al., 2019a), which reports 91.6 compared to our 92.9.

...but is outperformed when it is base model. However, the base (RoBERTa) models for rounds 2 and 3 are outperformed by both BERT and XLNet (rows 5, 6 and 10). This shows that annotators found examples that RoBERTa generally struggles with, which cannot be mitigated by more examples alone. It also implies that BERT, XLNet, and RoBERTa all have different weaknesses, possibly as a function of their training data (BERT, XLNet and RoBERTa were trained on different data sets, which might or might not have contained information relevant to the weaknesses).

Continuously augmenting training data does not downgrade performance. Even though ANLI training data is different from SNLI and MNLI, adding it to the training set does not harm performance on those tasks. Our results (see also rows 2-3 of Table 5.6) suggest the method could successfully be applied for multiple additional rounds.

[4] Chance is at 33%, since the test set labels are balanced.

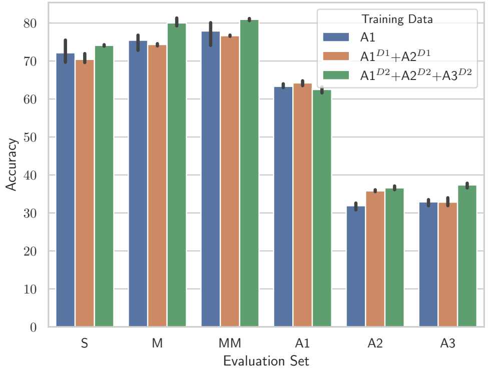


Figure 5.2: RoBERTa performance on dev, with A1–3 downsampled s.t. $|\text{A1}^{D1}|$=$|\text{A2}^{D1}|$=$\frac{1}{2}|\text{A1}|$ and $|\text{A1}^{D2}|$=$|\text{A2}^{D2}|$=$|\text{A3}^{D2}|$=$\frac{1}{3}|\text{A1}|$.

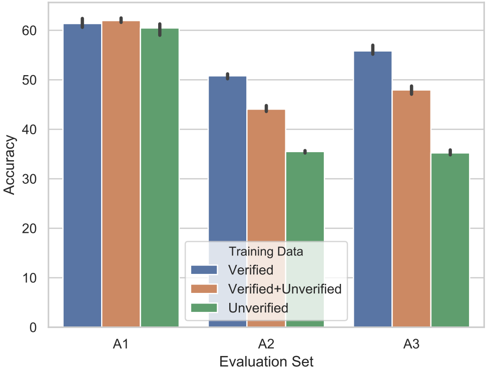


Figure 5.3: Comparison of verified, unverified and combined data, where data sets are downsampled to ensure equal training sizes.

Exclusive test subset difference is small. We included an exclusive test subset (ANLI-E) with examples from annotators never seen in training, and find negligible differences, indicating that our models do not over-rely on annotator's writing styles.

5.4.1 The effectiveness of adversarial training

We examine the effectiveness of the adversarial training data in two ways. First, we sample from respective datasets to ensure exactly equal amounts of training data. Table 5.5 shows that the adversarial data improves performance, including on SNLI and MNLI when we replace part of those datasets with the adversarial data. This suggests that the adversarial data is more data efficient than "normally collected" data. Figure 5.2 shows that adversarial data collected in later rounds is of higher quality and more data-efficient.

Model	**SNLI-Hard**	**NLI Stress Tests**					
		AT (m/mm)	NR	LN (m/mm)	NG (m/mm)	WO (m/mm)	SE (m/mm)
Previous models	72.7	14.4 / 10.2	28.8	58.7 / 59.4	48.8 / 46.6	50.0 / 50.2	58.3 / 59.4
BERT (All)	82.3	75.0 / 72.9	65.8	84.2 / 84.6	64.9 / 64.4	61.6 / 60.6	78.3 / 78.3
XLNet (All)	83.5	88.2 / 87.1	85.4	87.5 / 87.5	59.9 / 60.0	68.7 / 66.1	84.3 / 84.4
RoBERTa (S+M+F)	84.5	81.6 / 77.2	62.1	88.0 / 88.5	61.9 / 61.9	67.9 / 66.2	86.2 / 86.5
RoBERTa (All)	84.7	85.9 / 82.1	80.6	88.4 / 88.5	62.2 / 61.9	67.4 / 65.6	86.3 / 86.7

Table 5.4: Model Performance on NLI stress tests (tuned on their respective dev. sets). All=S+M+F+ANLI. AT='Antonym'; 'NR'=Numerical Reasoning; 'LN'=Length; 'NG'=Negation; 'WO'=Word Overlap; 'SE'=Spell Error. Previous models refers to the Naik et al. (2018) implementation of Conneau et al. (2017a) for the Stress Tests, and to the Gururangan et al. (2018b) implementation of Gong et al. (2018) for SNLI-Hard.

Train Data	A1	A2	A3	S	M-m/mm
SM^{D1}+SM^{D2}	45.1	26.1	27.1	**92.5**	89.8/**89.7**
SM^{D1}+A	**72.6**	**42.9**	**42.0**	92.3	**90.3**/89.6
SM	48.0	24.8	31.1	93.2	90.8/90.6
SM^{D3}+A	**73.3**	**42.4**	**40.5**	**93.3**	**90.8/90.7**

Table 5.5: RoBERTa performance on dev set with different training data. S=SNLI, M=MNLI, A=A1+A2+A3. 'SM' refers to combined S and M training set. D1, D2, D3 means down-sampling SM s.t. $|\text{SM}^{D2}|=|\text{A}|$ and $|\text{SM}^{D3}|+|\text{A}|=|\text{SM}|$. Therefore, training sizes are identical in every pair of rows.

Second, we compared verified correct examples of model vulnerabilities (examples that the model got wrong and were verified to be correct) to unverified ones. Figure 5.3 shows that the verified correct examples are much more valuable than the unverified examples, especially in the later rounds (where the latter drops to random).

5.4.2 Stress Test Results

We also test models on two recent hard NLI test sets: SNLI-Hard (Gururangan et al., 2018a) and the NLI stress tests (Naik et al., 2018) . The results are in Table 5.4. We observe that all our models outperform the models presented in original papers for these common stress tests. The RoBERTa models perform best on SNLI-Hard and achieve accuracy levels in the high 80s on the 'antonym' (AT), 'numerical reasoning' (NR), 'length' (LN), 'spelling error'(SE) sub-datasets,

Train Data	A1	A2	A3	S	M-m/mm
ALL	73.8	48.9	44.4	92.6	91.0/90.6
S+M	47.6	25.4	22.1	92.6	90.8/90.6
ANLI-Only	71.3	43.3	43.0	83.5	86.3/86.5
ALL^H	49.7	46.3	42.8	71.4	60.2/59.8
S+M^H	33.1	29.4	32.2	71.8	62.0/62.0
ANLI-Only^H	51.0	42.6	41.5	47.0	51.9/54.5

Table 5.6: Performance of RoBERTa with different data combinations. ALL=S,M,F,ANLI. Hypothesis-only models are marked H where they are trained and tested with only hypothesis texts.

and show marked improvement on both ‘negation’ (NG), and ‘word overlap’ (WO). Training on ANLI appears to be particularly useful for the AT, NR, NG and WO stress tests.

5.4.3 Hypothesis-only results

For SNLI and MNLI, concerns have been raised about the propensity of models to pick up on spurious artifacts that are present just in the hypotheses (Gururangan et al., 2018a; Poliak et al., 2018c). Here, we compare full models to models trained only on the hypothesis (marked H). Table 5.6 reports results on ANLI, as well as on SNLI and MNLI. The table shows that hypothesis-only models perform poorly on ANLI[5], and obtain good performance on SNLI and MNLI. Hypothesis-only performance decreases over rounds for ANLI.

We observe that in rounds 2 and 3, RoBERTa is not much better than hypothesis-only. This could mean two things: either the test data is very difficult, or the training data is not good. To rule out the latter, we trained only on ANLI ($\sim$163k training examples): RoBERTa matches BERT when trained on the much larger, fully in-domain SNLI+MNLI combined dataset (943k training examples) on MNLI, with both getting $\sim$86 (the third row in Table 5.6). Hence, this shows that the test sets are so difficult that state-of-the-art models cannot outperform a hypothesis-only prior.

[5] Obviously, without manual intervention, some bias remains in how people phrase hypotheses—e.g., contradiction might have more negation—which explains why hypothesis-only performs slightly above chance when trained on ANLI.

Round	Numerical & Quant.	Reference & Names	Standard	Lexical	Tricky	Reasoning & Facts	Quality
A1	38%	13%	18%	13%	22%	53%	4%
A2	32%	20%	21%	21%	20%	59%	3%
A3	10%	18%	27%	27%	27%	63%	3%
Average	27%	17%	22%	22%	23%	58%	3%

Table 5.7: Analysis of 500 development set examples per round and on average.

5.5 Linguistic analysis

We explore the types of inferences that fooled models by manually annotating 500 examples from each round's development set. A dynamically evolving dataset offers the unique opportunity to track how model error rates change over time. Since each round's development set contains only verified examples, we can investigate two interesting questions: which types of inference do writers employ to fool the models, and are base models differentially sensitive to different types of reasoning?

The results are summarized in Table 5.7. We devised an inference ontology containing six types of inference: Numerical & Quantitative (i.e., reasoning about cardinal and ordinal numbers, inferring dates and ages from numbers, etc.), Reference & Names (coreferences between pronouns and forms of proper names, knowing facts about name gender, etc.), Standard Inferences (conjunctions, negations, cause-and-effect, comparatives and superlatives etc.), Lexical Inference (inferences made possible by lexical information about synonyms, antonyms, etc.), Tricky Inferences (wordplay, linguistic strategies such as syntactic transformations/reorderings, or inferring writer intentions from contexts), and reasoning from outside knowledge or additional facts (e.g., "You can't reach the sea directly from Rwanda"). The quality of annotations was also tracked; if a pair was ambiguous or a label debatable (from the expert annotator's perspective), it was flagged. Quality issues were rare at 3-4% per round. Any one example can have multiple types, and every example had at least one tag.

We observe that both round 1 and 2 writers rely heavily on numerical and quantitative reasoning in over 30% of the development set—the percentage in A2 (32%) dropped roughly 6% from A1 (38%)—while round 3 writers use numerical or quantitative reasoning for only 17%. The majority of numerical reasoning types were references to cardinal numbers that referred to dates and ages. Inferences predicated on references and names were present in about 10% of rounds 1 & 3 development sets, and reached a high of 20% in round 2, with coreference featuring prominently. Standard inference types increased in prevalence as the rounds increased, ranging from 18%–27%, as did 'Lexical' inferences (increasing from 13%–31%). The percentage of sentences relying on reasoning and outside facts remains roughly the same, in the mid-50s, perhaps slightly increasing over the rounds. For round 3, we observe that the model used to collect it appears to be more susceptible to Standard, Lexical, and Tricky inference types. This finding is compatible with the idea that models trained on adversarial data perform better, since annotators seem to have been encouraged to devise more creative examples containing harder types of inference in order to stump them.

5.6 Related work

Bias in datasets Machine learning methods are well-known to pick up on spurious statistical patterns. For instance, in the first visual question answering dataset (Antol et al., 2015), biases like "2" being the correct answer to 39% of the questions starting with "how many" allowed learning algorithms to perform well while ignoring the visual modality altogether (Jabri et al., 2016; Goyal et al., 2017). In NLI, Gururangan et al. (2018b), Poliak et al. (2018b) and Tsuchiya (2018) showed that hypothesis-only baselines often perform far better than chance. NLI systems can often be broken merely by performing simple lexical substitutions (Glockner et al., 2018), and struggle with quantifiers (Geiger et al., 2018) and certain superficial syntactic properties (McCoy et al., 2019).

In question answering, Kaushik and Lipton (2018) showed that question- and passage-only models can perform surprisingly well, while Jia and Liang (2017) added adversarially constructed sentences to passages to cause a drastic drop in performance. Many tasks do not actually require sophisticated linguistic reasoning, as shown by the surprisingly good performance of random encoders (Wieting and Kiela, 2019). Similar observations were made in machine translation

(Belinkov and Bisk, 2017) and dialogue (Sankar et al., 2019). Machine learning also has a tendency to overfit on static targets, even if that does not happen deliberately (Recht et al., 2018). In short, the field is rife with dataset bias and papers trying to address this important problem. This work presents a potential solution: if such biases exist, they will allow humans to fool the models, resulting in valuable training examples until the bias is mitigated.

Dynamic datasets. Bras et al. (2020) proposed AFLite, an approach for avoiding spurious biases through adversarial filtering, which is a model-in-the-loop approach that iteratively probes and improves models. Kaushik et al. (2019) offer a causal account of spurious patterns, and counterfactually augment NLI datasets by editing examples to break the model. That approach is human-in-the-loop, using humans to find problems with one single model. In this work, we employ both human and model-based strategies iteratively, in a form of human-and-model-in-the-loop training, to create completely *new* examples, in a potentially never-ending loop (Mitchell et al., 2018).

Human-and-model-in-the-loop training is not a new idea. Mechanical Turker Descent proposes a gamified environment for the collaborative training of grounded language learning agents over multiple rounds. The "Build it Break it Fix it" strategy in the security domain (Ruef et al., 2016) has been adapted to NLP (Ettinger et al., 2017) as well as dialogue safety (Dinan et al., 2019). The QApedia framework (Kratzwald and Feuerriegel, 2019) continuously refines and updates its content repository using humans in the loop, while human feedback loops have been used to improve image captioning systems (Ling and Fidler, 2017). Wallace et al. (2019) leverage trivia experts to create a model-driven adversarial question writing procedure and generate a small set of challenge questions that QA-models fail on. Relatedly, Lan et al. (2017) propose a method for continuously growing a dataset of paraphrases.

There has been a flurry of work in constructing datasets with an adversarial component, such as Swag (Zellers et al., 2018) and HellaSwag (Zellers et al., 2019), CODAH (Chen et al., 2019), Adversarial SQuAD (Jia and Liang, 2017), Lambada (Paperno et al., 2016) and others. Our dataset is not to be confused with abductive NLI (Bhagavatula et al., 2019), which calls itself αNLI, or ART.

5.7 Discussion & Conclusion

In this work, we used a human-and-model-in-the-loop training method to collect a new benchmark for natural language understanding. The benchmark is designed to be challenging to current state-of-the-art models. Annotators were employed to act as adversaries, and encouraged to find vulnerabilities that fool the model into misclassifying, but that another person would correctly classify. We found that non-expert annotators, in this gamified setting and with appropriate incentives, are remarkably creative at finding and exploiting weaknesses. We collected three rounds, and as the rounds progressed, the models became more robust and the test sets for each round became more difficult. Training on this new data yielded the state of the art on existing NLI benchmarks.

The ANLI benchmark presents a new challenge to the community. It was carefully constructed to mitigate issues with previous datasets, and was designed from first principles to last longer. The dataset also presents many opportunities for further study. For instance, we collected annotator-provided explanations for each example that the model got wrong. We provided inference labels for the development set, opening up possibilities for interesting more fine-grained studies of NLI model performance. While we verified the development and test examples, we did not verify the correctness of each training example, which means there is probably some room for improvement there.

A concern might be that the static approach is probably cheaper, since dynamic adversarial data collection requires a verification step to ensure examples are correct. However, verifying examples is probably also a good idea in the static case, and adversarially collected examples can still prove useful even if they didn't fool the model and weren't verified. Moreover, annotators were better incentivized to do a good job in the adversarial setting. Our finding that adversarial data is more data-efficient corroborates this theory. Future work could explore a detailed cost and time trade-off between adversarial and static collection.

It is important to note that our approach is model-agnostic. HAMLET was applied against an ensemble of models in rounds 2 and 3, and it would be straightforward to put more diverse ensembles in the loop to examine what happens when annotators are confronted with a wider variety of architectures.

The proposed procedure can be extended to other classification tasks, as well as to ranking with hard negatives either generated (by adversarial models) or retrieved and verified by humans. It is less clear how the method can be applied in generative cases.

Adversarial NLI is meant to be a challenge for measuring NLU progress, even for as yet undiscovered models and architectures. Luckily, if the benchmark does turn out to saturate quickly, we will always be able to collect a new round.

CHAPTER 6: SUMMARY, LIMITATIONS, AND FUTURE WORK

6.1 Summary of Contributions

We have presented our recent work centered around data-driven natural language inference. The main contributions can be described twofold: (1) understanding how we can train data-driven NLI models for downstream NLP applications; (2) how we can improve and evaluate NLI models more generally in a dynamic human-and-model-in-the-loop fashion.

In Chapter 2, we describe our efforts in developing sentence encoder-based NLI models that achieved state-of-the-art performance. The resultant sentence vectors are shown to be more discriminative about certainty topical attributes than other existing sentence encoders. In Chapter 3, we developed a retriever and reader framework that combines NLI and VQ with an information retriever system and achieve state-of-the-art performance in both fact verification and open-domain question answering tasks. In Chapter 4, we initiate a research effort on utilizing NLI models for contradiction detection in dialogue and consistent generation. In Chapter 5, we propose a dynamic and adversarial human and model in the loop evaluation and training framework for natural language inference. We first train state-of-the-art NLI models using the best NLP model which is the BERT and then we deploy the models and ask the annotator to play with the models and write difficult examples that fool the models. Once we have those difficult examples, we make train/dev/test split and re-train the model using the data and do it again. The process can be seen as a competitive iteration between model training and cloud sourcing and in the end, we can have both more difficult evaluation sets and more robust models. The result shows that it not only got state-of-the-art performance on existing NLI datasets but turn out to be more robust in NLI stress test.

All the presented datasets and algorithms in this thesis are publicly available to the community and have attracted substantial attention. We hope our work could provide continued support and inspiration for the research community in this area.

6.2 Limitations and Future Work

While progress has been achieved in natural language inference as we detailed above, the task of natural language inference is still far from achieved. First of all, the definition of the task imposes certain ambiguity on the category boundaries between the three NLI labels. For a noticeable amount of examples, there are significant human disagreements about the ground truth label. How to refine the definition or the utility of NLI is still an open question. Secondly, NLI has been used as an important benchmark task for natural language understanding while existing datasets are saturated quickly. Adversarial NLI is the first proposal to solve such issues. However, there are also existing problems with ANLI such as difficulties of annotators writing good adversarial examples and potentially low inter-annotation agreements in adversaries. Moreover, adversarial training can only be an antidote for improvement on NLU and we will eventually hit the limit of model capacity. Other methods like invariant risk minimization or restricting the modeling to linguist grounded structures should be further explored to advance general natural language understanding.

In the current world of NLU learning, training large scale language models on static corpus or crawled web text data is able to yield a decently well NLP foundation models and the community has achieved significant progress. However, now we might have reached a bottleneck with this learning paradigm. With the increasing user-case in human-machine interaction, it starts to become easier for us to have access to tons of human-and-bot interaction data and we are getting closer to the idea of learning language through interaction where the model will have a better alignment with human intention. One crucial future research topic is how we can take the advantages of this interaction interfaces or data and train models with more direct access to human feedbacks. We can envision some scenarios in which we deploy some pre-trained models and then sparse but necessary human feedback will be needed to further improve the models. Research on NLI and ANLI can be a pilot case study where such an interactive NLU endeavor can be tested.

REFERENCES

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al. (2020). Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., and Parikh, D. (2015). Vqa: Visual question answering. In *ICCV*.

Bang, J., Noh, H., Kim, Y., and Lee, G. G. (2015). Example-based chat-oriented dialogue system with personalized long-term memory. In *2015 International Conference on Big Data and Smart Computing (BigComp)*, pages 238–243. IEEE.

Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., and Blackburn, J. (2020). The pushshift reddit dataset. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 830–839.

Belinkov, Y. and Bisk, Y. (2017). Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Bentivogli, L., Dagan, I., Dang, H. T., Giampiccolo, D., and Magnini, B. (2009). The Fifth PASCAL Recognizing Textual Entailment Challenge. *TAC*.

Bhagavatula, C., Bras, R. L., Malaviya, C., Sakaguchi, K., Holtzman, A., Rashkin, H., Downey, D., Yih, S. W.-t., and Choi, Y. (2019). Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015a). A large annotated corpus for learning natural language inference. In *EMNLP*.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015b). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Bowman, S. R., Gupta, R., Gauthier, J., Manning, C. D., Rastogi, A., and Potts, C. (2016). A fast unified model for parsing and sentence understanding. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, pages 1466–1477. Association for Computational Linguistics (ACL).

Bras, R. L., Swayamdipta, S., Bhagavatula, C., Zellers, R., Peters, M. E., Sabharwal, A., and Choi, Y. (2020). Adversarial filters of dataset biases. *arXiv preprint arXiv:2002.04108*.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020a). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020b). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017a). Reading wikipedia to answer open-domain questions. In *ACL*.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017b). Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.

Chen, M., D'Arcy, M., Liu, A., Fernandez, J., and Downey, D. (2019). CODAH: an adversarially authored question-answer dataset for common sense. *CoRR*, abs/1904.04365.

Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., and Inkpen, D. (2017c). Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Cireşan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *arXiv preprint arXiv:1202.2745*.

Clark, C. and Gardner, M. (2017). Simple and effective multi-paragraph reading comprehension. *Association for Computational Linguistics (ACL)*.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2019). Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Conneau, A. and Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017a). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680. Association for Computational Linguistics.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017b). Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

Dagan, I., Glickman, O., and Magnini, B. (2005). The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Dehghani, M., Zamani, H., Severyn, A., Kamps, J., and Croft, W. B. (2017). Neural ranking models with weak supervision. In *SIGIR*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019a). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019b). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Dinan, E., Humeau, S., Chintagunta, B., and Weston, J. (2019). Build it Break it Fix it for Dialogue Safety: Robustness from Adversarial Human Attack. In *Proceedings of EMNLP*.

Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I., Lowe, R., et al. (2020). The second conversational intelligence challenge (convai2). In *The NeurIPS'18 Competition*, pages 187–208. Springer.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., and Weston, J. (2018). Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Ding, M., Zhou, C., Chen, Q., Yang, H., and Tang, J. (2019). Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703.

Dziri, N., Kamalloo, E., Mathewson, K., and Zaiane, O. (2019a). Evaluating coherence in dialogue systems using entailment. 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)*, pages 3806–3812, Minneapolis, Minnesota. Association for Computational Linguistics.

Dziri, N., Kamalloo, E., Mathewson, K., and Zaiane, O. R. (2019b). Augmenting neural response generation with context-aware topical attention. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 18–31.

Ettinger, A., Rao, S., Daumé III, H., and Bender, E. M. (2017). Towards linguistically generalizable nlp systems: A workshop and shared task. *arXiv preprint arXiv:1711.01505*.

Fan, A., Lewis, M., and Dauphin, Y. (2018). Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Gardner, M., Artzi, Y., Basmov, V., Berant, J., Bogin, B., Chen, S., Dasigi, P., Dua, D., Elazar, Y., Gottumukkala, A., Gupta, N., Hajishirzi, H., Ilharco, G., Khashabi, D., Lin, K., Liu, J., Liu, N. F., Mulcaire, P., Ning, Q., Singh, S., Smith, N. A., Subramanian, S., Tsarfaty, R., Wallace, E., Zhang, A., and Zhou, B. (2020). Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Geiger, A., Cases, I., Karttunen, L., and Potts, C. (2018). Stress-testing neural models of natural language inference with multiply-quantified sentences. *arXiv preprint arXiv:1810.13033*.

Geva, M., Goldberg, Y., and Berant, J. (2019). Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. *arXiv preprint arXiv:1908.07898*.

Glockner, M., Shwartz, V., and Goldberg, Y. (2018). Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of ACL*.

Gong, Y., Luo, H., and Zhang, J. (2018). Natural language inference over interaction space. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (2017). Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913.

Grice, H. P. (1975). Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016). A deep relevance matching model for ad-hoc retrieval. In *CIKM*.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. (2018a). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., and Smith, N. A. (2018b). Annotation artifacts in natural language inference data. In *NAACL*.

Hanselowski, A., Zhang, H., Li, Z., Sorokin, D., Schiller, B., Schulz, C., and Gurevych, I. (2018). Ukp-athene: Multi-sentence textual entailment for claim verification. In *The 1st Workshop on Fact Extraction and Verification*.

Hashimoto, K., Xiong, C., Tsuruoka, Y., and Socher, R. (2017). A joint many-task model: Growing a neural network for multiple nlp tasks. In *EMNLP*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *CVPR*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hill, F., Bordes, A., Chopra, S., and Weston, J. (2015). The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*.

Htut, P. M., Bowman, S. R., and Cho, K. (2018). Training a ranking function for open-domain question answering. In *NAACL-HLT*.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *CIKM*.

Humeau, S., Shuster, K., Lachaux, M.-A., and Weston, J. (2019). Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on International Conference on Machine Learning (ICML)*.

Jabri, A., Joulin, A., and Van Der Maaten, L. (2016). Revisiting visual question answering baselines. In *European conference on computer vision*, pages 727–739. Springer.

Jia, R. and Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In *Proceedings of EMNLP*.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Kaushik, D., Hovy, E., and Lipton, Z. C. (2019). Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

Kaushik, D. and Lipton, Z. C. (2018). How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR*.

Kratzwald, B. and Feuerriegel, S. (2019). Learning from on-line user feedback in neural question answering on the web. In *The World Wide Web Conference*, pages 906–916. ACM.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Lan, W., Qiu, S., He, H., and Xu, W. (2017). A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the Institute of Electrical and Electronics Engineers*, 86(11):2278–2324.

Lee, J., Yun, S., Kim, H., Ko, M., and Kang, J. (2018). Ranking paragraphs for improving answer recall in open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., and Dolan, B. (2016). A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany. Association for Computational Linguistics.

Li, M., Roller, S., Kulikov, I., Welleck, S., Boureau, Y.-L., Cho, K., and Weston, J. (2020). Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, Online. Association for Computational Linguistics.

Li, M., Weston, J., and Roller, S. (2019). Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087*.

Ling, H. and Fidler, S. (2017). Teaching machines to describe images via natural language feedback. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5075–5085.

Liu, X., He, P., Chen, W., and Gao, J. (2019a). Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.

MacCartney, B. and Manning, C. D. (2009). *Natural language inference*. Citeseer.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

McCoy, R. T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL*.

Mikolov, T., Karafiát, M., Burget, L., et al. (2010). Recurrent neural network based language model. In *In INTERSPEECH 2010,*. Citeseer.

Miller, A. H., Feng, W., Fisch, A., Lu, J., Batra, D., Bordes, A., Parikh, D., and Weston, J. (2017). Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., et al. (2018). Never-ending learning. *Communications of the ACM*, 61(5):103–115.

Mitra, B., Diaz, F., and Craswell, N. (2017). Learning to match using local and distributed representations of text for web search. In *WWW*.

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., and Allen, J. (2016). A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*.

Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., and Jin, Z. (2016). Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136.

Munkhdalai, T. and Yu, H. (2017). Neural semantic encoders. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 1, page 397. NIH Public Access.

Naik, A., Ravichander, A., Sadeh, N., Rose, C., and Neubig, G. (2018). Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nangia, N., Williams, A., Lazaridou, A., and Bowman, S. (2017). The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 1–10.

Nass, C. and Moon, Y. (2000). Machines and mindlessness: Social responses to computers. *Journal of social issues*, 56(1):81–103.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. (2016). Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Nie, Y. and Bansal, M. (2017). Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 41–45.

Nie, Y., Chen, H., and Bansal, M. (2019a). Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Nie, Y., Wang, Y., and Bansal, M. (2019b). Analyzing compositionality-sensitivity of nli models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6867–6874.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2020a). Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2020b). Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Nie, Y., Williamson, M., Bansal, M., Kiela, D., and Weston, J. (2021). I like fish, especially dolphins: Addressing contradictions in dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. (2016). The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Pasunuru, R. and Bansal, M. (2017). Multi-task video captioning with video and entailment generation. In *ACL*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *NAACL-HLT*.

Poliak, A., Belinkov, Y., Glass, J., and Van Durme, B. (2018a). On the evaluation of semantic phenomena in neural machine translation using natural language inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 513–523.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018b). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018c). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191. Association for Computational Linguistics.

Qian, Q., Huang, M., Zhao, H., Xu, J., and Zhu, X. (2018). Assigning personality/profile to a chatting machine for coherent conversation generation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4279–4285.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Rashkin, H., Smith, E. M., Li, M., and Boureau, Y.-L. (2019). Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2018). Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. 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)*, pages 3982–3992.

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. (2020). Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Shuster, K., Smith, E. M., et al. (2020). Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Ruef, A., Hicks, M., Parker, J., Levin, D., Mazurek, M. L., and Mardziel, P. (2016). Build it, break it, fix it: Contesting secure development. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 690–703. ACM.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Sankar, C., Subramanian, S., Pal, C., Chandar, S., and Bengio, Y. (2019). Do neural dialog systems use the conversation history effectively? an empirical study. *arXiv preprint arXiv:1906.01603*.

See, A., Roller, S., Kiela, D., and Weston, J. (2019). What makes a good conversation? how controllable attributes affect human judgments. 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)*, pages 1702–1723, Minneapolis, Minnesota. Association for Computational Linguistics.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A. C., and Bengio, Y. (2017). A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*.

Shakespeare, W. (1603). *The Tragedy of Hamlet, Prince of Denmark.*

Smith, E. M., Williamson, M., Shuster, K., Weston, J., and Boureau, Y.-L. (2020). Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030, Online. Association for Computational Linguistics.

Thorne, J. and Vlachos, A. (2018). Automated fact checking: Task formulations, methods and future directions. In *International Conference on Computational Linguistics (COLIN)*.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018a). FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018b). Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

Tsuchiya, M. (2018). Performance impact caused by hidden bias of training data for recognizing textual entailment. In *LREC*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D., and Batra, D. (2016). Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Wallace, E., Rodriguez, P., Feng, S., Yamada, I., and Boyd-Graber, J. (2019). Trick me if you can: Human-in-the-loop generation of adversarial question answering examples. In *Transactions of the Association for Computational Linguistics*.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018a). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Wang, D., Jojic, N., Brockett, C., and Nyberg, E. (2017a). Steering output style and topic in neural response generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2140–2150, Copenhagen, Denmark. Association for Computational Linguistics.

Wang, S., Yu, M., Guo, X., Wang, Z., Klinger, T., Zhang, W., Chang, S., Tesauro, G., Zhou, B., and Jiang, J. (2018b). R: Reinforced ranker-reader for open-domain question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wang, Z., Hamza, W., and Florian, R. (2017b). Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

Welbl, J., Stenetorp, P., and Riedel, S. (2018). Constructing datasets for multi-hop reading comprehension across documents. *TACL*.

Welleck, S., Weston, J., Szlam, A., and Cho, K. (2019). Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Weston, J., Dinan, E., and Miller, A. H. (2018). Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.

Wieting, J. and Kiela, D. (2019). No training required: Exploring random encoders for sentence classification. *arXiv preprint arXiv:1901.10444*.

Williams, A., Nangia, N., and Bowman, S. (2018a). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Williams, A., Nangia, N., and Bowman, S. (2018b). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Williams, A., Nangia, N., and Bowman, S. (2018c). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics.

Williams, A., Nangia, N., and Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*.

Worsnick, S. (2018). Mitsuku wins loebner prize 2018.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. (2018). Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Yang, Z., Zhang, S., Urbanek, J., Feng, W., Miller, A. H., Szlam, A., Kiela, D., and Weston, J. (2017). Mastering the dungeon: Grounded language learning by mechanical turker descent. *arXiv preprint arXiv:1711.07950*.

Yoneda, T., Mitchell, J., Welbl, J., Stenetorp, P., and Riedel, S. (2018). Ucl machine reading group: Four factor framework for fact finding (hexaf). In *The 1st Workshop on Fact Extraction and Verification*.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. (2018). Swag: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. (2019). Hellaswag: Can a machine really finish your sentence? In *ACL*.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Zhang, W.-N., Zhu, Q., Wang, Y., Zhao, Y., and Liu, T. (2019a). Neural personalized response generation as domain adaptation. *World Wide Web*, 22(4):1427–1446.

Zhang, Y., Chen, G., Yu, D., Yao, K., Khudanpur, S., and Glass, J. (2016). Highway long short-term memory rnns for distant speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5755–5759. IEEE.

Zhang, Y., Gao, X., Lee, S., Brockett, C., Galley, M., Gao, J., and Dolan, B. (2019b). Consistent dialogue generation with self-supervised feature learning. *arXiv preprint arXiv:1903.05759*.

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, B. (2020). DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Zhou, L., Gao, J., Li, D., and Shum, H.-Y. (2020). The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93.