longhorns at DADC 2022: How many linguists does it take to fool a Question Answering model? A systematic approach to adversarial attacks.

Venelin Kovatchev[1]†* Trina Chatterjee[3]* Venkata S Govindarajan[2]*
Jifan Chen[3] Eunsol Choi[3] Gabriella Chronis[2] Anubrata Das[1] Katrin Erk[2]
Matthew Lease[1] Junyi Jessy Li[2] Yating Wu[4] Kyle Mahowald[2]*
* Contributors towards the official submission
[1] School of Information, The University of Texas at Austin
[2] Department of Linguistics, The University of Texas at Austin
[3] Department of Computer Science, The University of Texas at Austin
[4] Department of Electrical and Computer Engineering, The University of Texas at Austin

Abstract

Developing methods to adversarially challenge NLP systems is a promising avenue for improving both model performance and interpretability. Here, we describe the approach of the team "longhorns" on Task 1 of the The First Workshop on Dynamic Adversarial Data Collection (DADC), which asked teams to manually fool a model on an Extractive Question Answering task. Our team finished first, with a model error rate of 62%.[1] We advocate for a systematic, linguistically informed approach to formulating adversarial questions, and we describe the results of our pilot experiments, as well as our official submission.

1 Introduction

Rapid progress in NLP has resulted in systems obtaining apparently super-human performance on popular benchmarks such as GLUE (Wang et al., 2018), SQUaD (Rajpurkar et al., 2016), and SNLI (Bowman et al., 2015). Dynabench (Kiela et al., 2021) proposes an alternative approach to benchmarking: a dynamic benchmark wherein a human adversary creates examples that can "fool" a state-of-the-art model but not a human language user. The idea is that, by generating and compiling examples that fool a particular system, the community can gain a better idea of that system's actual strengths and weaknesses, as well as ideas and data for iteratively improving it.

There is no straightforward recipe, however, for generating successful adversarial examples. To contribute to that knowledge base, this paper describes the strategy used by team "longhorns" in Task 1 of The First Workshop on Dynamic Adversarial Data Collection (DADC), which was on Extractive Question Answering (answering a question about a passage by pointing to a particular span of text within that passage).[2] We focus not only on describing the details of our strategy, but also on our process for approaching the task. At the time of this paper submission, pending expert validation of the results, our team ranked first in the competition, obtaining 62% Model Error Rate (MER).

Our approach towards creating adversarial examples was designed to be systematic, analytical, and draw on linguistically informed ideas. We first compiled a list of linguistically inspired "attack strategies" and used it to create adversarial examples in a systematic manner. We then analyzed some existing biases of the model-in-the-loop and its performance on a variety of different attacks. We used this piloting phase to select the best performing attacks for the official submission.

Based on the approaches that were most successful both in our pilot studies and in our official submission, we posit that the following broad areas should be of particular interest for theoretically motivated adversarial attacks on contemporary NLP systems, as evidenced by their strong performance on our target task:

- **Taking advantage of models' strong priors.** The model was proficient at identifying the correct kind of named entity being asked for (e.g., a person for a "who" question, a place for a "where" question), but was biased to give answers which were salient (either topically or because they appeared first (Ko et al., 2020)) or which had high lexical overlap with the question. Thus, picking a distractor with the same entity type as the target answer (e.g., another person mentioned in the text when

†Primary author and coordinator (venelin@utexas.edu)
[1]The results and the team ranking are pending validation from the organizers of the task at the time of the submission.
[2]https://dadcworkshop.github.io/shared-task/

Proceedings of the First Workshop on Dynamic Adversarial Data Collection, pages 41 - 52

the question was a "who" question) was often effective. This result is broadly consistent with observations that modern NLP systems can perform well in the general case but can be biased towards frequency-based priors (e.g., Wei et al., 2021) that mean they are sometimes "right for the wrong reasons" (McCoy et al., 2019).

- **Using language that is linguistically taxing for humans (and machines) to process.** Psycholinguists who study human language processing often study constructions that are grammatical but difficult for humans to process in real time, such as garden path sentences (Frazier and Rayner, 1982; Ferreira and Henderson, 1991) and complex coreference resolution (Kaiser and Fedele, 2019; Durrett and Klein, 2013). We found that the model was indeed often fooled by questions that included these types of constructions. While we did not collect any human data, the sentences that fooled the model are likely to be hard for humans as measured by tests of real-time processing difficulty (e.g., eye tracking, self-paced reading), even though humans would be able to successfully process these sentences given enough time.

- **Tapping into domain-general, non-linguistic reasoning.** We found that asking questions which do not require mere linguistic processing but require other kinds of reasoning (e.g., numerical reasoning, temporal reasoning, common-sense reasoning, list manipulation) were hard for the model. This result is consistent with prior work showing that language models struggle with these kinds of reasoning tasks (Marcus, 2020; Elazar et al., 2021; Talmor et al., 2020) and may be more generally explained by evidence from cognitive science that these kinds of reasoning tap into cognitive processes that are distinct from linguistic processing (Diachek et al., 2020; Blank et al., 2014).

Because these strategies and this general approach are broad and theoretically motivated, we believe that our methods could be used to generate adversarial examples on other Natural Language Understanding tasks besides Question Answering. In what follows, we characterize our approach in both the pilot phase and official submission, provide our list of attack strategies, and discuss the limitations of the task and model.

2 Task Definition

In Task 1 of DADC, titled "Better Annotators", each participating team submits 100 "official" extractive question answering (QA) examples through the Dynabench platform. The organizers of the shared task provide short passages as context and the participants have to create questions that "can be correctly answered from a span in the passage and DO NOT require a Yes or No answer". The objective is to find as many model-fooling examples as possible – the winning team is the one with the highest validated model error rate (vMER)[3].

The competition uses Dynabench (Kiela et al., 2021): "an open-source platform for dynamic dataset creation and model benchmarking". Dynabench aims to facilitate human-and-model-in-the-loop dataset creation. The annotators' aim to generate examples that will be misclassified by an automated model, but can be answered correctly by competent human speakers. Dynabench has been used to create data for Question Answering (Kaushik et al., 2021), Natural Language Inference (Williams et al., 2022), Online Hate Detection (Vidgen et al., 2021), and Sentiment Analysis (Potts et al., 2021), among others.

3 Approach

Our team consisted of faculty, postdocs, and students from the UT Austin linguistics department, computer science department, information school, and electrical and computer engineering department.

We approached the problem of creating adversarial attacks in a systematic manner, informed by ideas from computational linguistics, psycholinguistics, and theoretical linguistics. We composed a list of linguistic phenomena and reasoning capabilities that we hypothesized would be difficult for a state-of-the-art QA model. We then used some of those phenomena to create our official submission of adversarial examples. While the list is not exhaustive, it covers a wide range of potential attack strategies and can be used to guide the creation of adversarial examples for other tasks and systems.

[3]For full instructions, see https://dadcworkshop.github.io/shared-task/

Separate from our official submission for the competition, we ran a series of pilot experiments in which we used the list as a guide for experimenting with a variety of strategies for creating adversarial example. To ensure a fair and competitive official submission, all pilot experiments were carried out either before the official start of the shared task or after the official submission was made.

Our objective when evaluating the different adversarial strategies was to explore the space of potential attack strategies to determine the most successful ones for fooling the model. For each strategy, we measured the Model Error Rate (MER) on a small sample of example, and we also analyzed how frequently the attack can be used.

Based on the results of these pilot experiments, we targeted the best strategies for our official submission. Official question submissions were made by subsets of the team, in group sizes ranging from 1 to around 10. Since not all strategies can be used for all example passages, we used the specific passages we were presented with in order to guide our decision about what strategy to focus on for a particular question. When more than one participant was present, question submissions were made by consensus agreement among those present.

Anecdotally, we found that the attacks were often more successful when multiple team members are present, with each member hypothesizing model behaviors from diverse angles. Overall, we found the adversarial question generation process nontrivial, taking 5-10 minutes per passage, although we became faster over time. We also chose to skip passages occasionally when the passage covers very well-known entity, is too simple, or is not promising to most of our strategies (not having distractor entities, etc). We generated multiple questions for promising passages.

4 Pilot Experiments: Evaluating Adversarial Strategies

Many of our "adversarial strategies" are inspired by prior work in adversarial data generation and unit testing for Question Answering, Natural Language Inference, and Paraphrase Identification (Glockner et al., 2018; Kovatchev et al., 2018; Naik et al., 2018; Dua et al., 2019; Kovatchev et al., 2019; Nie et al., 2019; Wallace et al., 2019; Bartolo et al., 2020; Gardner et al., 2020; Hossain et al., 2020; Jeretic et al., 2020; Kaushik et al., 2020; Ribeiro et al., 2020; Saha et al., 2020). We propose the following linguistic and reasoning phenomena as a source for potential adversarial attacks:

Lexical knowledge Examples that require understanding lexical properties and in partiuclar lexical entailments that require knowledge of **hypernyms/hyponyms** (e.g., knowing that dog => animal, but animal =/> dog), **named entities** and their properties (e.g., knowing Shakira is a singer), **nominalization** (e.g., knowing "a submission" implies something has been submitted), **(a)symmetrical relations** (e.g., knowing that John marrying Mary implies Mary marrying John, but John loving Mary does not imply Mary loving John), **polarity substitutions** (e.g., knowing that a movie is good implies that it is not bad), **converse substitution** (e.g., knowing that if something has been provided, it has been received), **comparisons with antonyms** (e.g., knowing that if Clara is the tallest, she is not shorter than Mary), **reasoning about modal verbs** (e.g., understanding that if something *could* happen, that does not mean it *did* happen), and **reasoning about quantifers** (e.g., knowing that if some swans are white, that does not imply all swans are white).

Syntax and Discourse knowledge Examples that require syntactic or discourse-level understanding such as **Genitives** (e.g., knowing that elephant's foot = the foot of the elephant) and **Datives** (e.g., knowing that give her a cake = give a cake to her). **Relative Clauses** can be used in attacks to either include distracting information (e.g., "Maria, who is the president of the company" when the correct answer has nothing to do with Maria's role in the company) or to specify the correct referent (e.g., "the actor who bought the house" when that actor must be distinguished from a set of other actors).

When **Conjunction** or **Disjunction** appear in the passage (e.g., John and Mary love strawberries and cake, but John doesn't like chocolate), an adversarial question targets the ability of the model to correctly identify the syntactic scope (e.g.: Who loves cake and chocolate?). Closely related are the phenomena of **Intersectivity** (e.g., knowing that "a singer and a good man" =/> a good singer) and **Restrictivity** (e.g., understanding that "all my work due today" =/> all my work).

When a complex **prepositional phrase attachment** appears in the passage (e.g., "I saw two men with a telescope in the park"), an adversarial question requires disambiguation (e.g., "Who has the telescope"). Questions based on the **Argumenta-**

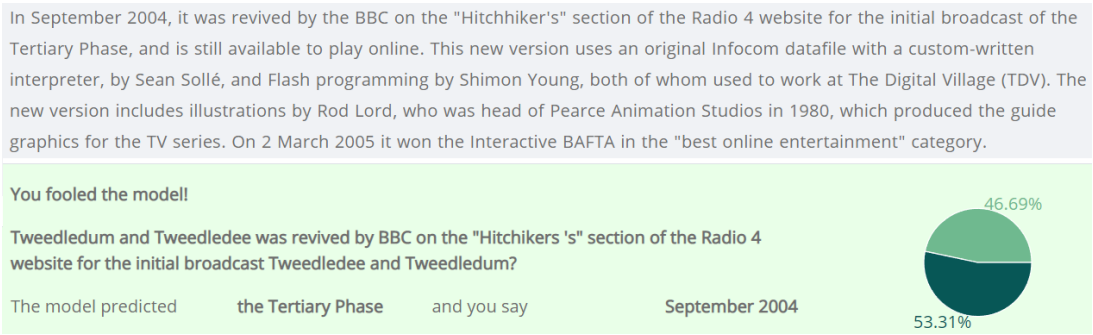

Figure 1: An example of "semantic similarity" model bias: when the model is fed a nonsensical question, it responds with an answer with high semantic overlap with the question.

tive structure require the model to correctly identify the core arguments (e.g., knowing that "John broke the vase" implies that "The vase broke"; but does not imply "John broke."). This attack can be further complicated when the same verb appears multiple times in the passage. Adversarial attacks based on **Ellipsis**, **Anaphora**, and **Coreference** test the ability of the model to process long distance syntactic dependencies.

Negation can appear both in the passage and in the question. It can be expressed in a variety of ways: simple (e.g., no, not), adverbial (e.g., never), pronoun (e.g., nobody), morphological (e.g., unfinished), lexical (e.g., refuse to), implicit (e.g., I wish I had a boat), double negation. Adversarial questions can also target the ability of the model to identify the scope of negation either in the question or in the passage.

Garden Path questions (e.g., Who is the director of the movie directing?) are syntactically confusing and much-studied in psycholinguistics for causing processing difficulty in humans (Frazier, 1979).

Questions about **Mental States** of individuals are inspired by work in psychology showing that it can be challenging to reason about the mental states of others (e.g., knowing that "Why does Maria think that Sandra is leaving?" could require a different answer than "Why is Sandra leaving?"(Wellman, 1992; Kovatchev et al., 2020).

Reasoning Questions that require various kinds of non-linguistic reasoning such as **Conditionals and hypothetical situations** (e.g., Who would be the champion if Mary didn't lose the final?), **Numerical Reasoning** (e.g., Who is the second richest person?), **Temporal Reasoning** (e.g., What happened in a specific timeframe?), **Commonsense reasoning** (including logical implications, contradiction, etc.), and **List manipulations** (e.g., Which two of the actors in the list are male?).

Finally, **distractor**-based attacks make use of model priors by expanding the question with additional information. **Meaningful distractors** directs the model towards a wrong answer, while the strategy of **adding noise** relies on increasing the complexity of the question.

The different phenomena can appear in the passages, in the question, or in both. Not every phenomena can be used to generate attacks for every passage, and the phenomena also appear with different frequency in the data. In our pilot experiments we distributed the different phenomena across the members of the team. We measured the Model Error Rate (MER) for the different strategies and determined how frequently each attack could be used.

5 Exploring Model Biases

During the pilot experiment step in Section 4 we found that the model-in-the-loop performs surprisingly well on a variety of different attacks. We hypothesized that at least in some situations, the strong performance is due to spurious correlations, the nature of the underlying language model, and the nature of the task. We further carried out a set of experiments to determine some specifics of the model behavior. We briefly discuss two "shortcuts" used by the model.

Semantic similarity Figure 1 illustrates the model bias towards "semantic similarity" on a nonsensical question. When the model is unsure what to do, or like in this example, when the question is not a valid English sentence, it identifies parts of the context that are similar to the question and predicts neighboring words. Due to the relatively short length of most of the passages, this strategy

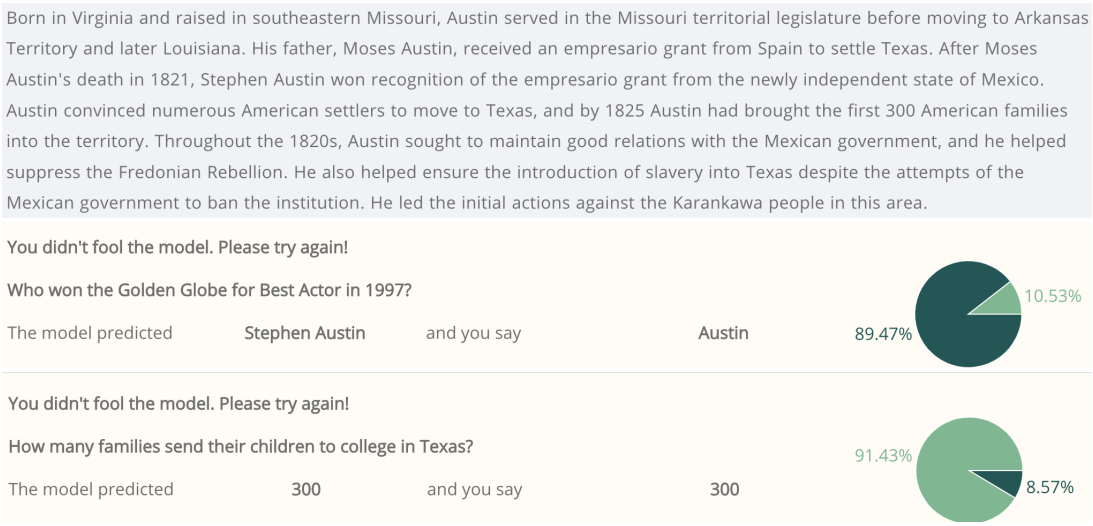

Figure 2: When faced with nonsensical questions, the model will give salient answers of the correct entity type. The passage is about Texas colonizer Stephen Austin, but the first question is about the Golden Globe awards. The model confidently answers that Stephen Austin won the Gold Globe for Best Actor in 1997.

often gets the correct answer without actually understanding the question.

Type of question We noticed that the model is very good at identifying some properties of the answer based on the type of question. For example, a "who" question typically asks for a named entity, while a "how many" question asks for a quantity. A strong heuristic adopted by the model is to return an answer of the correct "type" regardless of the actual question. Figure 2 illustrates that: neither question can be answered from the passage, but the model makes a guess based on the type of question. Once again, the short length of the passages and the fact that they typically contain just a few tokens of the correct "type" artificially boosts the performance of the model.

In our official submission, we used those model biases to increase the difficulty of the adversarial examples. When possible, we used those biases to guide the model towards a wrong answer. In passages where we could not confuse the model (e.g.: only one named entity in a "who" question), we rephrased the questions in such a way that makes it harder for the model to use heuristics.

6 Official Submission

After analyzing and discussing the results of our preliminary experiments, for our official submission we focus on the following strategies: using distractors, numerical reasoning, temporal reasoning, garden path questions, complex coreference, list manipulations, and common-sense reasoning. We also used the model biases to either confuse the model or reduce it's ability to rely on heuristics. In the rest of this section we briefly describe each of our strategies and provide examples.

6.1 Taking advantage of model priors

Distractors One of the most successful and easy-to-use adversarial strategies was using distractors. An example of that strategy can be seen in Figure 3: the phrasing of question has a high degree of similarity with the portion of the text talking about narrow belts ("between X and Y AU and relatively sharp boundaries"), however the correct response is "wide belts" due to the specified sizes. The "distractor" strategy can be used frequently. We often combined the distractor strategies with other strategies. For example, in Figure 3, we combine it with "numerical reasoning". Anecdotally, we found the distractors to be most successful when the correct answer was not the most salient entity of its type in the passage (e.g., targeting a briefly mentioned director in a passage mostly about one particular actor) and when there were many other entities of the desired type available, as opposed to just 1 or 2 (e.g., a "who" question for a passage that mentions 10 people is more challenging than a "who" question for a passage that mentions only 1 person).

6.2 Linguistically difficult utterances

Garden Path Questions Figure 4 shows an example of a garden path question (Frazier, 1979).

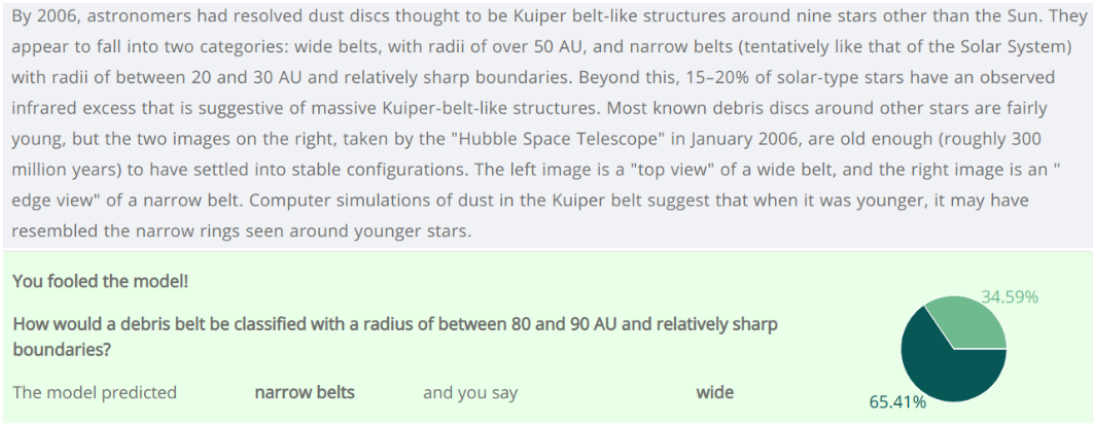

Figure 3: An example of "distractor" and "numerical reasoning" strategies. The model has to reason that "between 80 and 90" is greater than the 50 AU boundary identified in the passage.

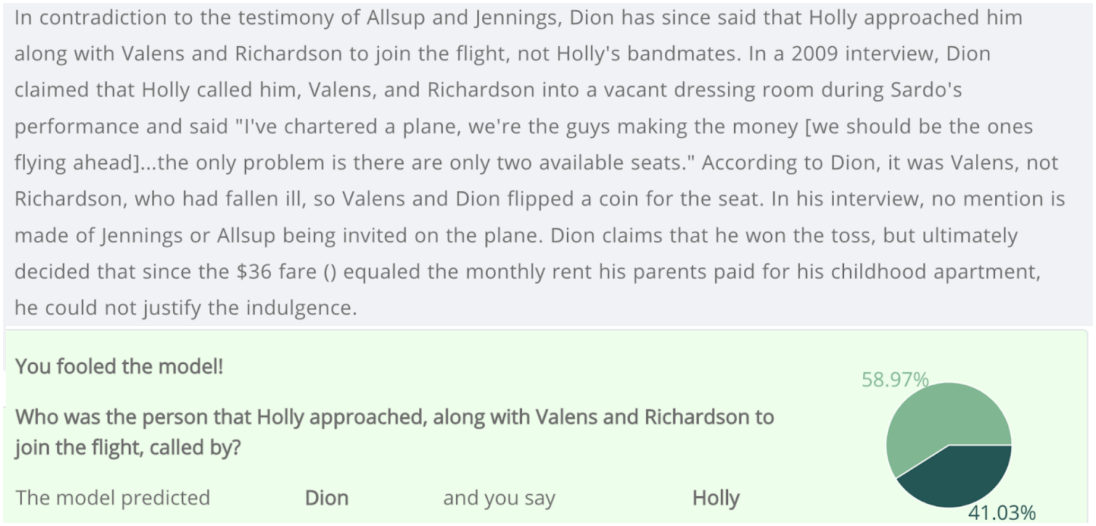

Figure 4: "Garden path" strategy. Until the very end of the sentence, the question seems to be about something else.

Until the last word, the question appears to be asking about "the person that Holly approached, along with Valens and Richardson" (to which the answer would be Dion). But the last word makes it clear that the reader needs to reparse the question, to see that it is actually asking who *calls* that person (i.e., Dion) – which makes the answer Holly. The model is unable to correctly process the complex syntactic structure of the sentence and responds "Dion". Garden path questions are easier to generate than temporal and numerical reasoning questions since they can be generated for a wider variety of texts, but we found that the model can often handle even quite complex syntactic constructions. We hypothesize that this is mainly due to the length of the passages and the "type of question" model bias.

Anaphora and Coreference Adversarial examples based on anaphora and coreference can require the model to demonstrate the ability to resolve long distance syntactic dependencies and often require making common-sense inferences as well. In Figure 5, the founders were worried about their own death. To correctly respond to the question, the model first has to identify "their" as the answer and then resolve the coreference between "their" and "the founders". Instead, the model just returns a salient named entity. Examples based on anaphora and coreference are relatively infrequent, as they require multiple entities and potentially ambiguous coreference in the passage.

6.3 Non-linguistic reasoning

Numerical Reasoning Adversarial examples based on numerical reasoning require the model to carry out simple mathematical calculations or comparisons to identify the correct answer. For example, in Figure 3, the model had to calculate

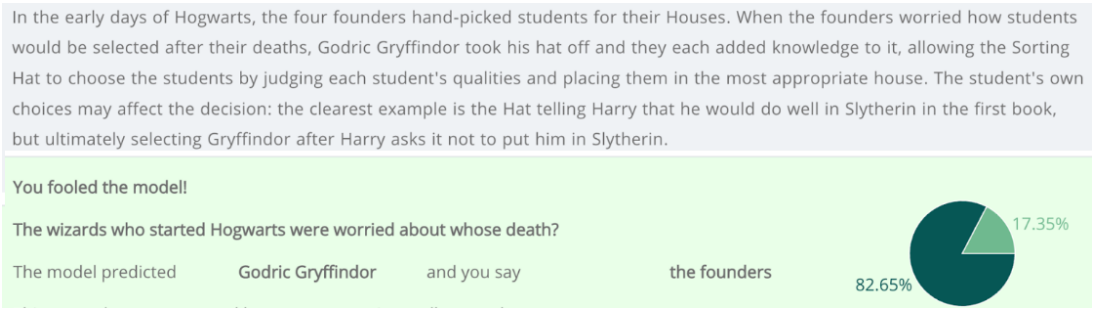

Figure 5: "Coreference" strategy. The model has to figure out that the word "their" refers to the founders.

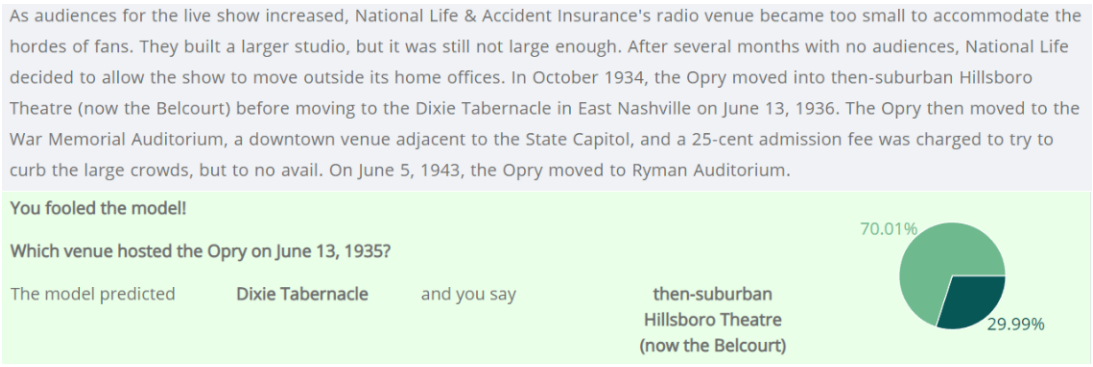

Figure 6: "Temporal reasoning" strategy. The model has to understand that June 13, 1935 is during the period when the Opry moved to the Hillsboro Theatre.

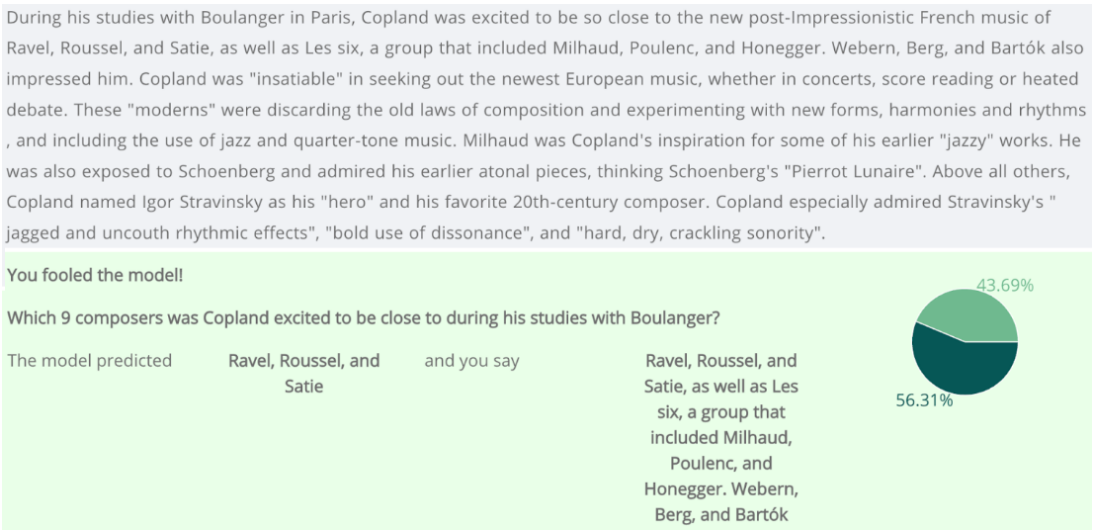

Figure 7: "List manipulations" strategy. The model has to identify the 9 composers asked for, but only gives 3.

that "between 80 and 90" is "over 50" in order to answer correctly.

Temporal Reasoning Adversarial examples based on temporal reasoning require the model to reason about the chronological order of events and the different states of the world at different points in time. In the example shown in Figure 6 the Opry moves to Hillsboro in 1934 and then to Dixie Tabernacle in 1936. We ask the model for a date that is not mentioned explicitly (June 13, 1935). The correct answer is "Hillsboro", but the model is fooled by recognizing a portion of the date (June 13) and predicts Dixie Tabernacle.

Temporal-based examples are relatively rare, as they require the passage to have multiple dates as well as multiple different events and world states associated with the dates. However, when available, temporal-based attacks were very successful.

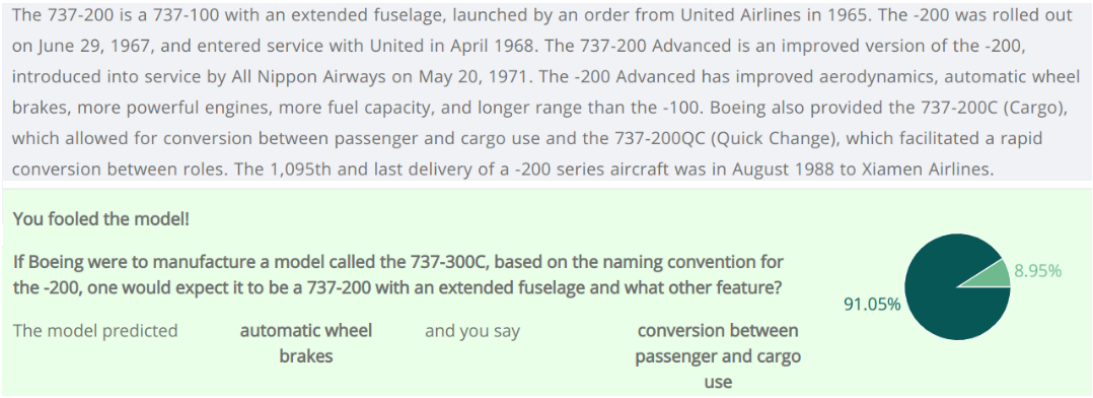

Figure 8: "Common-sense reasoning" strategy. The model has to flexibly adapt the naming convention to a hypothetical example: a kind of creative reasoning task that humans do easily but that models often struggle with.

List Manipulations We used two different strategies to create adversarial examples based on lists. Figure 7 illustrates one of them. The question requests the full list of 9 composers, while the model only extracts the first three due to the syntactic structure of the sentence. The second list-based strategy asks for a subset of a list that fulfills certain criteria. List-based adversarial attacks are relatively infrequent in a single passage setting we study.

Common-sense Reasoning Adversarial attacks based on common sense reasoning test the basic understanding of the world of the model or its ability to reason about different entities and objects. In Figure 8, the model is required to break apart the string "737-300C" and "737-200C" correctly and then reason about the naming convention: we are told that "C" stands for cargo and so the hypothetical "737-300C" should also have the cargo feature.

7 Discussion

A fundamental feature of language is that it is a cooperative enterprise (Clark, 1996) that enables efficient communication between parties (Gibson et al., 2019). Therefore, in ordinary language, people typically talk about discourse-relevant entities (Sperber and Wilson, 1986), avoid difficult syntactic constructions (Futrell et al., 2015; Gibson, 1998), and structure information in a way that is easy to produce and understand (MacDonald, 2013; Levy, 2008). If anything unifies all of our most successful attack strategies, it is that they eschew these principles in the context of the given task and passages. Instead, successful attacks ask about surprising aspects of the text (e.g., by including distractors), often using complex language (e.g., garden path sentences and complex coreference resolution) and reasoning (e.g., temporal and numeric reasoning).

So, in some ways, the successful attack questions are less likely to be encountered in ordinary language use (leading to claims that adversarial examples are brittle, e.g., Phang et al., 2021; Bowman, 2022). But another key property of human language is that it is flexible and generative, such that people can produce and understand surprising and unexpected utterances. To that end, we think these adversarial questions are a fair target for improving systems precisely because they are linguistically unusual: human language is not just for the "average case" but can be used to express meanings that are subtle, interesting, and complicated.

Perhaps because these questions also require humans to think creatively outside their ordinary linguistic experience, we also found that we achieved better performance when we had larger groups of people working on generating questions at once, so that there was a wider diversity of ideas.

Indeed, while some questions may be less likely to appear in a "extractive question answering" dataset, they are understandable by humans and are likely to be useful for efficient communication in real-world settings. The objective behind "extractive QA" is that a machine should answer any question that a human would, given the passage. A variety of real-world tasks can be reduced to extractive QA and in many cases the "correct" passage corresponding to the question is not known a priori. Asking questions such as "Where was X at a time Y" and "What is the difference between 737-200 and 737-200C" may be less natural for a

human that has access to the passage, but are questions that someone would, for example, ask their automated assistant. Therefore, a well functioning model needs to embrace the creativity and be able to correctly answer adversarial questions.

Finally, the adversarial attacks that we present are not just interesting from the scientific point of view, but also have clear practical implications. Most of the attacks correspond to specific capacities of the model-in-the-loop such as coreference resolution, numerical and temporal reasoning. The consistently high MER indicates that the model underperforms in tasks that require those capacities.

Our approach towards creating adversarial examples allows us to implicitly evaluate the performance of the model and the quality of the data with respect to a wide variety of linguistic and reasoning categories. Overall, we found that the model-in-the-loop performs impressively good on the majority of question types. Only a small subset of the strategies could consistently obtain above 50% MER and these strategies did not necessarily work for all questions. For instance, questions with relatively few possible entities matching the question type meant fewer possibilities for distractors.

The performance of the model is also a function of the varying difficulty of the passages. We found the majority of the passages to be short declarative texts with a simple syntactic structure, few named entities, and low amount of information. Generating and answering questions from those passages is a rather trivial task. The selection of more complex paragraphs will likely result in a lower performance of the model and a lot more possibilities for creative and successful adversarial attacks.

8 Conclusions

In this paper we presented the strategies used by team "longhorns" for Task 1 of DADC: generating high-quality adversarial examples. We obtain the best results in the competition by taking a systematic approach, using linguistic knowledge, and working in a collaborative environment.

Our approach outperforms prior work in terms of model error rate and also provides a variety of insights. For instance, our pilot analysis covers a large number of linguistic and reasoning phenomena and explores different model biases. This facilitates a more in-depth analysis of the performance of the model. The systematic approach also gives us insight into the quality and difficulty of the data.

Our strategies for generating adversarial examples are not limited to extractive question answering. They can be adopted at larger scale to improve the quality of models and data on a variety of different tasks. We believe that our work opens new research directions with both scientific and practical implications.

Acknowledgements

This research was supported in part by NSF grants IIS-1850153 and IIS-2107524, as well as by Wipro, the Knight Foundation, the Micron Foundation, and by Good Systems,[4] a UT Austin Grand Challenge to develop responsible AI technologies. The statements made herein are solely the opinions of the authors and do not reflect the views of the sponsoring agencies.

References

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: investigating adversarial human annotations for reading comprehension. *CoRR*, abs/2002.00293.

Idan Blank, Nancy Kanwisher, and Evelina Fedorenko. 2014. A functional dissociation between language and multiple-demand systems revealed in patterns of bold signal fluctuations. *Journal of neurophysiology*, 112(5):1105–1118.

Samuel Bowman. 2022. The dangers of underclaiming: Reasons for caution when reporting how NLP systems fail. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7484–7499, Dublin, Ireland. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Herbert H Clark. 1996. *Using language*. Cambridge university press.

Evgeniia Diachek, Idan Blank, Matthew Siegelman, Josef Affourtit, and Evelina Fedorenko. 2020. The domain-general multiple demand (md) network does not support core aspects of language comprehension: a large-scale fmri investigation. *Journal of Neuroscience*, 40(23):4536–4550.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019.

[4] http://goodsystems.utexas.edu/

DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982, Seattle, Washington, USA. Association for Computational Linguistics.

Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021. Back to square one: Artifact detection, training and commonsense disentanglement in the Winograd schema. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10486–10500, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fernanda Ferreira and John M Henderson. 1991. Recovery from misanalyses of garden-path sentences. *Journal of Memory and Language*, 30(6):725–745.

Lyn Frazier and Keith Rayner. 1982. Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive psychology*, 14(2):178–210.

Lynn Frazier. 1979. On comprehending sentences: Syntactic parsing strategies. *ETD Collection for University of Connecticut*.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Edward Gibson, Richard Futrell, Steven P Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How efficiency shapes human language. *Trends in cognitive sciences*, 23(5):389–407.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.

Elsi Kaiser and Emily Fedele. 2019. Reference resolution. *The Oxford Handbook of Reference*.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Divyansh Kaushik, Douwe Kiela, Zachary C. Lipton, and Wen-tau Yih. 2021. On the efficacy of adversarial data collection for question answering: Results from a large-scale randomized study. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6618–6633, Online. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. *ArXiv*, abs/2004.14602.

Venelin Kovatchev, M. Antònia Martí, and Maria Salamó. 2018. ETPC - a paraphrase identification corpus annotated with extended paraphrase typology and negation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Venelin Kovatchev, M. Antonia Marti, Maria Salamo, and Javier Beltran. 2019. A qualitative evaluation framework for paraphrase identification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 568–577, Varna, Bulgaria. INCOMA Ltd.

Venelin Kovatchev, Phillip Smith, Mark Lee, Imogen Grumley Traynor, Irene Luque Aguilera, and Rory Devine. 2020. "what is on your mind?" automated scoring of mindreading in childhood and early adolescence. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6217–6228, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Maryellen C. MacDonald. 2013. How language production shapes language form and comprehension. *Frontiers in Psychology*, 4:226.

Gary Marcus. 2020. The next decade in ai: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 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.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019. Analyzing compositionality-sensitivity of nli models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6867–6874.

Jason Phang, Angelica Chen, William Huang, and Samuel R Bowman. 2021. Adversarially constructed evaluation sets are more challenging, but may not be fair. *arXiv preprint arXiv:2111.08181*.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. DynaSent: A dynamic benchmark for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 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.

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. 2020. ConjNLI: Natural language inference over conjunctive sentences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8240–8252, Online. Association for Computational Linguistics.

Dan Sperber and Deirdre Wilson. 1986. *Relevance: Communication and cognition*, volume 142. Citeseer.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682, Online. Association for Computational Linguistics.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Henry M Wellman. 1992. *The child's theory of mind.* The MIT Press.

Adina Williams, Tristan Thrush, and Douwe Kiela. 2022. ANLIzing the adversarial natural language inference dataset. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 23–54, online. Association for Computational Linguistics.