Robust Question Answering Through Sub-part Alignment

Jifan Chen and **Greg Durrett**
The University of Texas at Austin
{jfchen, gdurrett}@cs.utexas.edu

Abstract

Current textual question answering (QA) models achieve strong performance on in-domain test sets, but often do so by fitting surface-level patterns, so they fail to generalize to out-of-distribution settings. To make a more robust and understandable QA system, we model question answering as an alignment problem. We decompose both the question and context into smaller units based on off-the-shelf semantic representations (here, semantic roles), and align the question to a subgraph of the context in order to find the answer. We formulate our model as a structured SVM, with alignment scores computed via BERT, and we can train end-to-end despite using beam search for approximate inference. Our use of explicit alignments allows us to explore a set of constraints with which we can prohibit certain types of bad model behavior arising in cross-domain settings. Furthermore, by investigating differences in scores across different potential answers, we can seek to understand what particular aspects of the input lead the model to choose the answer without relying on post-hoc explanation techniques. We train our model on SQuAD v1.1 and test it on several adversarial and out-of-domain datasets. The results show that our model is more robust than the standard BERT QA model, and constraints derived from alignment scores allow us to effectively trade off coverage and accuracy.

1 Introduction

Current text-based question answering models learned end-to-end often rely on spurious patterns between the question and context rather than learning the desired behavior. They may ignore the question entirely (Kaushik and Lipton, 2018), focus primarily on the answer type (Mudrakarta et al., 2018), or otherwise bypass the "intended" mode of reasoning for the task (Chen and Durrett, 2019; Niven and Kao, 2019). Thus, these models are not robust to adversarial attacks (Jia and Liang, 2017; Iyyer et al., 2018; Wallace et al., 2019): they can be fooled by surface-level distractor answers that follow the spurious patterns. Methods like adversarial training (Miyato et al., 2016; Wang and Bansal, 2018; Lee et al., 2019; Yang et al., 2019), data augmentation (Welbl et al., 2020), and posterior regularization (Pereyra et al., 2016; Zhou et al., 2019) have been proposed to improve robustness. However, these techniques often optimize for a certain type of error. We want models that can adapt to new types of adversarial examples and work under other distribution shifts, such as on questions from different text domains (Fisch et al., 2019).

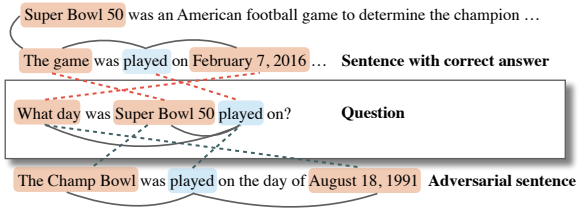


Figure 1: A typical example on adversarial SQuAD. By breaking the question and context down into smaller units, we can expose the incorrect entity match and use explicit constraints to fix it. The solid lines denote edges from SRL and coreference, and the dotted lines denote the possible alignments between the arguments (desired in red, actual in black).

In this paper, we explore a model for text-based question answering through sub-part alignment. The core idea behind our method is that if every aspect of the question is well supported by the answer context, then the answer produced should be trustable (Lewis and Fan, 2018); if not, we suspect that the model is making an incorrect prediction. The sub-parts we use are predicates and arguments from Semantic Role Labeling (Palmer et al., 2005), which we found to be a good semantic representation for the types of questions we studied. We then view the question answering procedure as a constrained graph alignment problem (Sachan and Xing, 2016), where the nodes represent the predi-

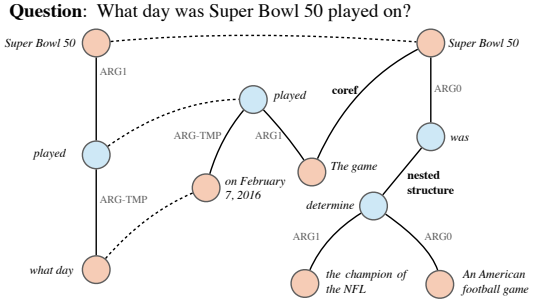
Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1251–1263
June 6–11, 2021.

cates and arguments and the edges are formed by relations between them (e.g. predicate-argument relations and coreference relations). Our goal is to align each node in the question to a counterpart in the context, respecting some loose constraints, and in the end the context node aligned to the wh-span should ideally contain the answer. Then we can use a standard QA model to extract the answer.

Figure 1 shows an adversarial example of SQuAD (Jia and Liang, 2017) where a standard BERT QA model predicts the wrong answer *August 18, 1991*. In order to choose the adversarial answer, our model must **explicitly** align *Super Bowl 50* to *Champ Bowl*. Even if the model still makes this mistake, this error is now exposed directly, making it easier to interpret and subsequently patch.

In our alignment model, each pair of aligned nodes is scored using BERT (Devlin et al., 2019). These alignment scores are then plugged into a beam search inference procedure to perform the constrained graph alignment. This structured alignment model can be trained as a structured support vector machine (SSVM) to minimize alignment error with heuristically-derived oracle alignments. The alignment scores are computed in a black-box way, so these individual decisions aren't easily explainable (Jain and Wallace, 2019); however, the score of an answer is directly a sum of the score of each aligned piece, making this structured prediction phase of the model faithful by construction (Jain et al., 2020). Critically, this allows us to understand what parts of the alignment are responsible for a prediction, and if needed, constrain the behavior of the alignment to correct certain types of errors. We view this interpretability and extensibility with constraints as one of the principal advantages of our model.

We train our model on the SQuAD-1.1 dataset (Rajpurkar et al., 2016) and evaluate on SQuAD Adversarial (Jia and Liang, 2017), Universal Triggers on SQuAD (Wallace et al., 2019), and several out-of-domain datasets from MRQA (Fisch et al., 2019). Our framework allows us to incorporate natural constraints on alignment scores to improve zero-shot performance under these distribution shifts, as well as explore coverage-accuracy tradeoffs in these settings. Finally, our model's alignments serve as "explanations" for its prediction, allowing us to ask why certain predictions are made over others and examine scores for hypothetical other answers the model could give.



Context: Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season … The game was played on February 7, 2016 …

Figure 2: Example of our question-passage graph. Edges come from SRL, coreference (*Super Bowl 50—the game*), and postprocessing of predicates nested inside arguments (*was—determine*). The oracle alignment (Section 3.4) is shown with dotted lines. Blue nodes are predicates and orange ones are arguments.

2 QA as Graph Alignment

Our approach critically relies on the ability to decompose questions and answers into a graph over text spans. Our model can in principle work for a range of syntactic and semantic structures, including dependency parsing, SRL (Palmer et al., 2005), and AMR (Banarescu et al., 2013). We use SRL in this work and augment it with coreference links, due to the high performance and flexibility of current SRL systems (Peters et al., 2018). Throughout this work, we use the BERT-based SRL system from Shi and Lin (2019) and the SpanBERT-based coreference system from Joshi et al. (2020).

An example graph we construct is shown in Figure 2. Both the question and context are represented as graphs where the nodes consist of predicates and arguments. Edges are undirected and connect each predicate and its corresponding arguments. Since SRL only captures the predicate-argument relations within one sentence, we add coreference edges as well: if two arguments are in the same coreference cluster, we add an edge between them. Finally, in certain cases involving verbal or clausal arguments, there might exist nested structures where an argument to one predicate contains a separate predicate-argument structure. In this case, we remove the larger argument and add an edge directly between the two predicates. This is shown by the edge from *was* to *determine* (labeled as *nested structure*) in Figure 2). Breaking down such large arguments helps avoid ambiguity during alignment.

Aligning questions and contexts has proven

useful for question answering in previous work (Sachan et al., 2015; Sachan and Xing, 2016; Khashabi et al., 2018). Our framework differs from theirs in that it incorporates a much stronger alignment model (BERT), allowing us to relax the alignment constraints and build a more flexible, higher-coverage model.

Alignment Constraints Once we have the constructed graph, we can align each node in the question to its counterpart in the context graph. In this work, we control the alignment behavior by placing explicit constraints on this process. We place a **locality constraint** on the alignment: adjacent pairs of question nodes must align no more than k nodes apart in the context. $k = 1$ means we are aligning the question to a connected sub-graph in the context, $k = \infty$ means we can align to a node anywhere in a connected component in the context graph. In our experiments, we set $k = 3$. In the following sections, we will discuss more constraints. Altogether, these constraints define a set $\mathcal{A}$ of possible alignments.

3 Graph Alignment Model

3.1 Model

Let $\mathbf{T}$ represent the text of the context and question concatenated together. Assume a decomposed question graph $\mathbf{Q}$ with nodes $q_1, q_2, \ldots, q_m$ represented by vectors $\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q}_m$, and a decomposed context $\mathbf{C}$ with nodes $c_1, \ldots, c_n$ represented by vectors $\mathbf{c_1}, \ldots, \mathbf{c}_n$. Let $\mathbf{a} = (a_1, \ldots, a_m)$ be an alignment of question nodes to context nodes, where $a_i \in \{1, \ldots, n\}$ indicates the alignment of the ith question node. Each question node is aligned to exactly one context node, and multiple question nodes can align to the same context node.

We frame question answering as a maximization of an alignment scoring function over possible alignments: $\max_{\mathbf{a}\in\mathcal{A}} f(\mathbf{a}, \mathbf{Q}, \mathbf{C}, \mathbf{T})$. In this paper, we simply choose f to be the sum over the scores of all alignment pairs $f(\mathbf{a}, \mathbf{Q}, \mathbf{C}, \mathbf{T}) = \sum_{i=1}^{n} S(q_i, c_{a_i}, \mathbf{T})$, where $S(q, c, \mathbf{T})$ denotes the alignment score between a question node q and a context node c. This function relies on BERT (Devlin et al., 2019) to compute embeddings of the question and context nodes and will be described more precisely in what follows. We will train this model as a structured support vector machine (SSVM), described in Section 3.2.

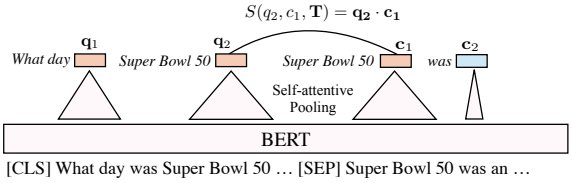


Figure 3: Alignment scoring. Here the alignment score is computed by the dot product between span representations of question and context nodes. The final alignment score (not shown) is the sum of these edge scores.

Scoring Our alignment scoring process is shown in Figure 3. We first concatenate the question text with the document text into $\mathbf{T}$ and then encode them using the pre-trained BERT encoder. We then compute a representation for each node in the question and context using a span extractor, which in our case is the self-attentive pooling layer of Lee et al. (2017). The node representation in the question can be computed in the same way. Then the score of a node pair is computed as a dot product $S(q, c, \mathbf{T}) = \mathbf{q} \cdot \mathbf{c}$.

Answer Extraction Our model so far produces an alignment between question nodes and context nodes. We assume that one question node contains a wh-word and this node aligns to the context node containing the answer.[1] Ideally, we can use this aligned node to extract the actual answer. However, in practice, the aligned context node may only contain part of the answer and in some cases answering the question only based the aligned context node can be ambiguous. We therefore use the sentence containing the wh-aligned context node as the "new" context and use a standard BERT QA model to extract the actual answer post-hoc. In the experiments, we also show the performance of our model by only use the aligned context node without the sentence, which is only slightly worse.

3.2 Training

We train our model as an instance of a structured support vector machine (SSVM). Ignoring the regularization term, this objective can be viewed as a sum over the training data of a structured hinge loss with the following formulation:

$$\sum_{i=1}^{N} \max(0, \max_{\mathbf{a}\in\mathcal{A}} [f(\mathbf{a}, \mathbf{Q}_i, \mathbf{C}_i, \mathbf{T}_i) + \text{Ham}(\mathbf{a}, \mathbf{a}_i^*) - f(\mathbf{a}_i^*, \mathbf{Q}_i, \mathbf{C}_i, \mathbf{T}_i)])$$

[1] We discuss what to do with other questions in Section 4.1.

where $\mathbf{a}$ denotes the predicted alignment, $\mathbf{a}_i^*$ is the oracle alignment for the ith training example, and Ham is the Hamming loss between these two. To get the predicted alignment $\mathbf{a}$ during training, we need to run loss-augmented inference as we will discuss in the next section. When computing the alignment for node j, if $a_j \neq a_j^*$, we add 1 to the alignment score to account for the loss term in the above equation. Intuitively, this objective requires the score of the gold prediction to be larger than any other hypothesis $\mathbf{a}$ by a margin of $\text{Ham}(\mathbf{a}, \mathbf{a}^*)$.

When training our system, we first do several iterations of *local training* where we treat each alignment decision as an independent prediction, imposing no constraints, and optimize log loss over this set of independent decisions. The local training helps the global training converge more quickly and achieve better performance.

3.3 Inference

Since our alignment constraints do not strongly restrict the space of possible alignments (e.g., by enforcing a one-to-one alignment with a connected subgraph), searching over all valid alignments is intractable. We therefore use beam search to find the approximate highest-scoring alignment: (1) Initialize the beam with top b highest aligned node pairs, where b is the beam size. (2) For each hypothesis (partial alignment) in the beam, compute a set of reachable nodes based on the currently aligned pairs under the locality constraint. (3) Extend the current hypothesis by adding each of these possible alignments in turn and accumulating its score. Beam search continues until all the nodes in the question are aligned.

An example of one step of beam hypothesis expansion with locality constraint $k = 2$ is shown in Figure 4. In this state, the two *played* nodes are already aligned. In any valid alignment, the neighbors of the *played* question node must be aligned within 2 nodes of the *played* context node to respect the locality constraint. We therefore only consider aligning to *the game*, *on Feb 7, 2016* and *Super Bowl 50*. The alignment scores between these reachable nodes and the remaining nodes in the question are computed and used to extend the beam hypotheses.

Note that this inference procedure allows us to easily incorporate other constraints as well. For instance, we could require a "hard" match on entity nodes, meaning that two nodes containing entities can only align if they share entities. With this constraint, as shown in the figure, *Super Bowl 50* can never be aligned to *on February 7, 2016*. We discuss such constraints more in Section 5.

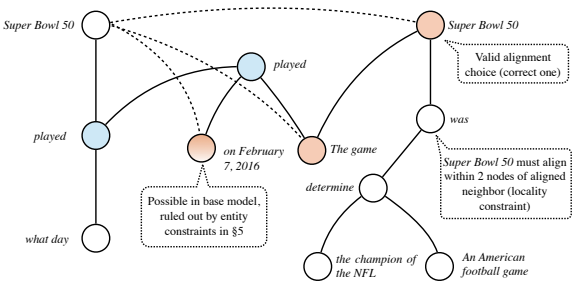


Figure 4: An example of constraints during beam search. The blue node *played* is already aligned. The orange nodes denote all the valid context nodes that can be aligned to for both *Super Bowl 50* and *what day* in the next step of inference given the locality constraint with $k = 2$.

3.4 Oracle Construction

Training assumes the existence of gold alignments $\mathbf{a}^*$, which must be constructed via an oracle given the ground truth answer. This process involves running inference based on heuristically computed alignment scores S_{oracle}, where $S_{\text{oracle}}(q, c)$ is computed by the Jaccard similarity between a question node q and a context node c. Instead of initializing the beam with the b best alignment pairs, we first align the wh-argument in the question with the node(s) containing the answer in the context and then initialize the beam with those alignment pairs.

If the Jaccard similarity between a question node and all other context nodes is zero, we set these as unaligned nodes. During training, our approach can gracefully handle unaligned nodes by treating these as latent variables in structured SVM: the gold "target" is then highest scoring set of alignments consistent with the gold supervision. This involves running a second decoding step on each example to impute the values of these latent variables for the gold alignment.

4 Experiments: Adversarial and Cross-domain Robustness

Our focus in this work is primarily robustness, interpretability, and controllability of our model. We focus on adapting to challenging settings in order to "stress test" our approach.

	SQuAD normal		SQuAD addSent		Natural Questions		NewsQA		BioASQ		TBQA	
	ans in wh	F1	ans in wh	F1	ans in wh	F1	ans in wh	F1	ans in wh	F1	ans in wh	F1
Sub-part Alignment	84.7	84.5	49.5	**50.5**	65.8	61.5	49.3	48.1	63.5	**53.4**	35.1	**38.4**
− global train+inf	85.8	85.2	45.0	46.8	65.9	**62.3**	48.9	47.1	62.5	52.1	31.9	34.6
− ans from full sent	84.7	81.8	49.5	46.7	65.8	57.8	49.3	45.0	63.5	51.1	35.1	37.5
BERT QA	–	**87.8**	–	39.2	–	59.4	–	**48.5**	–	52.4	–	25.3

Table 1: The performance and ablations of our proposed model on the development sets of SQuAD, adversarial SQuAD, and four out-of-domain datasets. Our Sub-part Alignment model uses both global training and inference as discussed in Section 3.2-3.3. − **global train+inf** denotes the locally trained and evaluated model. − **ans from full sent** denotes extracting the answer using only the wh-aligned node. **ans in wh** denotes the percentage of answers found in the span aligned to the wh-span, and F1 denotes the standard QA performance measure. Here for `addSent`, we only consider the adversarial examples. Note also that this evaluation is *only on wh-questions*.

4.1 Experimental Settings

For all experiments, we train our model *only* on the English SQuAD-1.1 dataset (Rajpurkar et al., 2016) and examine how well it can generalize to adversarial and out-of-domain settings with minimal modification, using *no fine-tuning* on new data and *no data augmentation* that would capture useful transformations. We evaluate on the `addSent` and `addOneSent` proposed by Jia and Liang (2017), and the `Universal Triggers on SQuAD` (Wallace et al., 2019). We also test the performance of our SQuAD-trained models in zero-shot adaptation to new English domains, namely Natural Questions (Kwiatkowski et al., 2019), NewsQA (Trischler et al., 2017), BioASQ (Tsatsaronis et al., 2015) and TextbookQA (Kembhavi et al., 2017), taken from the MRQA shared task (Fisch et al., 2019). Our motivation here was to focus on text from a variety of domains where transferred SQuAD models may at least behave credibly. We excluded, for example, HotpotQA (Yang et al., 2018) and DROP (Dua et al., 2019), since these are so far out-of-domain from the perspective of SQuAD that we do not see them as a realistic cross-domain target.

We compare primarily against a standard **BERT QA** system (Devlin et al., 2019). We also investigate a local version of our model, where we only try to align each node in the question to its oracle, without any global training (− **global train + inf**), which can still perform reasonably because BERT embeds the whole question and context. When comparing variants of our proposed model, we only consider the questions that have a valid SRL parse and have a wh word (results in Table 1, Table 2, and Figure 5). When comparing with prior systems, for questions that do not have a valid SRL parse or wh word, we back off to the standard BERT QA system (results in Table 3).

We set the beam size $b = 20$ for the constrained alignment. We use `BERT-base-uncased` for all of our experiments, and fine-tune the model using Adam (Kingma and Ba, 2014) with learning rate set to 2e-5. Our preprocessing uses a SpanBERT-based coreference system (Joshi et al., 2020) and a BERT-based SRL system (Shi and Lin, 2019). We limit the length of the context to 512 tokens. For our global model, we initialize the weights using a locally trained model and then fine-tune using the SSVM loss. We find the initialization helps the model converge much faster and it achieves better performance than learning from scratch. When doing inference, we set the locality constraint $k = 3$.

4.2 Results on Challenging Settings

The results[2] on the normal SQuAD development set and other challenging sets are shown in Table 1.

Our model is not as good as BERT QA on normal SQuAD but outperforms it in challenging settings. Compared to the BERT QA model, our model is fitting a different data distribution (learning a constrained structure) which makes the task harder. This kind of training scheme does cause some performance drop on normal SQuAD, but we can see that it consistently improves the F1 on the adversarial (on SQuAD addSent, a 11.3 F1 improvement over BERT QA) and cross-domain datasets except NewsQA (where it is 0.4 F1 worse). This demonstrates that learning the alignment helps improve the robustness of our model.

[2]Here we omit `SQuAD addOneSent` for simplicity, since the performance on it has the same trend as `SQuAD addSent`. Refer to the Appendix for the results on `SQuAD addOneSent`.

Type	Sub-part Alignment Normal	Trigger	Δ	BERT Normal	Trigger	Δ
who	84.7	82.7	2.0	87.1	78.5	8.6
why	75.1	71.3	3.8	76.5	59.7	16.8
when	88.4	82.8	5.6	90.3	80.9	9.4
where	83.6	81.4	2.2	84.1	75.8	8.3

Table 2: The performance of our model on the Universal Triggers on SQuAD dataset (Wallace et al., 2019). Compared with BERT, our model sees smaller performance drops on all triggers.

Global training and inference improve performance in adversarial settings, despite having no effect in-domain. Normal SQuAD is a relatively easy dataset and the answer for most questions can be found by simple lexical matching between the question and context. From the ablation of **− global train+inf**, we can see that more than 80% of answers can be located by matching the wh-argument. We also observe a similar pattern on Natural Questions.[3] However, as there are very strong distractors in `SQuAD addSent`, the wh-argument matching is unreliable. In such situations, the constraints imposed by other argument alignments in the question are useful to correct the wrong wh-alignment through global inference. We see that the global training plus inference is consistently better than the local version on all other datasets.

Using the strict wh answer extraction still gives strong performance From the ablation of **− ans from full sent**, we observe that our "strictest" system that extracts the answer only using the wh-aligned node is only worse by 3-4 points of F1 on most datasets. Using the full sentence gives the system more context and maximal flexibility, and allows it to go beyond the argument spans introduced by SRL. We believe that better semantic representations tailored for question answering (Lamm et al., 2020) will help further improvement in this regard.

4.3 Results on Universal Triggers

The results on subsets of the universal triggers dataset are shown in Table 2. We see that every trigger results in a bigger performance drop on BERT QA than our model. Our model is much more stable, especially on *who* and *where* question types, in which case the performance only drops by around 2%. Several factors may contribute to the stability: (1) The triggers are ungrammatical and their arguments often contain seemingly random words, which are likely to get lower alignment scores. (2) Because our model is structured and trained to align all parts of the question, adversarial attacks on span-based question answering models may not fool our model as effectively as they do BERT.

4.4 Comparison to Existing Systems

In Table 3, we compare our best model (not using constraints from Section 5) with existing adversarial QA models in the literature. We note that the performance of our model on SQuAD-1.1 data is relatively lower compared to those methods, yet we achieve the best overall performance; we trade some in-distribution performance to improve the model's robustness. We also see that our model achieves the smallest normal vs. adversarial gap on `addSent` and `addOneSent`, which demonstrates that our constrained alignment process can enhance the robustness of the model compared to prior methods like adversarial training (Yang et al., 2019) or explicit knowledge integration (Wang and Jiang, 2018).

5 Generalizing by Alignment Constraints

One advantage of our explicit alignments is that we can understand and inspect the model's behavior more deeply. This structure also allows us to add constraints to our model to prohibit certain behaviors, which can be used to adapt our model to adversarial settings.

In this section, we explore how two types of constraints enable us to reject examples the model is less confident about. Hard constraints can enable us to reject questions where the model finds no admissible answers. Soft constraints allow us to set a calibration threshold for when to return our answer. We focus on evaluating our model's accuracy at various coverage points, the so-called selective question answering setting (Kamath et al., 2020).

Constraints on Entity Matches By examining `addSent` and `addOneSent`, we find the model is typically fooled when the nodes containing entities in the question align to "adversarial" entity nodes. An intuitive constraint we can place on the alignment is that we require a hard entity match—for each argument in the question, if it contains

[3]For the MRQA task, only the paragraph containing the short answer of NQ is provided as context, which eliminates many distractors. In such cases, those NQ questions have a similar distribution as those in SQuAD-1.1, and similarly make no use of the global alignment.

	Normal	addSent			addOneSent		
		overall	adv	Δ	overall	adv	Δ
R.M-Reader (Hu et al., 2018)	86.6	58.5	–	31.1	67.0	–	19.6
KAR (Wang and Jiang, 2018)	83.5	60.1	–	23.4	72.3	–	**11.2**
BERT + Adv (Yang et al., 2019)	92.4	63.5	–	28.9	72.5	–	19.9
Our BERT	87.8	61.8	39.2	27.0	70.4	52.6	18.4
Sub-part Alignment*	84.7	**65.8**	47.1	**18.9**	**72.8**	60.1	11.9

Table 3: Performance of our systems compared to the literature on both `addSent` and `addOneSent`. Here, overall denotes the performance on the full adversarial set, adv denotes the performance on the adversarial samples alone. Δ represents the gap between the normal SQuAD and the overall performance on adversarial set.

entities, it can only align to nodes in the context sharing exact the same entities.

Constraints on Alignment Scores The hard entity constraint is quite inflexible and does not generalize well, for example to questions that do not contain a entity. However, the alignment scores we get during inference time are good indicators of how well a specific node pair is aligned. For a correct alignment, every pair should get a reasonable alignment score. However, if an alignment is incorrect, there should exist some bad alignment pairs which have lower scores than the others. We can reject those samples by finding bad alignment pairs, which both improves the precision of our model and also serves as a kind of explanation as to why our model makes its predictions.

We propose to use a simple heuristic to identify the bad alignment pairs. We first find the max score $S_{\max}$ over all possible alignment pairs for a sample, then for each alignment pair (q_i, c_j) of the prediction, we calculate the worst alignment gap (WAG) $g = \min_{(q,c)\in\mathbf{a}}(S_{\max} - S(q,c))$. If g is beyond some threshold, it indicates that alignment pair is not reliable.[4]

Comparison to BERT Desai and Durrett (2020) show that pre-trained transformers like BERT are well-calibrated on a range of tasks. Since we are rejecting the unreliable predictions to improve the precision of our model, we reject the same number of examples for the baseline using the posterior probability of the BERT QA predictions. To be specific, we rank the predictions of all examples by the sum of `start` and `end` posterior probabilities and compute the F1 score on the top k predictions.

[4]The reason we look at differences from the max alignment is to calibrate the scores based on what "typical" scores look like for that instance. We find that these are on different scales across different instances, so the gap is more useful than an absolute threshold.

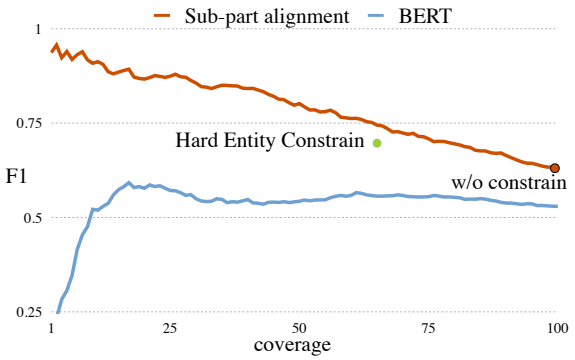


Figure 5: The F1-coverage curve of our model compared with BERT QA. If our model can choose to answer only the k percentage of examples it's most confident about (the coverage), what F1 does it achieve? For our model, the confidence is represented by our "worst alignment gap" (WAG) metric. Smaller WAG indicates higher confidence. For BERT, the confidence is represented by the posterior probability.

5.1 Results on Constrained Alignment

On Adversarial SQuAD, the confidence scores of a normal BERT QA model do not align with its performance. From Figure 5, we find that the highest-confidence answers from BERT (i.e., in low coverage settings) are very inaccurate. One possible explanation of this phenomenon is that BERT overfits to the pattern of lexical overlap, and is actually most confident on adversarial examples highly similar to the input. In general, BERT's confidence is not an effective heuristic for increasing accuracy.

Hard entity constraints improve the precision but are not flexible. Figure 5 also shows that by adding a hard entity constraint, we achieve a 71.4 F1 score which is an 8.6 improvement over the unconstrained model at a cost of only 60% of samples being covered. Under the hard entity constraint, the model is not able to align to the nodes in the adversarial sentence, but the performance is still

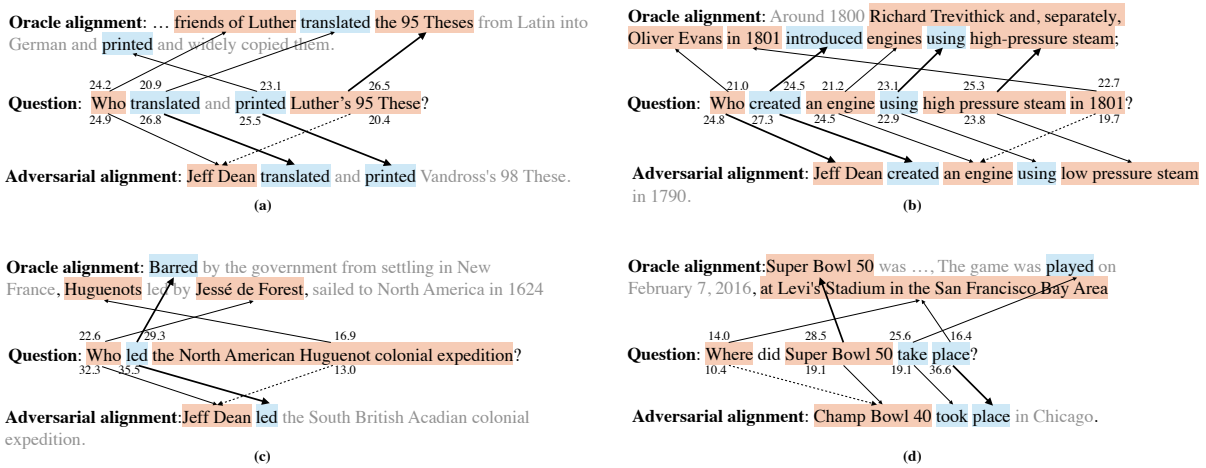


Figure 6: Examples of alignment of our model on `addOneSent`: both the correct alignment and also adversarial alignment are shown. The numbers are the actual alignment scores of the model's output. Dashed arrows denote the least reliable alignments and bolder arrows denote the alignment that contribute more to the model's prediction.

lower than what it achieves on normal SQuAD. We examine some of the error cases and find that for a certain number of samples, there is no path from the node satisfying the constraint to the node containing the answer (e.g. they hold a more complex discourse relation while we only consider coreference as cross-sentence relation). In such cases, our method cannot find the answer.

A smaller worst alignment gap indicates better performance. As opposed to BERT, our alignment score is well calibrated on those adversarial examples. This substantiates our claim that those learned alignment scores are good indicators of how trustful alignment pairs are. Also, we see that when the coverage is the same as the entity constraint, the performance under the alignment score constraint is even better. The alignment constraints are simultaneously more flexible than the hard constraint and also more effective.

5.2 Case Study on Alignment Scores

In this section, we give several examples of the alignment and demonstrate how those scores can act as an explanation to the model's behavior. Those examples are shown in Figure 6.

As shown by the dashed arrows, all adversarial alignments contain at least one alignment with significantly lower alignment score. The model is overconfident towards the other alignments with a high lexical overlap as shown by the bold arrows. These overconfident alignments also show that the predicate alignment learned on SQuAD-1.1 is not reliable. To further improve the quality of predicate alignment, either a more powerful training set or a new predicate alignment module is needed.

Crucially, with these scores, it is easy for us to interpret our model's behavior. For instance, in example (a), the very confident predicate alignment forces *Luther's 95 Theses* to have no choice but align to *Jeff Dean*, which is unrelated. Because we have alignments over the sub-parts of a question, we can inspect our model's behavior in a way that the normal BERT QA model does not allow. We believe that this type of debuggability provides a path forward for building stronger QA systems in high-stakes settings.

6 Related Work

Adversarial Attacks in NLP. Adversarial attacks in NLP may take the form of adding sentences like adversarial SQuAD (Jia and Liang, 2017), universal adversarial triggers (Wallace et al., 2019), or sentence perturbations: Ribeiro et al. (2018) propose deriving transformation rules, Ebrahimi et al. (2018) use character-level flips, and Iyyer et al. (2018) use controlled paraphrase generation. The highly structured nature of our approach makes it more robust to such attacks and provides hooks to constrain the system to improve performance further.

Neural module networks. Neural module networks are a class of models that decompose a task into several sub-tasks, addressed by independent neural modules, which make the model more robust and interpretable (Andreas et al., 2016; Hu et al., 2017; Cirik et al., 2018; Hudson and Manning, 2018; Jiang and Bansal, 2019). Like these, our model is trained end-to-end, but our approach

uses structured prediction and a static network structure rather than dynamically assembling a network on the fly. Our approach could be further improved by devising additional modules with distinct parameters, particularly if these are trained on other datasets to integrate additional semantic constraints.

Unanswerable questions Our approach rejects some questions as unanswerable. This is similar to the idea of unanswerable questions in SQuAD 2.0 (Rajpurkar et al., 2018), which have been studied in other systems (Hu et al., 2019). However, techniques to reject these questions differ substantially from ours – many SQuAD 2.0 questions require not only a correct alignment between the question and context but also need to model the relationship between arguments, which is beyond the scope of this work and could be a promising future work. Also, the setting we consider here is more challenging, as we do not assume access to such questions at training time.

Graph-based QA Khashabi et al. (2018) propose to answer questions through a similar graph alignment using a wide range of semantic abstractions of the text. Our model differs in two ways: (1) Our alignment model is trained end-to-end while their system mainly uses off-the-shelf natural language modules. (2) Our alignment is formed as node pair alignment rather than finding an optimal sub-graph, which is a much more constrained and less flexible formalism. Sachan et al. (2015); Sachan and Xing (2016) propose to use a latent alignment structure most similar to ours. However, our model supports a more flexible alignment procedure than theirs does, and can generalize to handle a wider range of questions and datasets.

Past work has also decomposed complex questions to answer them more effectively (Talmor and Berant, 2018; Min et al., 2019; Perez et al., 2020). Wolfson et al. (2020) further introduce a Question Decomposition Meaning Representation (QDMR) to explicitly model this process. However, the questions they answer, such as those from HotpotQA (Yang et al., 2018), are *fundamentally* designed to be multi-part and so are easily decomposed, whereas the questions we consider are not. Our model theoretically could be extended to leverage these question decomposition forms as well.

7 Discussion and Conclusion

We note a few limitations and some possible future directions of our approach. First, errors from SRL and coreference resolution systems can propagate through our system. However, because our graph alignment is looser than those in past work, we did not observe this to be a major performance bottleneck. The main issue here is the inflexibility of the SRL spans. For example, not every SRL span in the question can be appropriately aligned to a single SRL span in the context. Future works focusing on the automatic span identification and alignment like recent work on end-to-end coreference systems (Lee et al., 2017), would be promising.

Second, from the error analysis we see that our proposed model is good at performing noun phrase alignment but not predicate alignment, which calls attention to the better modeling of the predicate alignment process. For example, we can decompose the whole alignment procedure into separate noun phrase and predicate alignment modules, in which predicate alignment could be learned using different models or datasets.

Finally, because our BERT layer looks at the entire question and answer, our model can still leverage uninterpretable interactions in the text. We believe that modifying the training objective to more strictly enforce piecewise comparisons could improve interpretability further while maintaining strong performance.

In this work, we presented a model for question answering through sub-part alignment. By structuring our model around explicit alignment scoring, we show that our approach can generalize better to other domains. Having alignments also makes it possible to filter out bad model predictions (through score constraints) and interpret the model's behavior (by inspecting the scores).

Acknowledgments

This work was partially supported by NSF Grant IIS-1814522 and NSF Grant SHF-1762299. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources used to conduct this research. Results presented in this paper were obtained using the Chameleon testbed supported by the National Science Foundation. Thanks as well to the anonymous reviewers for their helpful comments.

References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. *NAACL*.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. 2018. Using syntax to ground referring expressions in natural images. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of NAACL-HLT*, pages 2368–2378.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension. *arXiv preprint arXiv:1910.09753*.

Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. 2018. Reinforced mnemonic reader for machine reading comprehension. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4099–4106. AAAI Press.

Minghao Hu, Furu Wei, Yu xing Peng, Zhen Xian Huang, Nan Yang, and Ming Zhou. 2019. Read + Verify: Machine Reading Comprehension with Unanswerable Questions. In *Thirty-Third AAAI Conference on Artificial Intelligence*.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813.

Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *NAACL*.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C Wallace. 2020. Learning to faithfully rationalize by construction. *arXiv preprint arXiv:2005.00115*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Yichen Jiang and Mohit Bansal. 2019. Self-assembling modular networks for interpretable multi-hop reasoning. 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 4464–4474.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696.

Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *EMNLP*.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4999–5007.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. 2020. Qed: A framework and dataset for explanations in question answering. *arXiv preprint arXiv:2009.06354*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

Seanie Lee, Donggyu Kim, and Jangwon Park. 2019. Domain-agnostic question-answering with adversarial training. *arXiv preprint arXiv:1910.09342*.

Mike Lewis and Angela Fan. 2018. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the Model Understand the Question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Comput. Linguist.*, 31(1):71–106.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. 2016. Regularizing neural networks by penalizing confident output distributions.

Ethan Perez, Patrick A. Lewis, Wen tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised Question Decomposition for Question Answering. In *arXiv*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *NAACL*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*.

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.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

Mrinmaya Sachan, Kumar Dubey, Eric Xing, and Matthew Richardson. 2015. Learning answer-entailing structures for machine comprehension. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 239–249.

Mrinmaya Sachan and Eric Xing. 2016. Machine comprehension using rich semantic representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 486–492.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. *ACL 2017*, page 191.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia

Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel Ngonga, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Empirical Methods in Natural Language Processing*.

Chao Wang and Hui Jiang. 2018. Explicit utilization of general knowledge in machine reading comprehension. *arXiv preprint arXiv:1809.03449*.

Yicheng Wang and Mohit Bansal. 2018. Robust machine comprehension models via adversarial training. 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 575–581.

Johannes Welbl, Pasquale Minervini, Max Bartolo, Pontus Stenetorp, and Sebastian Riedel. 2020. Undersensitivity in neural reading comprehension. *arXiv preprint arXiv:2003.04808*.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. *EMNLP*.

Ziqing Yang, Yiming Cui, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. 2019. Improving machine reading comprehension via adversarial training. *arXiv preprint arXiv:1911.03614*.

Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. 2019. Robust reading comprehension with linguistic constraints via posterior regularization. *arXiv preprint arXiv:1911.06948*.

A Adversarial Datasets

Added sentences Jia and Liang (2017) propose to append an adversarial distracting sentence to the normal SQuAD development set to test the robustness of a QA model. In this paper, we use the two main test sets they introduced: `addSent` and `addOneSent`. Both of the two sets augment the normal test set with adversarial samples annotated by Turkers that are designed to look similar to question sentences. In this work, we mainly focus on the adversarial examples.

Universal Triggers Wallace et al. (2019) use a gradient based method to find a short trigger sequence. When they insert the short sequence to the original text, it will trigger the target prediction in the sequence independent of the rest of the passage content or the exact nature of the question. For QA, they generate different triggers for different types of questions including "who", "when", "where" and "why".

Datasets from MRQA For Natural Questions (Kwiatkowski et al., 2019), NewsQA (Trischler et al., 2017), BioASQ (Tsatsaronis et al., 2015) and TextbookQA (Kembhavi et al., 2017), we use the pre-processed datasets from MRQA (Fisch et al., 2019). They differ from the original datasets in that only the paragraph containing the answer is picked as the context and the maximum length of the context is cut to 800 tokens.

B Results on SQuAD addOneSent

The results of our model compared to BERT QA on `SQuAD addOneSent` is shown in Table 4. Here we see the results on `addOneSent` and `addSent` generally have the same trend. The **global train+inf** helps more on the more difficult `addSent`.

	SQuAD normal		SQuAD addSent		SQuAD addOneSent	
	ans in wh	F1	ans in wh	F1	ans in wh	F1
Sub-part Alignment	84.7	84.5	49.5	**50.5**	61.9	**62.8**
- global train+inf	85.8	85.2	45.0	46.8	58.9	59.6
- ans from full sent	84.7	81.8	49.5	46.7	61.9	59.2
BERT QA	–	87.8	–	39.2	–	52.6

Table 4: The performance and ablations of our proposed model on the development set of SQuAD normal, SQuAD addSent, and SQuAD addOneSent. – **global train+inf** denotes the locally trained and evaluated model. – **ans from full sent** denotes extracting the answer using only the wh-aligned node. **ans in wh** denotes the percentage of answers found in the span aligned to the wh-span, and F1 denotes the standard QA performance measure.