Advancing with Adversaries: Comparing LSTMs Across Adversarial Inputs

Angela Chen
achen19@stanford.edu

Darian Martos
dtmartos@stanford.edu

Jerold Yu
jeroldyu@stanford.edu

Abstract

Recently, more effort has been placed to make reading comprehension systems more robust against adversarial examples. This includes the inclusion of unanswerable questions in the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2018) and the generation of adversarial evaluation schemes (Jia and Liang, 2017). Robust systems are suggested to have "real language understanding abilities" (Jia and Liang, 2017) and are more transferrable to real-world question answering tasks, such as ones that involve social media posts. Starting with the CS224N baseline BiDAF model, we improved on its performance on SQuAD and on adversarial datasets by (1) implementing character embeddings and (2) replacing the BiDAF attention with a reattention mechanism. As of March 17, 2019, our model achieved an EM score of 59.121 and an average F1 score of 62.979 on the Non-PCE Test SQuAD leaderboard. We hope that our modifications to the BiDAF model provide a framework for the robustness of future models against adversarial data.

1 Introduction

In artificial intelligence, many models are built with the goal of mirroring human intelligence. However, while models can perform well on standard evaluation metrics (e.g. evaluation on a held-out test set), it is difficult to assess whether good performance is an indicator of human intelligence. For instance, such models could have been trained in a way that happened to be predictive of the test set examples; in doing so, it gives the false impression that they can perform well with different types of examples.

In NLP, more tests have been created in order to ensure that systems are robust. In 2017, Jia and Liang aimed to reward systems with "real language understanding abilities" by creating adversarially-generated examples. They found that the performance of many of the models published on the SQuAD leaderboard at the time suffered greatly when dealing with adversarial data; the average accuracy across sixteen published models dropped from 75% to 36% (Jia and Liang, 2017). Additionally, SQuAD itself changed in 2018 by introducing examples that were unanswerable (Rajpurkar et al., 2018). It was found that a model that achieved an F1 score of 86% on SQuAD 1.1 dropped to a score of 66% on SQuAD 2.0 (Rajpurkar et al., 2018).

The hope is that reading comprehension systems that can adapt to adversarial data will display a better understanding of language and be more applicable in different settings. In the context of question answering, for instance, one will likely encounter irrelevant information when attempting to answer a question on text from social media posts. The idea that systems can be applied in more real-world contexts merits further investigation into improving on adversarial data.

1.1 Problem Description

Given a context and a question, a machine must read and understand the context, and then find the correct answer to the question. Let the context $C = \{x_1, ..., x_M\}$ and the question $Q = \{x_1, ..., x_N\}$

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

be represented as a series of word tokens, where M is the number of words in the context and N is the number of words in the question. In SQuAD 2.0 (Rajpurkar et al., 2018), the answer A may or may not exist depending on the context and question. If an answer does exist, A is guaranteed to be a contiguous span in C (i.e. $A = \{x_{start}, ..., x_{start+k}\}$), where $k \leq M$ is the number of words in the answer.

1.2 Related Work

Early attempts at tackling adversarial data focused on introducing nonlinearities within neural networks, arguing that the limited flexibility of linear features makes it vulnerable to adversarial perturbations (i.e. examples that look similar to the examples the model was trained on) (Goodfellow et al., 2015). Thus, in papers that attempt to build robust reading comprehension systems on SQuAD, introducing nonlinearities is a common technique. Hu et al. introduced a new architecture known as the Reinforced Mnemonic Reader, a system that relies on a reattention mechanism and tweaks to traditional reinforcement learning algorithms (2018a). The reattention mechanism's multi-round alignment architecture allowed models to capture complex interactions between the question and context better, avoiding problems of redundancy and deficiency within traditional attentive systems (Hu et al., 2018a).

With the introduction of SQuAD 2.0, more systems have been created with the goal of identifying no-answer problems better. Hu et al. proposed a "Read + Verify" system, which leverages a no-answer reader and an answer verifier to both produce no-answer probabilities and check the validity of the predicted answer (2018b). By leveraging the use of pointer networks and the calculation of no-answer scores, they created a modified objective loss function that encourages the prediction of no-answer (Hu et al., 2018b). This model was able to achieve state-of-the-art results on SQuAD at the time of its submission (August 2018), suggesting progress on unanswerable questions.

2 Approach

In this section, we introduce modifications to the CS224N baseline BiDAF model[1], particularly in becoming more robust against adversarial examples.

2.1 Baseline

The CS224N baseline BiDAF model is quite similar to the architecture described in Seo, et al. (2017). The two significant differences are that the baseline does not contain a character embedding layer and has functionality to predict no answer. To allow the model to make no-answer predictions, an out-of-vocabulary token is prepended to each sequence; it then follows its normal course of prediction. The baseline BiDAF model achieved an EM score of 56.298 and an average F1 score of 59.920 on the SQuAD test set. It also achieved an average F1 score of UNK on AddSent examples and UNK on AddOneSent examples.

2.2 Model

Character embeddings We first added character embeddings to the baseline to match the BiDAF model from Seo et al. (2017). We obtained character-level embeddings for each word using Convolutional Neural Networks (CNN), choosing a filter size f equal to the word embedding size D and a kernel size k equal to 5. The character embeddings will help the model better represent the internal structure of words and predict out-of-vocabulary words.

Attention computation We modified the attention scoring to be multiplicative. Luong et al. proposed that such an approach would be more relevant in reading comprehension, as each question-answer pair will vary substantially (2015). More formally, the attention is computed as such:

$$a_i = \bar{S}_i W_a q \ \ \forall i \in \{1, ..., N\}$$
$$b_i = S'_i W_b c \ \ \forall i \in \{1, ..., N\}$$

[1] Starter code for baseline found at http://github.com/chrischute/squad.

Here, a_i and b_i is the attention outputs for the Context-to-Question (C2Q) and Question-to-Context (Q2C) Attentions, respectively; $S \in \mathbb{R}^{N \times M}$ is a similarity matrix between the context hidden states c and question hidden states q; and $W_a, W_b \in \mathbb{R}^{2H \times 2H}$, where H is the hidden size, are learnable weight matrices.

Reattention mechanism We also modified the single-round alignment architecture for the BiDAF model in order to capture complex interactions between the question and context. Akin to Hu, et al. (2018a), we implemented a reattention mechanism in which attention is temporarily memorized in a multi-round alignment architecture in order to refine future attentions. This architecture contains a stack of three alignment layers, all containing:

- alignment $H = \{h_j\}_{j=1}^{N} \in \mathbb{R}^{2H \times N}$ between the question Q and the context C;
- alignment $Z = \{z_j\}_{j=1}^{N} \in \mathbb{R}^{2H \times N}$ between the C and itself; and
- a fully-aware context representation $R = [r_1, ..., r_N]$ used for evidence collection.

In a single layer, the following matrices are computed as such:

- H: starting from the similarity matrix S as described earlier, an attended question vector $\tilde{q_j} = C * \text{softmax}(S_{:j})$ is first computed. Then, for all j, $h_j = \text{fusion}(c_j, \tilde{q_j})$, where $\text{fusion}(x, y)$ is as described in Hu, et al. (2018a).
- Z: self-alignment is first applied to generate another similarity matrix $B \in \mathbb{R}^{N \times N}$, where $B_{ij} = \mathbb{1}_{\{i \neq j\}} f(h_i, h_j)$ (note: $f(x, y)$ is the function used to compute the similarity matrix for the baseline). An attended context vector $\tilde{h_j} = H * \text{softmax}(B_{:j})$ is then computed, which helps generate $z_j = \text{fusion}(h_j, \tilde{h_j})$.
- R: using Z from each subsequent layer as inputs, a BiLSTM is used to generate R. For example, on the second layer, $r_j^2 = \text{BiLSTM}([z_j^1, z_j^2])$. This is used as the hidden representation of the context in the next layer $t \in \{2, 3\}$.

In the second and third layers (i.e. $t \in \{2, 3\}$), the reattention mechanism is used, which uses past attentions to help calculate current attention. Particularly, the similarity matrices S^t and B^t are now computed as such:

$$\begin{aligned} \tilde{S_{ij}^t} &= \text{softmax}(S_{i:}^{t-1}) * \text{softmax}(B_{:j}^{t-1}) \\ S_{ij}^t &= f(q_i, r_j^{t-1}) + \gamma \tilde{S_{ij}^t} \\ \tilde{B_{ij}^t} &= \text{softmax}(B_{i:}^{t-1}) * \text{softmax}(B_{:j}^{t-1}) \\ B_{ij}^t &= \mathbb{1}_{i \neq j}(f(h_i^t, h_j^t) + \gamma \tilde{B_{ij}^t}). \end{aligned}$$

Here, γ is a learnable parameter. From this, one should be able to follow the equations above to generate the necessary quantities in all three layers. This architecture ultimately outputs $R^3 = [r_1^3, ..., r_N^3]$, the final fully-aware context vectors. It is hypothesized that the reattention mechanism can better capture complex interactions between the question and the context, yielding better results against adversarial examples (Jia and Liang, 2017).

3 Experiments

3.1 Setup

Our experiments focus primarily on the SQuAD 2.0 dataset, which evaluates reading comprehension for models. Using paragraphs from Wikipedia, our models will be given a paragraph and a question about the paragraph as an input. The model's goal will be to answer the question correctly. There are over 150,000 questions in total, about half of which are not answerable. If an answer does exist for a question though, the answer is guaranteed to be a chunk of text taken directly from the paragraph.

To further ensure that our models have "real" language understanding abilities, we will also test our models against two adversarial datasets, AddSent and AddOneSent (Jia and Liang, 2017). AddSent paragraphs contain adversarially-generated sentences that look similar to the question at the end of the

Table 1: Performance on SQuAD

		Dev		Test	
Model	Train time	EM	F1	EM	F1
BiDAF Baseline	8 hrs	55.991	59.291	56.298	59.920
BiDAF + Char embed	17.5 hrs	61.099	64.412	58.394	62.413
BiDAF + Char embed + Reattention	26 hrs	60.494	63.897	59.121	62.979

paragraph, whereas AddOneSent paragraphs have randomly-generated, human-approved sentences instead (Jia and Liang, 2017).

The official evaluation criteria for the SQuAD datasets are Exact Match (EM) and F1 score. EM measures how many predicted answers match the correct answer exactly, whereas F1 scores are a weighted average of the precision and recall. For the adversarial datasets, we will only measure our models' average F1 score across the adversarial examples. This is because only the F1 score is shown for our given baseline.

3.2 Implementation

We will first train our models by minimizing the negative log likelihood averaged across the batches, using the Adadelta optimizer. The batch size is 64 and a dropout rate of 0.2 is used to prevent overfitting. The word embeddings are 300-dimensional pre-trained GloVe word vectors. The character embedding dimensions is 64 and the hidden size is 100.

3.3 Results

The models were submitted to the Test Non-PCE SQuAD leaderboard. The resulting EM and F1 scores, along with the time it took to train each model, is shown in Table 1. Our full model (i.e. BiDAF + character embeddings + reattention) performed the best on the test set, though its performance is quite comparable to the BiDAF model with just character embeddings included. In fact, the character embeddings did outperform the full model on the dev set. Nevertheless, the additions to the baseline model yielded significant improvements.

We expected the performance of the full model to be better than all the other models, but it seems that the introduction of the reattention mechanism yields only slightly better generalization as opposed to the regular BiDAF attention. The motivator for adding the reattention mechanism is that it would have been able to capture complex interactions between the question and context, but perhaps in most cases, this is not quite necessary.

4 Analysis

In this section, we investigate the effect of the reattention mechanism by comparing its performance to the model with only the character embeddings added. As expected, BiDAF with character embeddings increases performance because, for any UNK tokens for out-of-vocabulary words, the character embeddings are still able to create the best word representation and hence make an adequate prediction. The addition of character embeddings, while increasing the number of correct predictions, does not change whether the model thinks a question is N/A or not much better than BiDAF without embeddings, though.

> **Question:** What president eliminated the Christian position in the curriculum?
>
> **Context:** Charles W. Eliot, president 1869–1909, eliminated the favored position of Christianity from the curriculum while opening it to student self-direction. While Eliot was the most crucial figure in the secularization of American higher education, he was motivated not by a desire to secularize education, but by Transcendentalist Unitarian convictions. Derived from William Ellery Channing and Ralph Waldo Emerson, these convictions were focused on the dignity and worth of human nature, the right and ability of each person to perceive truth, and the indwelling God in each person.

Answer: Charles W. Eliot

Prediction: Charles W. Eliot

With the addition of the reattention mechanism, more complex connections between passages and questions can be made. In most cases, the full model matches the performance of the character embeddings only model; however, there are cases in which it correctly predicts no answer when the other did not.

Consider the following example in which reattention correctly predicts no answer when the character embeddings model does not:

Character embedding:

> Question: How did peace start?
>
> Context: The war was fought primarily along the frontiers between New France and the British colonies, from Virginia in the South to Nova Scotia in the North. It began with a dispute over control of the confluence of the Allegheny and Monongahela rivers, called the Forks of the Ohio, and the site of the French Fort Duquesne and present-day Pittsburgh, Pennsylvania. The dispute erupted into violence in the Battle of Jumonville Glen in May 1754, during which Virginia militiamen under the command of 22-year-old George Washington ambushed a French patrol.
>
> Answer: N/A
>
> Prediction: with a dispute over control of the confluence of the Allegheny and Monongahela rivers

Full model:

> Question: How did peace start?
>
> Context: The war was fought primarily along the frontiers between New France and the British colonies, from Virginia in the South to Nova Scotia in the North. It began with a dispute over control of the confluence of the Allegheny and Monongahela rivers, called the Forks of the Ohio, and the site of the French Fort Duquesne and present-day Pittsburgh, Pennsylvania. The dispute erupted into violence in the Battle of Jumonville Glen in May 1754, during which Virginia militiamen under the command of 22-year-old George Washington ambushed a French patrol.
>
> Answer: N/A
>
> Prediction: N/A

In this example, we presume that the reattention mechanism is able to recognize that "peace" is not the same as "it" in "it began with a dispute" within the passage. Since reattention is able to detect relationships between passages and questions, it does a better job at detecting when those relationships don't exist at all.

On the other hand, consider the following example in which reattention creates a prediction, albeit an incorrect one. This case, reattention is able to detect that there is a relationship between the question and context (while just the embeddings do not), but it doesn't find the correct answer:

Character embedding:

> Question: Currently, how many votes out of the 352 total votes are needed for a majority?
>
> Context: The second main legislative body is the Council, which is composed of different ministers of the member states. The heads of government of member states also convene a "European Council" (a distinct body) that the TEU article 15 defines as providing the 'necessary impetus for its development and shall define the general political directions and priorities'. It meets each six months and its President (currently former Poland Prime Minister Donald Tusk) is meant to 'drive forward its work', but it does not itself 'legislative functions'. The Council does this: in effect this is the governments of the member states, but there will be a different minister at each meeting, depending on the topic discussed (e.g. for environmental issues, the member states' environment ministers attend and vote; for foreign affairs, the foreign ministers, etc.). The minister must have the authority to

represent and bin the member states in decisions. When voting takes place it is weighted inversely to member state size, so smaller member states are not dominated by larger member states. In total there are 352 votes, but for most acts there must be a qualified majority vote, if not consensus. TEU article 16(4) and TFEU article 238(3) define this to mean at least 55 per cent of the Council members (not votes) representing 65 per cent of the population of the EU: currently this means around 74 per cent, or 260 of the 352 votes. This is critical during the legislative process.

Answer: 260

Prediction: N/A

Full model:

Question: Currently, how many votes out of the 352 total votes are needed for a majority?

Context: The second main legislative body is the Council, which is composed of different ministers of the member states. The heads of government of member states also convene a "European Council" (a distinct body) that the TEU article 15 defines as providing the 'necessary impetus for its development and shall define the general political directions and priorities'. It meets each six months and its President (currently former Poland Prime Minister Donald Tusk) is meant to 'drive forward its work', but it does not itself 'legislative functions'. The Council does this: in effect this is the governments of the member states, but there will be a different minister at each meeting, depending on the topic discussed (e.g. for environmental issues, the member states' environment ministers attend and vote; for foreign affairs, the foreign ministers, etc.). The minister must have the authority to represent and bin the member states in decisions. When voting takes place it is weighted inversely to member state size, so smaller member states are not dominated by larger member states. In total there are 352 votes, but for most acts there must be a qualified majority vote, if not consensus. TEU article 16(4) and TFEU article 238(3) define this to mean at least 55 per cent of the Council members (not votes) representing 65 per cent of the population of the EU: currently this means around 74 per cent, or 260 of the 352 votes. This is critical during the legislative process.

Answer: 260

Prediction: 352

5 Conclusion

Starting from the baseline BiDAF model provided to us, we sought to make it more robust against adversarial data. To do so, we implemented a reattention mechanism that aimed to capture more complex interactions between the question and the context. However, we saw only marginal improvement on the test SQuAD leaderboard. Unfortunately, we were not fully able to analyze the effectiveness of the additions to the model due to certain limitations.

The first limitation we encountered in our paper was that we were unable to evaluate our trained models on the adversarial examples generated by Jia and Liang (2017). There were non-trivial differences in the JSON files between the normal dev evaluation files we normally would use to evaluate the models versus the adversarial JSON file. We hypothesize that these differences arise because the adversarial data was generated as part of SQuAD 1.1, which does not include unanswerable questions. In order to do a more comprehensive analysis on the robustness of our systems, we would ideally have adversarially-generated examples for the most recent iteration of SQuAD.

Additionally, since the training time of our models was a lot slower than what we expected, we were not able to train other iterations of our model before the submission deadline. Most notably, we planned on implementing a no-answer reader that calculates a special no-answer score z in addition to the original span scores (Hu et al., 2018b). This would involve modifying the objective loss function to take into account such no-answer scores. We would also have an independent no-answer loss to further encourage the prediction of no-answer. Our objective loss function would then look like:

$$\mathcal{L}_{\text{joint}} = -\log\left(\frac{(1-\delta)e^{z}+\delta e^{p_{start}[a]p_{end}[b]}}{e^{z}+\sum_{i=1}^{N}\sum_{j=1}^{N} e^{p_{start}[i]p_{end}[j]}}\right)$$

$$\mathcal{L}_{\text{no-answer}} = -(1-\delta)\log\sigma(z) - \delta\log(1-\sigma(z))$$
$$\mathcal{L} = \mathcal{L}_{\text{joint}} + \lambda_n \mathcal{L}_{\text{no-answer}},$$

where δ is an indicator for the answerability of a question, a and b are the ground-truth start and end positions of an answer, and λ_n is a trainable parameter. We hypothesize that optimizing a different objective loss function will yield better results on questions that are unanswerable.

In the paper by Hu et al. (2018b), there was another term that signified an independent span loss that was summed to the objective function. Ideally, one could include such a term, but we chose to omit it because of computational limitations in our project.

References

R. Jia and P. Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Empirical Methods in Natural Language Processing (EMNLP) 2017.*

P. Rajpurkar, R. Jia, and P. Liang. 2018. Know What You Don't Know. Unanswerable Questions for SQuAD. In *ACL 2018.*

I. Goodfellow, J. Shlens, and C. Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR 2015.*

M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. 2017. Bi-Directional Attention Flow for Machine Comprehension. In *ICLR 2017.*

M. Luong, H. Pham, and C. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP 2015.*

M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou. 2018. Reinforced Mnemonic Reader for Machine Reading Comprehension. In *27th International Joint Conference on Artificial Intelligence (IJCAI).*

M. Hu, F. Wei, Y. Peng, Z. Huang, N. Yang, and D. Li. 2018. Read + Verify: Machine Reading Comprehension with Unanswerable Questions. In *AAAI-19.*