Semantics Altering Modifications for Evaluating Comprehension in Machine Reading

Viktor Schlegel, Goran Nenadic and Riza Batista-Navarro
Department of Computer Science, University of Manchester
Manchester, United Kingdom
{viktor.schlegel,gnenadic,riza.batista}@manchester.ac.uk

Abstract

Advances in NLP have yielded impressive results for the task of machine reading comprehension (MRC), with approaches having been reported to achieve performance comparable to that of humans. In this paper, we investigate whether state-of-the-art MRC models are able to correctly process Semantics Altering Modifications (SAM): linguistically-motivated phenomena that alter the semantics of a sentence while preserving most of its lexical surface form. We present a method to automatically generate and align challenge sets featuring original and altered examples. We further propose a novel evaluation methodology to correctly assess the capability of MRC systems to process these examples independent of the data they were optimised on, by discounting for effects introduced by domain shift. In a large-scale empirical study, we apply the methodology in order to evaluate extractive MRC models with regard to their capability to correctly process SAM-enriched data. We comprehensively cover 12 different state-of-the-art neural architecture configurations and four training datasets and find that – despite their well-known remarkable performance – optimised models consistently struggle to correctly process semantically altered data.

P: 1 *After the kickoff Naomi Daniel...*
(B) Original: *curled in*
(I1) Modal negation: couldn't curl *in*
(I2) Adverbial Modification: almost *curled in*
(I3) Implicit Negation: was prevented from *curling in*
(I4) Explicit Negation: didn't succeed in *curling in*
(I5) Polarity Reversing: lacked the nerve to *curl in*
(I6) Negated Polarity Preserving: wouldn't find the opportunity to *curl in*
...a goal from 26 metres away following a decisive counter-attack. 2 *Then Amanda Collins added more insult to the injury when she slotted in from 23 metres after Linda Burger's soft clearance. [...]*

Q: *Who scored the farthest goal?*
A: Naomi Daniel **A with SAM:** Amanda Collins

Figure 1: Categories of SAM used in this paper with their implications on answering the given question. Modifying the original *"Baseline"* passage (B) by selecting any *"Intervention"* category (I1)−(I6), or removing the first sentence (*"Control"*) changes the correct answer from "Naomi Daniel" located in sentence 1 to "Amanda Collins" located in sentence 2.

Introduction

Machine Reading Comprehension (MRC), also commonly referred to as Question Answering, is defined as finding the answer to a natural language question given an accompanying textual context. State-of-the-art approaches build upon large transformer-based language models (Vaswani et al. 2017) that are optimised on large corpora in an unsupervised manner (Devlin et al. 2019) and further fine-tuned on large crowd-sourced task-specific MRC datasets (Rajpurkar et al. 2016; Yang et al. 2018; Trischler et al. 2017). They achieve remarkable performance, consistently outperforming human baselines on multiple reading comprehension and language understanding benchmarks (Lan et al. 2020; Raffel et al. 2019).

More recently, however, research on "data biases" in NLP suggests that these task-specific datasets exhibit various cues and spurious correlations between input and expected output (Gururangan et al. 2018; Poliak et al. 2018; Schlegel, Nenadic, and Batista-Navarro 2020). Indeed, data-driven approaches such as the state-of-the-art models (described above) that are optimised on these datasets learn to exploit these (Jia and Liang 2017; McCoy, Pavlick, and Linzen 2019), thus circumventing the actual requirement to perform comprehension and understanding.

For a (simplified) example, consider the question *"Who scored the farthest goal?"* illustrated in Figure 1. If a model is only exposed to examples where the accompanying passage contains sentences similar to "X scored a goal from Y metres" during training, a valid approximating decision based on this information could be similar to *"select the name next to the largest number and the word goal"* without actually fully reading the passage.

Alarmingly, conventional evaluation methodology where the dataset is split randomly into training and test data would not solve this issue. As both splits still stem from the same generative process (typically crowd-sourcing), the same types of cues are likely to exist in evaluation data, and a model can achieve high performance by relying on exploiting them. These and other problems suggest that the actual reading *comprehension* of state-of-the-art MRC models is potentially over-estimated.

In an attempt to present a more realistic estimate, we focus on the capability to correctly process *Semantic Altering Modifications* (SAM): minimal modifications to the passage that change its meaning and therefore the expected answer. On the one hand, it is important to know whether these modifications are processed correctly by MRC models, as they drastically change the meaning, for example if "X *almost* scored a goal from Y metres" then the goal effectively did not happen. On the other hand, distinguishing between original and modified examples is hard by relying on lexical cues only, as the modifications keep a similar lexical form. As a consequence, the simplified decision rule hypothesised above would not apply anymore.

Manually curating evaluation data to incorporate SAM is expensive and requires expert knowledge; also, the process must be repeated for each dataset resource (Gardner et al. 2020). Automatically changing existing MRC data is not a feasible strategy either, as the effects of a change on the meaning of the passage cannot be traced through the process and will still need to be verified manually. Instead, in this paper we propose a novel methodology to generate[1] SAM MRC challenge sets. We employ template-based natural language generation to maintain control over the presence of SAM and their effect onto the expected answer to a given question.

A problem that arises when evaluating models on challenge sets that were optimised on different training data, as it is the case in this paper, is the domain shift between training and evaluation data. For example, a model trained to retrieve answers from Wikipedia paragraphs might have never encountered a question involving comparing distances. In this case, wrong predictions on SAM examples cannot be attributed to the presence of SAM alone. To disentangle the effects introduced by the domain shift from the actual capability of correctly processing examples featuring SAM, we introduce a novel evaluation methodology with a corresponding metric, which we refer to as Domain Independent Consistency Evaluation or $DICE$. This allows us to precisely measure the capability of MRC models to process SAM of interest, and therefore, evaluate comprehension and language understanding that cannot be easily circumvented by relying on superficial cues. In a large-scale empirical study, we evaluate the performance of state-of-the-art transformer-based architectures optimised on multiple extractive MRC datasets. We find that while approaches based on larger language models tend to perform better, all investigated models struggle on the proposed challenge set, even after discounting for domain shift effects.

Semantics Altering Modifications

The task of (extractive) Machine Reading Comprehension is formalised as follows: given a question Q and a context P consisting of words $p_0 \dots p_n$, predict the start and end indices s, e (where $s < e$) that constitute the answer span $A = p_s \dots p_e$ in P. A Semantics Altering Modification (SAM) refers to the process of changing answer A to $A' \neq A$ by applying a modification to the accompanying context P. The rationale is to create a new *intervention* instance (Q, P', A') that is lexically similar to the original but has a different meaning and therefore a different expected answer for the same question. Predicting both A and A' given the question and the respective passages becomes a more challenging task than predicting A alone, since it requires correctly processing and distinguishing both examples. Due to their similarity, any simplifying heuristics inferred from training data are more likely to fail.

Furthermore, this intuitive description aligns with one of the prevalent linguistic definitions of modifiers as "an expression that combines with an unsaturated expression to form another unsaturated expression of the same [semantic] type" (McNally 2002). Particularly applicable to our scenario is the pragmatic or discourse-related view, specifically the distinction between modifiers that contribute to the content of a sentence with regard to a specific issue, and those that do not. In the context of MRC, the issue is whether the modification is relevant to finding the answer A to the question Q.

The linguistics literature is rich in reporting phenomena conforming with this definition. In this paper we explore negation (Morante and Daelemans 2012), (adverbial) restrictivity modification (Tenny 2000, Sec. 6), polarity reversing verbs and expressions (Karttunen 1971, 2012) and expressions of implicit negation (Iyeiri 2010). The categories with representative examples are shown in Figure 1 and labelled *I1-I6*. They reflect our intuitive definition as they involve relatively small edits to the original context, by inserting between one and four words that belong to the most frequent parts of speech classes of the English language, i.e. adverbials, modals, verbs and nouns. Note, however, that this selection is non-exhaustive. Other linguistic phenomena such as privative adjectives (Pavlick and Callison-Burch 2016), noun phrase modification (Stanovsky and Dagan 2016) or—if one were to expand the semantic types-based definition introduced above—corresponding discourse relations, such as Contrast or Negative Condition (Prasad et al. 2008), or morphological negation constitute further conceivable candidates. We leave it for future work to evaluate MRC on other types of SAM.

Domain Independent Consistency Evaluation

Consistency on "contrastive sets" (Gardner et al. 2020) was recently proposed as a metric to evaluate the comprehension of NLP models beyond simplifying decision rules. A contrastive set is—similar to SAM—a collection of similar data points that exhibit minimal differences such that the expected prediction (e.g. answer for MRC) differs for each member. Consistency is then defined as the ratio of contrastive sets where the model yielded a correct prediction for all its members to the total number of contrastive sets.

This notion requires that evaluation examples stem from the same generative process as the training data, making the

[1] Code at https://github.com/schlevik/sam

process of finding contrastive sets dataset-dependent. If the processes are different however, as it is the case with training set-independent challenge sets, this difference can be a confounding factor for wrong predictions, i.e. a model might produce a wrong prediction because the input differs too much from its training data and not solely because it was not capable of solving the investigated phenomenon. As we aim to establish an evaluation methodology independent of training data, we propose the following approach in order to rightfully attribute the capability to correctly process SAM even under domain shift.

We align each *baseline* MRC instance consisting of question, expected answer and context triple $B_i = (Q_i, A_i, P_i)$ with an *intervention* instance $I_i = (Q_i, A'_i, P'_i)$ s.t. $A'_i \neq A_i$. In practice, we achieve this by inserting a SAM in the sentence of P_i that contains A_i in order to obtain P'_i. We further align a *control* instance where we completely remove the sentence that was modified in P'_i, i.e. $C_i = (Q_i, A'_i, P''_i)$. Thus, an *aligned* instance consists of the triple (B_i, I_i, C_i) sharing the question Q. The answer A' is equivalent for both I_i and C_i. P, P' and P'' are shown in Figure 1 by selecting original (B) for P, any of the alternatives (I1) through (I6) for P' and completely removing the first sentence for P''.

The goal is to establish first, whether the model under evaluation "understood" the question and the accompanying context. Namely, if the model predicted A_i and A'_i correctly given Q_i, P_i and Q_i, P''_i, respectively, we conclude that the domain shift is not pivotal for the prediction performance of this particular instance, thus predicting the correct answer A'_i for I_i can be attributed to the model's capability to correctly process the SAM in P'_i. Conversely, if the model fails to predict A' we assume that the reason for this is is its incapability to process SAM (for this instance), regardless of the domain shift.

Initial experiments showed that models sometimes struggle to predict the exact span boundaries of the expected answer while retrieving the correct information in principle (e.g. predicting "from 26 metres" vs. the expected answer "26 metres"). Therefore we relax the usual *Exact Match* measure EM to establish the correctness of a prediction in the following way: $rEM_k(\hat{A}, A) = 1$ if a $\hat{A}$ has at most k words and A is a substring of $\hat{A}$ and 0 otherwise, where $\hat{A} = f_\theta(Q, P)$ is the answer prediction of an optimised MRC model f_θ given question Q and context P.

The metric $DICE$ is the number of examples the model predicted correctly in their baseline, intervention and control version divided by the number of those the model predicted correctly for the baseline and control version. This notion reflects the ratio of those modified instances that the model processed correctly regardless of the domain shift thus further evaluating the model's reading comprehension. Formally, for a challenge set $\mathcal{N} = \{\mathcal{B}, \mathcal{I}, \mathcal{C}\}$ consisting of N baseline, intervention and control examples, let

$$\begin{aligned} \mathcal{B}^+ &= \{i \mid rEM_k(f_\theta(Q_i, P_i), A_i) = 1\}_{i \in \{1 \dots N\}} \\ \mathcal{I}^+ &= \{i \mid rEM_k(f_\theta(Q_i, P'_i), A'_i) = 1\}_{i \in \{1 \dots N\}} \\ \mathcal{C}^+ &= \{i \mid rEM_k(f_\theta(Q_i, P''_i), A'_i) = 1\}_{i \in \{1 \dots N\}} \end{aligned} \tag{1}$$

denote the set of indices where an optimised model f_θ predicted a correct answer for baseline, intervention and control instances, respectively. Then

$$DICE(f_\theta) = \frac{|\mathcal{B}^+ \cap \mathcal{I}^+ \cap \mathcal{C}^+|}{|\mathcal{B}^+ \cap \mathcal{C}^+|} \in [0, 1]. \tag{2}$$

An inherent limitation of challenge sets is that they bear negative predictive power only (Feng, Wallace, and Boyd-Graber 2019). Translated to our methodology, this means that while low $DICE$ scores hint at the fact that models circumvent comprehension, high scores do not warrant the opposite, as a model still might learn to exploit some simple decision rules in cases not covered by the challenge set. In other words, while necessary, the capability of distinguishing and correctly processing SAM examples is not sufficient to evaluate reading comprehension.

A limitation specific to our approach is that it depends on a model's capability to perform under domain shift, at least to some extent. If a model performs poorly because of insufficient generalisation beyond training data or if the training data are too different from that of the challenge set, the sizes of $\mathcal{B}^+, \mathcal{I}^+$ and $\mathcal{C}^+$ decrease and therefore variations due to chance have a larger contribution to the final result. Concretely, we found that if the question is not formulated in natural language, as is the case for WIKIHOP (Welbl, Stenetorp, and Riedel 2018), or the context does not consist of coherent sentences (with SEARCHQA (Dunn et al. 2017) as an example) optimised models transfer poorly. Having a formalised notion of dataset similarity with respect to domain transfer for the task of MRC would help articulate the limitations and application scenarios of the proposed approach beyond pure empirical evidence.

SAM Challenge Set Generation

We now present the methodology for generating and modifying passages at scale. We aim to generate examples that require "reasoning skills" typically found in state-of-the-art MRC benchmarks (Sugawara et al. 2017; Schlegel et al. 2020). Specifically, we choose to generate football match reports as it intuitively allows us to formulate questions that involve simple (e.g. *"Who scored the first/last goal?"*) and more complex (e.g. *"When was the second/second to last goal scored?"*) linear retrieval capabilities, bridging and understanding the temporality of events (e.g. *"Who scored before/after X was fouled?"*) as well as ordering (e.g. *"What was the farthest/closest goal?"*) and comparing numbers and common properties (e.g. *"Who assisted the earlier goal, X or Y?"*). Answer types for these questions are named entities (e.g. players) or numeric event attributes (e.g. time or distance).

To generate passages and questions, we pursue a staged approach, common in Natural Language Generation (Gatt and Krahmer 2018). Note that we choose a purely symbolic approach over statistical approaches in order to maintain full control over the resulting questions and passages as well as the implications of their modification for the task of retrieving the expected answer. Our pipeline is exemplified in Figure 2 and consists of *(1)* content determination

Selected Content Plan

```
1 (Order (Distance (Modified Goal) 0)
2 (Order (Distance (Just Goal) 1)
Q (Argselect Max Goal Distance Actor)
```

Generated Events

```
1 {actor: p4, distance: 26, mod: I2 ...}
2 {actor: p2, distance: 23 ...}
A: p4 A': p2
```

Chosen Templates (Simplified)

```
1 %Con #Actor @SAM $V.Goal $PP.Distance...
2 #Actor %Con she $V.Score $PP.Distance...
Q Who scored the farthest goal ?
```

Generated Text

P: After the kickoff Naomi Daniel curled in a goal ...
P': After the kickoff Naomi Daniel almost curled in ...
P'': Then Amanda Collins added more insult to the ...

Figure 2: Stages of the generative process that lead to the question answer and context in Figure 1. The *Content Plan* describes the general constraints that the question type imposes on the *Events* (both sentences must describe goal events, first sentence must contain SAM, distance attribute must be larger in the modified sentence). Appropriate *Templates* are chosen randomly to realise the final Baseline P, Intervention P' and Control P'' version of the passage.

and structuring, followed by *(2)* content generation (as we generate the content from scratch) and finally *(3)* lexicalisation and linguistic realisation combining templates and a generative grammar.

Content planning and generation The output of this stage is a structured report of events that occurred during a fictitious match, describing event properties such as actions, actors and time stamps. Furthermore each report is paired with a corresponding question, an indication of which event is to be modified, and the corresponding answers. The report is generated semi-randomly, as the requirement to generate instances with a *meaningful* modification–i.e. actually changing the valid answer to the question–imposes constraints that depend on the type of the question. For example, for the retrieval type question *"Who scored the farthest goal?"* the report must contain at least two events of the type "goal" and the distance attribute associated with the event to be modified must be larger. We generate events of the type "goal", which are the target of the generated questions and modifications, and "other" that diverify the passages. Furthermore, to prevent repetition, we ensure that the order of the types of events is unique for each report-question combination in the final set of generated reports.

Realisation For the sake of simplicity, we choose to represent each event with a single sentence, although it is possible to omit this constraint by using sentence aggregation techniques and multi-sentence templates. Given a structured event description, we randomly select a "seed" template suitable for the event type. Seed templates consist of variables that are further substituted by event properties and expressions generated by the grammar. Thereby, we distinguish between context-free and context-sensitive substitutions. For example `$PP.Distance` in Figure 2 is substituted by a randomly generated prepositional phrase describing the distance (e.g. *"from 26 metes away"*) regardless of its position in the final passage. `%Con` in the same figure is substituted by an expression that connects to the previous sentence and depends on its content (e.g. *"After the kick-off"* can only appear in the first sentence of the paragraph). We collect the templates and construct the grammar by combining various manual and automated measures described in the Appendix in more detail. Similarly to the content generation, we ensure that the same template is not used more than once per report and the permutation of templates used to realise a report is unique in the final set of realised reports.

Data description The challenge set used in the experiments to evaluate MRC models trained on existing datasets consists of 4200 aligned baseline, intervention and control examples generated using the above process. The modified intervention examples contain between one and three SAM from the six categories described earlier. Using 25 "seed" templates and a generative grammar with 230 production rules, we can realise an arbitrary event in 4.8×10^6 lexically different ways; for a specific event the number is approx. 7.8×10^5 on average (the difference is due to context-sensitive parts of the grammar). When fine-tuning MRC models on our generated data, we separate the seed templates in two distinct sets, in order to ensure that the models do not perform well by just memorising the templates. These template sets are used to generate a training (12000 instances) and an evaluation (2400 instances) set with aligned baseline, intervention and control instances. All reports consist of six events and sentences, the average length of a realised passage is 174 words, averaging 10.8 distinct named entities and 6.9 numbers as answer candidates.

To estimate how realistic the generated MRC data is, we compare the paragraphs to the topically most similar MRC data: the NFL subset of the DROP dataset (Dua et al. 2019). We measure the following two metrics. *Lexical Similarity* is the estimated Jaccard similarity between two paragraphs, i.e. the ratio of overlapping words, with lower scores indicating higher (lexical) diversity. As a rough estimate of *Naturality*, we calculate the average of those sentence-level cohesion indices that are reported to correlate with human judgements of writing quality (Crossley, Kyle, and McNamara 2016; Crossley, Kyle, and Dascalu 2019). For exact definitions and further results please consult the Appendix. The results are shown below:

Data/Metric	Lex. Similarity ↓	Naturality ↑
SAM ($n = 200$)	0.22	0.65
NFL ($n = 188$)	0.16	0.68

While not quite reaching the reference data due to its template-based nature we conclude that the generated data is of sufficient quality for our purposes.

Experiments Setup

Broadly, we address the following question: *How well does MRC perform on Semantic Altering Modifications?*

In this study we focus our investigations on extractive MRC where the question is in natural language, the context is one or more coherent paragraphs and the answer is a single continuous span to be found within the context. To that end we sample state-of-the-art (neural) MRC architectures and datasets and perform a comparative evaluation. Scores of models with the same architecture optimised on different data allow to compare how much these data enable models to learn to process SAM, while comparing models with different architectures optimised on the same data hints to which extent these architectures are able to obtain this capability from data. Below we outline and motivate the choices of datasets and models used in the study. For further details on data preparation, model training and how we obtain predictions for evaluation, please consult the Appendix.

Datasets We select the following datasets in an attempt to comprehensively cover various flavours of state-of-the-art MRC consistent with our definition above.

- SQuAD (Rajpurkar et al. 2016) is a widely studied dataset where the human baseline is surpassed by the state of the art.
- HotpotQA (Yang et al. 2018) in the "distractor" setting requires information synthesis from multiple passages in the context connected by a common entity or its property.
- DROP (Dua et al. 2019) requires performing simple arithmetical tasks in order to predict the correct answer.
- NewsQA (Trischler et al. 2017) contains questions that were created without having access to the provided context. The context is a news article, different from the other datasets where contexts are Wikipedia excerpts.

Similar to Talmor and Berant (2019), we convert the datasets into the same format for comparability and to suit the task definition of extractive MRC. For HotpotQA we concatenate multiple passages into a single context and for DROP and NewsQA we only include examples where the question is answerable and the answer is a continuous span in the paragraph and refer to them as DROP' and NewsQA', respectively.

Models The respective best-performing models on these datasets are employing a large transformer-based language model with a task-specific network on top. Note that we do not use architectures that make dataset-specific assumptions (e.g. "Multi-hop" for HotpotQA) in order to maintain comparability of the architectures across datasets. Instead, we employ a linear layer as the most generic form of the task-specific network (Devlin et al. 2019). Following common practice, we concatenate the question and context, and optimise the parameters of the linear layer together with those of the language model to minimise the cross-entropy loss between the predicted and expected start and end indices of the answer span (and the answer sequence for the generative model).

We are interested in the effects of various improvements that were proposed for the original BERT transformer-based language model (Devlin et al. 2019). Concretely, we compare the effects of more training data and longer training for the language model (e.g. XLNet (Yang et al. 2019), RoBERTa (Liu et al. 2019)), parameter sharing between layers of the transformer (e.g. ALBERT (Lan et al. 2020)) and utilising a unifying sequence-to-sequence interface (e.g. BART (Lewis et al. 2020), T5 (Raffel et al. 2019)) and reformulating extractive MRC as text generation conditioned on the question and passage. We evaluate models of different sizes, ranging from `base` (`small` for T5) to `large` (and `xl` and `xxl` for ALBERT). They describe specific configurations of the transformer architecture, such as the number of the self-attention layers and attention heads and the dimensionality of hidden vectors. For an in-depth discussion please refer to Devlin et al. (2019) and the corresponding papers introducing the architectures. For comparison, we also include the non-transformer based BiDAF model (Seo et al. 2017). Finally, we train a model of the best performing architecture on a combination of all four datasets (`*-comb`) to investigate the effects of increasing training data diversity. For this, we sample and combine 22500 instances from all four datasets to obtain a training set that is similar in size to the others. The final selection consists of the models reported in Table 1.

Baselines We implement a `random` baseline that chooses an answer candidate from the pool of all named entities and numbers and an `informed` baseline that chooses randomly between all entities matching the expected answer type (e.g. person for "Who" questions). Finally, in order to investigate whether the proposed challenge can be solved in general, we train a `bert-base` model on 12000 aligned baseline and intervention instances, each. We refer to this baseline as `learned`. We train two more `bert-base` *partial baselines*, `masked-q` and `masked-p` on the same data where, respectively, the question and passage tokens (except for answer candidates) are replaced by out-of-vocabulary tokens. Our motivation for doing this is to estimate the proportion of the challenge set that can be solved due to regularities in the data generation method, regardless of the realised lexical form to provide more context to the performance of the learned baseline. Finally, we estimate the `human` SAM performance by crowd-sourcing the manual annotation of 100 intervention examples.

Results and Discussion

We present the main findings of our study here. For the obtained $DICE$ scores we report the error margin as a confidence interval at $\alpha = 0.05$ using asymptotic normal approximation. Any comparisons between two $DICE$ scores reported in this section are statistically significant ($p < 0.05$) as determined by performing the Fisher's exact test.

SAM is learnable. As expected, the learned baseline achieves high accuracy on our challenge set, with 81% and 79% correctly predicted instances for baseline and intervention examples, respectively, as seen in Table 2. The

Architecture	Average $DICE$	SQuAD EM/F1	SQuAD $DICE$	HotpotQA EM/F1	HotpotQA $DICE$	NewsQA' EM/F1	NewsQA' $DICE$	DROP' EM/F1	DROP' $DICE$
`bidaf`	11 ± 3	67.2/76.9	12 ± 4	44.6/57.9	4 ± 3	40.0/54.3	13 ± 5	50.8/56.8	18 ± 12
`bert-base`	13 ± 2	76.3/84.9	13 ± 3	50.7/64.9	17 ± 4	46.6/62.5	13 ± 3	50.5/58.2	10 ± 3
`bert-large`	15 ± 2	81.9/89.4	15 ± 3	54.4/68.7	14 ± 3	49.1/65.7	14 ± 4	62.2/68.7	16 ± 3
`roberta-base`	15 ± 2	82.4/89.9	8 ± 3	51.9/66.4	17 ± 4	50.8/66.9	14 ± 3	63.5/69.3	20 ± 3
`roberta-large`	18 ± 1	86.4/93.3	16 ± 3	58.6/72.9	21 ± 3	54.4/71.1	15 ± 3	77.3/82.8	20 ± 2
`albert-base`	14 ± 2	82.8/90.3	10 ± 3	55.4/69.7	17 ± 3	49.7/65.7	11 ± 3	60.7/67.0	18 ± 4
`albert-large`	16 ± 1	85.4/92.1	18 ± 3	59.4/73.7	12 ± 2	52.5/68.9	17 ± 3	69.3/75.1	18 ± 3
`albert-xl`	27 ± 2	87.1/93.5	19 ± 2	62.4/76.2	21 ± 3	54.2/70.4	29 ± 3	76.4/81.8	40 ± 3
`albert-xxl`	27 ± 1	88.2/94.4	29 ± 2	65.9/79.5	29 ± 3	54.3/71.0	25 ± 3	78.4/84.5	23 ± 2
`t5-small`	10 ± 1	76.8/85.8	13 ± 3	51.8/65.6	10 ± 3	47.3/63.3	8 ± 2	60.4/66.1	10 ± 3
`t5-base`	16 ± 1	82.4/90.6	16 ± 3	61.0/74.4	20 ± 3	52.4/68.8	14 ± 3	69.0/74.9	15 ± 2
`t5-large`	20 ± 1	86.3/93.1	21 ± 2	65.0/78.5	29 ± 3	53.4/70.0	16 ± 3	70.1/75.3	8 ± 2
`average`	19 ± 0	76.4/83.2	18 ± 1	53.1/65.9	20 ± 1	47.1/62.1	17 ± 1	61.5/67.0	20 ± 1
`albert-xl-comb`	20 ± 2	85.3/92.2		60.6/74.3		53.6/70.4		76.9/82.4	
`random`	5 ± 0								
`learned`	98 ± 0								

Table 1: $DICE$ and EM/F1 score on the corresponding development sets of the evaluated models. Average $DICE$ scores are micro-averaged as it better shows the average performance on processing SAM examples while EM/F1 are macro-averaged as it reflects the average performance on the datasets (although the difference between both averaging methods is small in practice).

results are in line with similar experiments on Recognising Textual Entailment (RTE) and sentiment analysis tasks which involved aligned counterfactual training examples (Kaushik, Hovy, and Lipton 2020). They suggest that neural networks are in fact capable of learning to recognise and correctly process examples with minimal yet meaningful differences such as SAM when explicitly optimised to do so. Some part of this performance is to be attributed to exploiting the regularity of the generation method rather than processing the realised text only, however, as the partial baselines perform better than the random baselines. This is further indicated by the slightly lower performance on the *control* set, where due to deletion the number of context sentences is different compared to the baseline and intervention sets.

We note that the `learned` model does not reach 100% EM score on this comparatively simple task, possibly due to the limited data diversity imposed by the templates. Using more templates and production rules and a bigger vocabulary will further enhance the diversity of the data.

Pre-trained models struggle. Table 1 reports the results of evaluating state-of-the-art MRC. Trained models struggle to succeed on our challenge set, with the best $DICE$ score of 40 achieved by `albert-xlarge` when trained on DROP'. There is a log-linear correlation between the effective size of the language model (established by counting the shared parameters separately for each update per optimisation step) and the SAM performance with Spearman's $r = 0.93$. Besides the model size, we do not find any contribution that leads to a considerable improvement in performance of *practical* significance. We note that simply increasing the data diversity appears not beneficial, as the score of `albert-xl-comb` that was optimised on the combination of all four datasets is lower than the average score of the corresponding `albert-xl` model. Humans appear to have little issues to find the correct answer, with $87\% \pm 7\%$ of the intervention examples solved correctly. This provides a lower bound for the human $DICE$ score (assuming all corresponding baseline and control examples are solved correctly).

The easiest SAM category to process is *I6: Explicit negation* with all optimised models scoring 26 ± 1.4 on average. Models struggle most with *I2: Adverbial Modification*, with an average $DICE$ score of 14 ± 1 (see Appendix for breakdown by SAM category). A possible reason is that this category contains *degree modifiers*, such as *"almost"*. While they alter the semantics in the same way as other categories for our purposes, generally they act as a more nuanced modification (compare e.g. *"almost"* with *"didn't"*). Finally, we note that the performance scales negatively with the number of SAM present in an example. The average $DICE$ score on instances with a single SAM is 23 ± 0.9, while on instances with the maximum of three SAM it is 16 ± 0.8 (and 19 ± 1.0 for two SAM). This is reasonable, as more SAM requires to process (and discard) more sentences, giving more opportunities to err.

We highlight that models optimised on HotpotQA and DROP' perform slightly better than models optimised on SQuAD and NewsQA' (on average 20% vs 18% and 17%, respectively). This suggests that exposing models to training data that require more complex (e.g. "multihop" and arithmetic) reasoning to deduce the answer, as opposed to simple answer retrieval based on predicate-argument structure (Schlegel et al. 2020), has a positive effect on distinguishing and correctly processing lexically similar yet semantically different instances.

Small improvements can be important. Our results indicate that small differences at the higher end of the performance spectrum can be of practical significance for the comprehension of challenging examples, such as SAM. Taking `albert` as an example, the relative performance improvements between the `base` and `xxlarge` model when (macro) averaged over the EM and F1 scores on the corresponding development sets are 15% and 13%, respectively, while the relative difference in average $DICE$ score is 93%. This is likely due to a share of "easy" examples in MRC evaluation data (Sugawara et al. 2018) that artificially bloat the (lower-end) performance scores to an extent.

Meaningful training examples are missing. One possible explanation for low scores could be that the models simply never encountered the expressions we use to modify the passages and thus fail to correctly process them. To investigate this claim we count the occurrences of the expressions of the worst performing category overall, *I2: Adverbial Modification*. The expressions appear in $5\%, 14\%, 5\%$ and 22% of the training passages of SQuAD, HotpotQA, DROP' and NewsQA' respectively, showing that models do encounter them during task-specific fine-tuning (not to mention during the language-model pre-training). It is more likely that the datasets lack examples where these expressions affect the search for the expected answer in a meaningful way (Schlegel et al. 2020). In fact, after sampling 400 passages and 647 corresponding questions (100 passages from each dataset) where the expression occurs within 100 characters of the expected answer, and manually annotating whether the modification would yield a different answer, we find only one such case which we can thus consider as a naturally occurring SAM. Worse yet, in 4% of the cases the expected answer ignores the presence of the SAM. This lends further credibility to the hypothesis that current MRC struggles at distinguishing examples with minimal yet meaningful changes such as SAM, if not explicitly incentivised during training. For more details on the annotation task consult the Appendix.

An analysis of models' errors suggests a similar conclusion: examining wrong intervention predictions for those cases where the answers for baseline and control were predicted correctly, we find that in $82\% \pm 1\%$ of those cases the models predict the baseline answer. Models thus tend to ignore SAM, rather than being "confused" by their presence (as if never encountered during training) and predicting a different incorrect answer.

Baseline	$\mathcal{B}$	$\mathcal{I}$	$\mathcal{C}$
`learned`	81 ± 2	79 ± 2	76 ± 2
`masked-q`	20 ± 2	28 ± 2	26 ± 1
`masked-p`	29 ± 1	5 ± 1	1 ± 1
`random`	6 ± 1	5 ± 1	8 ± 1
`informed`	14 ± 1	14 ± 1	26 ± 2
`human`	–	87 ± 7	–

Table 2: Percentage of correct predictions of the introduced baselines under the rEM_5 metric on aligned baseline $\mathcal{B}$, intervention $\mathcal{I}$ and control $\mathcal{C}$ sets.

Related work

Systematically modified MRC data can be obtained by rewriting questions using rule-based approaches (Ribeiro, Singh, and Guestrin 2018; Ribeiro, Guestrin, and Singh 2019) or appending distracting sentences, e.g. by paraphrasing the question (Jia and Liang 2017; Wang and Bansal 2018), or whole documents (Jiang and Bansal 2019) to the context. Adversarial approaches with the aim to "fool" the evaluated model, e.g. by applying context perturbations (Si et al. 2020) fall into this category as well. These approaches differ from ours, however, in that they aim to preserve the semantics of the modified example, therefore the expected answer is unchanged. But the findings are similar: models struggle to capture the semantic equivalence of examples after modification, and rely on lexical overlap between question and passage (Jia and Liang 2017). Our approach explores a complementary direction by generating semantically altered passages.

Using rule-based NLG techniques for controlled generation of MRC data was employed to obtain stories (Weston et al. 2015) that aim to evaluate the learnability of specific reasoning types, such as inductive reasoning or entity tracking. Further examples are `TextWorld` (Côté et al. 2018), an environment for text-based role playing games with a dataset where the task is to answer a question by interactively exploring the world (Yuan et al. 2019) and extending datasets with unanswerable questions (Nakanishi, Kobayashi, and Hayashi 2018). Similar to our approach, these generation methods rely on symbolic approaches to maintain control over the semantics of the data.

Beyond MRC, artificially constructed challenge sets were established with the aim to evaluate specific phenomena of interest, particularly for the RTE task. Challenge sets were proposed to investigate neural RTE models for their capabilities of logic reasoning (Richardson and Sabharwal 2019), lexical inference (Glockner, Shwartz, and Goldberg 2018), and understanding language compositionality (Nie, Wang, and Bansal 2019; Geiger et al. 2019).

Conclusion

We introduce a novel methodology for evaluating the reading comprehension of MRC models by observing their capability to distinguish and correctly process lexically similar yet semantically different input. We discuss lingustic phenomena that act as Semantic Altering Modifications and present a methodology to automatically generate and meaningfully modify MRC evaluation data. In an empirical study, we show that while the capability to process SAM correctly is learnable in principle, state-of-the-art MRC architectures optimised on various MRC training data struggle to do so. We conclude that one of the key reasons for this is the lack of challenging SAM examples in the corresponding datasets.

Future work will include the search for and evaluation on further linguistic phenomena suitable for the purpose of SAM, expanding the study from strictly extractive MRC to other formulations such as generative or multiple-choice MRC, and collecting a large-scale natural language MRC dataset featuring aligned SAM examples (e.g. via crowdsourcing) in order to investigate the impact on the robustness of neural models when exposed to those examples during training.

Acknowledgements

The authors would like to thank the anonymous reviewers for valuable suggestions as well as Alessio Sarullo for many fruitful discussions and help with the design of the experiments. The computation-heavy aspects of this paper were made possible due to access to the Computational Shared Facility at The University of Manchester, for which the authors are grateful.

References

Côté, M.-A.; Kádár, A.; Yuan, X.; Kybartas, B.; Barnes, T.; Fine, E.; Moore, J.; Tao, R. Y.; Hausknecht, M.; Asri, L. E.; Adada, M.; Tay, W.; and Trischler, A. 2018. TextWorld: A Learning Environment for Text-based Games. *arXiv preprint arXiv:1806.11532* .

Crossley, S. A.; Kyle, K.; and Dascalu, M. 2019. The Tool for the Automatic Analysis of Cohesion 2.0: Integrating semantic similarity and text overlap. *Behavior Research Methods* 51(1): 14–27.

Crossley, S. A.; Kyle, K.; and McNamara, D. S. 2016. The tool for the automatic analysis of text cohesion (TAACO): Automatic assessment of local, global, and text cohesion. *Behavior Research Methods* 48(4): 1227–1237.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 NAACL Conference*, 4171–4186.

Dua, D.; Wang, Y.; Dasigi, P.; Stanovsky, G.; Singh, S.; and Gardner, M. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *Proceedings of the 2019 NAACL Conference*, 2368–2378.

Dunn, M.; Sagun, L.; Higgins, M.; Guney, V. U.; Cirik, V.; and Cho, K. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *arXiv preprint arXiv 1704.05179* .

Feng, S.; Wallace, E.; and Boyd-Graber, J. 2019. Misleading Failures of Partial-input Baselines. In *Proceedings of the 57th ACL Annual Meeting*, 5533–5538.

Gardner, M.; Artzi, Y.; Basmova, V.; Berant, J.; Bogin, B.; Chen, S.; Dasigi, P.; Dua, D.; Elazar, Y.; Gottumukkala, A.; Gupta, N.; Hajishirzi, H.; Ilharco, G.; Khashabi, D.; Lin, K.; Liu, J.; Liu, N. F.; Mulcaire, P.; Ning, Q.; Singh, S.; Smith, N. A.; Subramanian, S.; Tsarfaty, R.; Wallace, E.; Zhang, A.; and Zhou, B. 2020. Evaluating NLP Models via Contrast Sets. *arXiv preprint arXiv 2004.02709* .

Gatt, A.; and Krahmer, E. 2018. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61: 65–170.

Geiger, A.; Cases, I.; Karttunen, L.; and Potts, C. 2019. Posing Fair Generalization Tasks for Natural Language Inference. In *Proceedings of the 2019 EMNLP-IJCNLP Conference*, 4484–4494.

Glockner, M.; Shwartz, V.; and Goldberg, Y. 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)*, 650–655.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; and Smith, N. A. 2018. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 NAACL Conference*, 107–112.

Iyeiri, Y. 2010. *Verbs of Implicit Negation and their Complements in the History of English*.

Jia, R.; and Liang, P. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 EMNLP Conference*, 2021–2031.

Jiang, Y.; and Bansal, M. 2019. Avoiding Reasoning Shortcuts: Adversarial Evaluation, Training, and Model Development for Multi-Hop QA. In *Proceedings of the 57th ACL Annual Meeting*, 2726–2736.

Karttunen, L. 1971. Implicative Verbs. *Language* 47(2): 340.

Karttunen, L. 2012. Simple and Phrasal Implicatives. Technical report.

Kaushik, D.; Hovy, E.; and Lipton, Z. C. 2020. Learning the Difference that Makes a Difference with Counterfactually-Augmented Data. In *Proceedings of ICLR*.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of ICLR*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. 7871–7880.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach .

McCoy, T.; Pavlick, E.; and Linzen, T. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th ACL Annual Meeting*, 3428–3448.

McNally, L. 2002. Modification. In *The Cambridge Handbook of Semantics*.

Morante, R.; and Daelemans, W. 2012. ConanDoyle-neg: Annotation of negation cues and their scope in Conan Doyle stories.

Nakanishi, M.; Kobayashi, T.; and Hayashi, Y. 2018. Answerable or Not: Devising a Dataset for Extending Machine

Reading Comprehension. In *Proceedings of the 27th COLING Conference*, 973–983.

Nie, Y.; Wang, Y.; and Bansal, M. 2019. Analyzing Compositionality-Sensitivity of NLI Models. In *Proceedings of the 33rd AAAI Conference*, 6867–6874.

Pavlick, E.; and Callison-Burch, C. 2016. So-Called Non-Subsective Adjectives. In *Proceedings of the 5th LEXSEM Conference*, 114–119.

Poliak, A.; Naradowsky, J.; Haldar, A.; Rudinger, R.; and Van Durme, B. 2018. Hypothesis Only Baselines in Natural Language Inference. In *Proceedings of the 7th LEXSEM Conference*, 180–191.

Prasad, R.; Dinesh, N.; Lee, A.; Miltsakaki, E.; Robaldo, L.; Joshi, A.; and Webber, B. 2008. The Penn Discourse TreeBank 2.0.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv preprint arXiv:1910.10683* .

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 55th ACL Annual Meeting*, 2383–2392.

Ribeiro, M. T.; Guestrin, C.; and Singh, S. 2019. Are Red Roses Red? Evaluating Consistency of Question-Answering Models. In *Proceedings of the 57th ACL Annual Meeting*, 6174–6184.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Semantically Equivalent Adversarial Rules for Debugging NLP models. In *Proceedings of the 56th ACL Annual Meeting)*, volume 1, 856–865.

Richardson, K.; and Sabharwal, A. 2019. What Does My QA Model Know? Devising Controlled Probes using Expert Knowledge. *arXiv preprint arXiv:1912.13337* .

Schlegel, V.; Nenadic, G.; and Batista-Navarro, R. 2020. Beyond Leaderboards: A survey of methods for revealing weaknesses in Natural Language Inference data and models. *arXiv preprint arXiv:2005.14709* .

Schlegel, V.; Valentino, M.; Freitas, A.; Nenadic, G.; and Batista-Navarro, R. 2020. A Framework for Evaluation of Machine Reading Comprehension Gold Standards. In *Proceedings of the 12th LREC*.

Seo, M. J.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2017. Bidirectional Attention Flow for Machine Comprehension. In *Proceedings of ICLR*.

Si, C.; Yang, Z.; Cui, Y.; Ma, W.; Liu, T.; and Wang, S. 2020. Benchmarking Robustness of Machine Reading Comprehension Models. *arXiv preprint arXiv 2004.14004* .

Stanovsky, G.; and Dagan, I. 2016. Annotating and Predicting Non-Restrictive Noun Phrase Modifications. In *Proceedings of the 54th ACL Annual Meeting*, 1256–1265.

Sugawara, S.; Inui, K.; Sekine, S.; and Aizawa, A. 2018. What Makes Reading Comprehension Questions Easier? In *Proceedings of the 2018 EMNLP Conference*, 4208–4219.

Sugawara, S.; Kido, Y.; Yokono, H.; and Aizawa, A. 2017. Evaluation Metrics for Machine Reading Comprehension: Prerequisite Skills and Readability. In *Proceedings of the 55th ACL Annual Meeting*, 806–817.

Talmor, A.; and Berant, J. 2019. MultiQA: An Empirical Investigation of Generalization and Transfer in Reading Comprehension. In *Proceedings of the 57th ACL Annual Meeting*, 4911–4921.

Tenny, C. 2000. Core events and adverbial modification. In Tenny, C.; and Pustejovsky, J., eds., *Workshop on events as grammatical objects from the combined perspectives of lexical semantics, logical semantics, and syntax*, 285–334.

Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; and Suleman, K. 2017. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 191–200.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30*, 5998–6008.

Wang, Y.; and Bansal, M. 2018. Robust Machine Comprehension Models via Adversarial Training. In *Proceedings of the 2018 NAACL Conference*, volume 2 (Short P, 575–581.

Welbl, J.; Stenetorp, P.; and Riedel, S. 2018. Constructing Datasets for Multi-hop Reading Comprehension Across Documents. *Transactions of the Association for Computational Linguistics* 6: 287–302.

Weston, J.; Bordes, A.; Chopra, S.; Rush, A. M.; van Merriënboer, B.; Joulin, A.; and Mikolov, T. 2015. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv preprint arXiv:1502.05698* .

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. Technical report.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 EMNLP Conference*, 2369–2380.

Yuan, X.; Côté, M.-A.; Fu, J.; Lin, Z.; Pal, C.; Bengio, Y.; and Trischler, A. 2019. Interactive Language Learning by Question Answering. In *Proceedings of the 2019 EMNLP-IJCNLP Conference*, 2796–2813.

Measure	SAM	NFL
positive correlation ↑		
m_1: Adj. sentence w2v similarity	0.58	0.67
negative correlation ↓		
m_2: Type-token ratio	0.72	0.66
m_3: Adj. sentence verb overlap	0.17	0.24
m_4: Pronoun-noun-ratio	0.07	0.05

Table 3: Detailed breakdown of measures used to obtain the final *Naturality* metric for the evaluation of the generated data.

Additional Details on the Challenge Set Generation

Algorithm 1 describes the challenge set generation in pseudo-code. The data used for generation were obtained by scraping football match reports from news and Wikipedia world cup finals websites[2][3]. They were automatically processed with the AllenNLP constituency parser [4] and manually arranged by their semantic content to form the generative grammar. Sentences were processed by the AllenNLP NER [5] and SRL [6] tools to substitute semantic roles of interest (e.g. player names, timestamps, verbs describing relevant actions) with variables, the output was manually verified and curated, resulting in the seed templates.

A detailed breakdown of the measures used to evaluate the quality of generated data is shown in Table 3. We measure the global and sentence-level indices that were reported to correlate with human judgements of writing quality by Crossley, Kyle, and McNamara (2016) and Crossley, Kyle, and Dascalu (2019). We use the associated tool `TAACO` to establish these measures. For the first three measures reported in Table 3 multiple indices are measured. For these measures we take the average of all available indices. We define the final *Naturality* metric as a combination of these measures. For simplicity we use the average:

$$Naturality = \frac{m_1 + (1 - m_2) + (1 - m_3) + (1 - m_4)}{4}$$

Note that we do not include intra-paragraph level measures, despite the fact that they are reported to correlate better with quality judgements. The reason for this is that both our generated passages and the reference DROP NFL data consist of a single paragraph.

Finally, Table 4 shows the effect of randomness on the metrics discussed in the paper. We measure the average result of 5 runs and report the standard deviation. As can be seen, for the data metrics of *Lexical Similarity* and *Naturality* as well as for the rEM_5 score, the impact of randomness is negligible. For the $DICE$ score the effect is more noticeable the lower the score, as discussed in the limitations section of the Section *Domain Independent Consistency Evaluation*.

[2]https://www.theguardian.com/tone/matchreports

[3]e.g. https://en.wikipedia.org/wiki/2006_FIFA_World_Cup_Final

[4]https://demo.allennlp.org/constituency-parsing

[5]https://demo.allennlp.org/named-entity-recognition

[6]https://demo.allennlp.org/semantic-role-labeling

Algorithm 1: `generate`

```
Data: question types T, question type event
      constraints C, number of examples per
      question type N, max. number of SAM per
      example S, number of events per report n,
      Question templates 𝒯_Q seed templates 𝒯_S,
      grammar 𝒢
ℬ, 𝒞, ℐ ← {}, {}, {}
foreach s ∈ 1...S do
    foreach i ∈ 1...|T| do
        plans ← generate all possible event plans for
          T_i with n events and s modifications s.t they
          satisfy C_i
        plans ← sample N_i w/o replacement from
          plans
        reports ← generate structured reports from
          each plan ∈ plans
        permutations ← {}
        foreach r ∈ reports do
            permutation ← choose permutation of n
              from 𝒯_S respecting order of events of r
              and not in permutations
            add permutation to permutations
            P ← ϵ
            P′ ← ϵ
            foreach template t ∈ permutation do
                foreach symbol v ∈ t do
                    l = realise v using 𝒢 with v as
                      start symbol
                    append l to P′
                    if v is not SAM then
                        append l to P
                    end
                end
            end
            P″ ← copy P′ and remove modified
              sentences
            Q, A, A′ ← realise question and answers
              given P, P′ and r
            add (Q, A, P) to ℬ, (Q, A′, P′) to ℐ and
              (Q, A′, P″) to 𝒞
        end
    end
end
return ℬ, 𝒞, ℐ
```

Data preparation and training details

For HotpotQA we concatenate multiple passages and their titles and prepend them with the `[text]` and `[title]` tokens, respectively. We further prepend the input with `yes` and `no` tokens, as some examples require this as answer. Following Devlin et al. (2019), we represent the question and context as a single sequence instance, separated by the

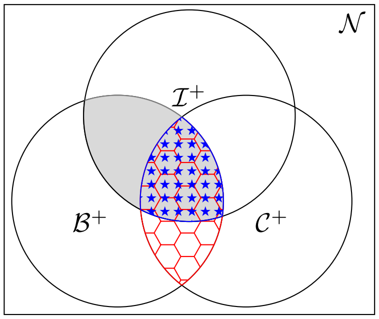

Figure 3: With circles $\mathcal{B}^+, \mathcal{I}^+$ and $\mathcal{C}^+$ from Eq. 1 representing instances that were answered correctly in their baseline, intervention respectively control version, respectively, $DICE$ is the proportion of the star covered area to the area covered by hexagons. Consistency, when defining contrastive sets as $\{B_i, I_i\}_{i \in \{1 \dots N\}}$ (Gardner et al. 2020) is the proportion of the grey area to the area of the entire square.

`[SEP]` token. The maximal size of this input sequence is 384 (subword) tokens, passages exceeding this length are split in multiple sequences each prepended by the question. The stride (overlap between subsequent splits of a passage) is 128 tokens. Sequences shorter than 384 are padded to maximal length. The (softmax over the) task specific layer outputs the probability distributions of tokens being the start or end index, respectively. The training objective is to minimise the cross-entropy loss between the logits of the final layer and the correct start and end indices. During inference we select the start and end index pair (s, e) with the maximum score $s + e$ with $s > e$, $e - s <=$ `max_answer_length` and neither s nor e being indices of the `SEP` or `PAD` tokens. In case the input was split, we select the pair with the highest score across all corresponding inputs.

For the generative `t5` encoder-decoder model we use a similar approach. We concatenate the question and context into a single sequence of maximal length of 512 tokens for SQuAD and DROP', 1024 for NewsQA' and 2048 for HotpotQA. We use the encoder to encode this sequence and use its hidden state as the initial representation for the decoder to generate a sequence of tokens as the answer. The training objective is to minimise the cross-entropy loss between the predicted tokens and the vocabulary indices of the expected answer. Similarly, during inference we iteratively generate a sequence of a maximum of `max_answer_length` using the hidden state of the encoder after encoding the question and passage for the first token and the hidden state of the decoder thereafter.

Metric	**Mean**	**Std. dev.**
Diversity	0.1642	0
Naturality	0.66	0.003
rEM_5 on $\mathcal{B}$ ($\lvert\mathcal{B}\rvert = 4200$) of `bert-base` optimised on SQuAD	0.19	0.006
$DICE$ score of `bert-base` optimised on SQuAD	0.16	0.023

Table 4: Various measures used in the paper averaged over 5 runs where the challenge set was generated from different random seeds. Mean and standard deviation are reported.

We implement the training and inference in PyTorch 1.6.0[7]. We use the pre-trained language models available in the `transformers`[8] library. We train the `bert`, `roberta` and `albert` models on 4 Nvidia V100 GPUs with 16 GB of RAM using data parallelism for the training on SQuAD and distributed training for the other datasets.

The `t5` models were trained using a single Nvidia V100 GPU, except when training the `t5-large` model, we employed 4-way Model Parallelism (i.e. spreading different parameters across multiple GPUs) for HotpotQA and 2-way model parallelism for `NewsQA'`, because of GPU memory constraints.

We fix the random seed to maintain deterministic behaviour and the hyper-parameters used for training are

- **Batch size:** employing (distributed) data parallelism, mixed precision and gradient accumulation we use a batch-size of 2^{10}. Note that due to this combination, the reported results on the development sets are slightly lower than what is reported in the literature (e.g. up to 3 points lower F1 score for `bert-base` and less than 1 point lower F1 score for `albert-xxlarge`). Given the training speed-up we obtain and the somewhat orthogonal goal of our study we deem this performance loss manageable.
- **Learning Rate:** We set learning rate to 5^{-5}, as it was reported to work best for the transformer training. For `t5` we found the learning rate of 0.001 used in the original experiments to work best. In both cases, we found that linearly decaying the learning rate to 0 over the course of the training is beneficial. We use the ADAM optimiser with the default parameters $\epsilon = 1 \times 10^{-8}, \beta_1 = 0.99$ and $\beta_2 = 0.999$.
- **Train Epochs:** We train on SQuAD for 3 training epochs, for 2 epochs on HotpotQA for 4 epochs on NewsQA' and for 12 epochs on DROP'. This is to ensure that the models across the different datasets have a roughly equal computational budget as the datasets vary in size and context length.
- **Maximal answer length**: We use `max_answer_length=30` when obtaining predictions on the original datasets and `max_answer_length=10` for predictions on the challenge set, because the challenge set answers are generally shorter.

The BiDAF model was trained using the AllenNLP framework using their released configuration file[9].

To replicate our experiments, please follow the "readme" in the supplemented source code.

[7] https://pytorch.org
[8] https://github.com/huggingface/transformers
[9] https://raw.githubusercontent.com/allenai/allennlp-models/v1.0.0/training_config/rc/bidaf.jsonnet

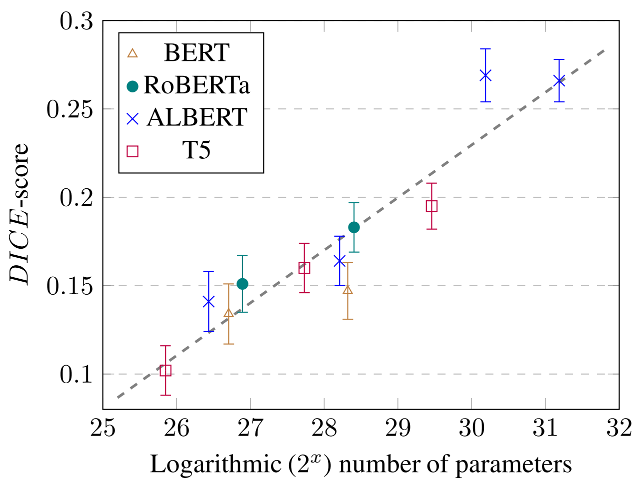

Figure 4: $DICE$ score averaged for all datasets by effective model size, shared parameters are counted separetely. The dashed line represents a fitted linear regression showing the correlation between the (logarithmic) model size and the score.

Additional Experiments and Results

We evaluate the impact of k in the rEM_k formulation and report the results in Table 6. As the parameter seems to have little impact on the final score, we settle for $k = 5$ for all relevant experiments in the paper.

We experiment with training a `bert-base` model on baseline examples without SAM and evaluate it with $DICE$ on $\mathcal{B}, \mathcal{I}, \mathcal{C}$, respectively. We use the same generated data that was used for other baselines, with 12000 baseline training instances and 2400 baseline, intervention and control evaluation examples generated from two distinct template sets. We obtain a $DICE$ score of 2 ± 1 as a result. Intuitively, this is expected and similar to what Kaushik, Hovy, and Lipton (2020) report: as the model only encounters non-altered examples during training, it is not incentivised to learn to process SAM correctly, which is required to obtain high $DICE$ scores. We take it as evidence towards the fact that the $DICE$ formulation effectively evaluates SAM performance, because models optimised on in-domain data that verifiably lacks SAM examples still obtain low $DICE$ scores.

We visualise the correlation between the effective size of the transformer language model and the $DICE$ score in Figure 4. A detailed breakdown of performance by SAM category is shown in Table 5.

SAM Category	**Average $DICE$**
Modal negation	20 ± 1.3
Adverbial Modification	14 ± 1.1
Implicit Negation	20 ± 1.4
Explicit Negation	26 ± 1.4
Polarity Reversing	18 ± 1.3
Negated Polarity Preserving	23 ± 1.5

Table 5: Average performance on the challenge set, by SAM category.

k	rEM_k	k	rEM_k	k	rEM_k
1	19.1 ± 1	2	19.2 ± 1	3	19.5 ± 1
4	19.5 ± 1	5	19.5 ± 1		

Table 6: Impact of k in the formulation of rEM_k when evaluating a SQuAD-optimised `bert-base` model on $\mathcal{B}$.

Label	**Occurences**	**in %**
Nonaltering	522	80.68%
Altering	1	0.15 %
QuestionAnswer	61	9.43 %
BadExample	36	5.56 %
Opposite	27	4.17 %

Table 7: Detailed results of the annotation task.

For the manual annotation of training examples from the four datasets used in the study, we sampled 100 passages from each dataset where the expressions "almost", "nearly" and "all but" from the category *I2: Adverbial Modification* were found in the passage within 100 characters of the expected answer. Because the datasets (except HOTPOTQA) feature more than one question per passage, the overall number of questions for annotation is 647. The examples were annotated by the first author with the following labels:

- *Nonaltering* if removing the matched expression does not change the expected answer
- *Altering* if removing the matched expression *does* change the expected answer
- *QuestionAnswer* if the matched expression is part of the question or expected answer
- *BadExample* if the match was erroneous (e.g. if "all but" was used in the sense "every single one except" rather than "almost")
- *Opposite* if the expected answer ignores the expression (e.g. when for the Question "What fraction did [...]" and the answer sentence "Nearly half of [...]" the expected answer is "half" rather than "nearly half".

20% of the passages were co-annotated by the last author, the inter-annotator agreement as per Cohen's κ is 0.82. The disagreements concerned the category *BadExample* and *Nonaltering*, with some of the labels being assigned differently by the two annotators. Besides these two categories the agreement score is in fact 1.0.

The human performance on SAM was established by means of crowd-sourcing a sample of 100 examples taken from the intervention set $\mathcal{I}$ used to evaluate the learned baselines. The annotators were asked to select a span that represents the correct answer given a question and a paragraph, without priming them on the nature of the task, (i.e. the fact that they contain SAM). Specifically, the annotators were presented the following instruction: *You will be presented a football match report and a question concerning* *said report. Read the paragraph and question* carefully *and then highlight the passage that in your opinion answers the question. All questions are answerable unambiguously by a single continuous span that is present in text. Background knowledge is not required (such as the distance of a penalty kick in football). When annotating, please annotate the full name (e.g. What Zit Tooya) and any numbers together with the belonging units (e.g. 1 pint), where appropriate. If you think an example is unanswerable, please flag it as an error and let me know.* This instruction was followed by an example annotation and a technical explanation of the annotation tool.

In order to better represent human performance rather than collect high-quality annotations, we collected a single annotation per example.

A manual analysis of erroneous annotations revealed that in 38% of the cases, SAM were ignored. The remaining errors (62%) were due to other reasons, such as confusing the interrogative pronoun (e.g. by answering with a span denoting a distance to a "When" question), or selecting otherwise wrong answers that are not related to presence of absence of SAM.