TORQUE: A Reading Comprehension Dataset of Temporal Ordering Questions

Qiang Ning[◇] **Hao Wu**[♣] **Rujun Han**[♠] **Nanyun Peng**[♠] **Matt Gardner**[◇] **Dan Roth**[♡]
[◇]Allen Institute for AI [♣]Hooray Data Co., Ltd
[♠]University of Southern California [♡]University of Pennsylvania
{qiangn,mattg}@allenai.org, haowu@hooray.ai
{rujunhan,npeng}@isi.edu, danroth@seas.upenn.edu

Abstract

A critical part of reading is being able to understand the temporal relationships between events described in a passage of text, even when those relationships are not explicitly stated. However, current machine reading comprehension benchmarks have practically no questions that test temporal phenomena, so systems trained on these benchmarks have no capacity to answer questions such as "what happened *before/after* [some event]?" We introduce TORQUE, a new English reading comprehension benchmark built on 3.2k news snippets with 21k human-generated questions querying temporal relationships. Results show that RoBERTa-large achieves an exact-match score of 51% on the test set of TORQUE, about 30% behind human performance.[1]

1 Introduction

Time is important for understanding events and stories described in natural language text such as news articles, social media, financial reports, and electronic health records (Verhagen et al., 2007, 2010; UzZaman et al., 2013; Minard et al., 2015; Bethard et al., 2016, 2017; Laparra et al., 2018). For instance, *"he won the championship yesterday"* is different from *"he will win the championship tomorrow"*: he may be celebrating if he has already won it, while if he has not, he is probably still preparing for the game tomorrow.

The exact time of an event is often implicit in text. For instance, if we read that a woman is *"expecting the birth of her first child"*, we know that the birth is in the future, while if she is *"mourning the death of her mother"*, the death is in the past. These relationships between an event and a time point (e.g., *"won the championship yesterday"*) or between two events (e.g., "expecting" is *before* "birth" and "mourning" is *after* "death") are called *temporal relations* (Pustejovsky et al., 2003).

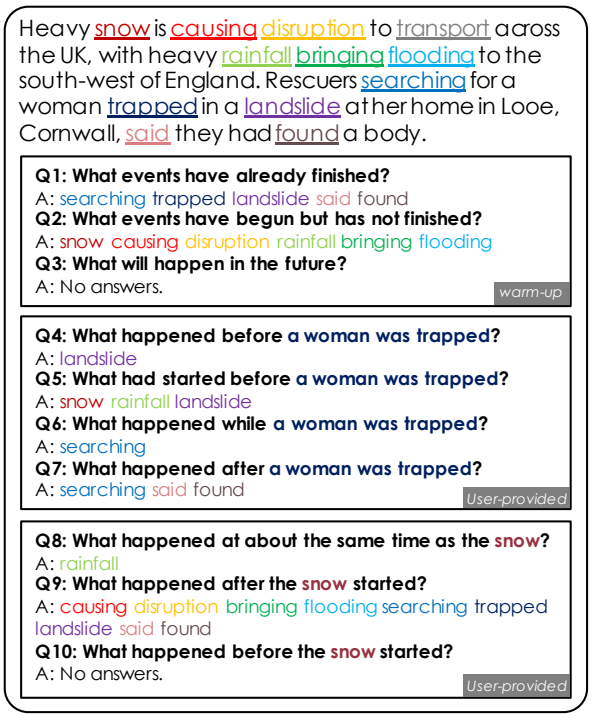

Figure 1: Example annotation of TORQUE. Events are highlighted in color and contrast questions are grouped.

This work studies reading comprehension for temporal relations, i.e., given a piece of text, a computer needs to answer temporal relation questions (Fig. 1). Reading comprehension is a natural format for studying temporal phenomena, as the flexibility of natural language annotations allows for capturing relationships that were not possible in previous formalism-based works. However, temporal phenomena are studied very little in reading comprehension (Rajpurkar et al., 2016, 2018; Dua et al., 2019; Dasigi et al., 2019; Lin et al., 2019), and existing systems are hence brittle when handling questions in TORQUE (Table 1).

Reading comprehension for temporal relationships has the following challenges. First, reading

[1] https://allennlp.org/torque.html

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1158–1172,
November 16–20, 2020.

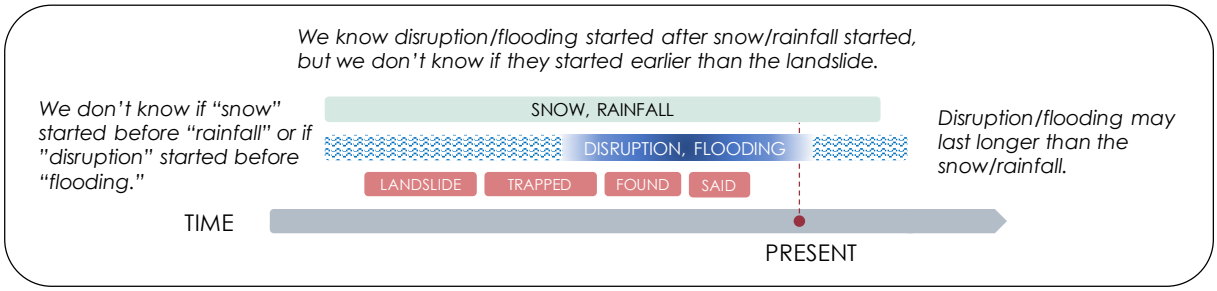

Figure 2: Timeline of the passage in Fig. 1.

Question	BERT (trained on SQuAD)	BERT (trained on SQuAD 2.0)
What happened before a woman was trapped?	a landslide	a landslide
What happened after a woman was trapped?	they had found a body	a landslide
What happened while a woman was trapped?	a landslide	a landslide
What happened before the snow started?	landslide	heavy rainfall . . . landslide
What happened after the snow started?	flooding to . . . England	heavy rainfall . . . England
What happened during the snow?	a landslide	landslide
What happened before the rescuers found a body?	a landslide	a landslide
What happened after the rescuers found a body?	Rescuers searching . . . Cornwall	landslide
What happened during the rescue?	a landslide	they had found a body

BERT (SQuAD): https://cogcomp.seas.upenn.edu/page/demo_view/QuASE
BERT (SQuAD 2.0): https://www.pragnakalp.com/demos/BERT-NLP-QnA-Demo/

Table 1: Example system outputs. The correct answers can be seen from the timeline depicted in Fig. 2.

comprehension works rarely require *event* understanding. For the example in Fig. 1, SQuAD (Rajpurkar et al., 2016) and most datasets largely only require an understanding of predicates and arguments, and would ask questions like *"what was a woman trapped in?"* But a temporal relation question would be *"what started before a woman was trapped?"* To answer it, the system needs to identify events (e.g., LANDSLIDE is an event and "body" is not), the time of these events (e.g., LANDSLIDE is a correct answer, while SAID is not because of the time when the two events happen), and look at the entire passage rather than the local predicate-argument structures within a sentence (e.g., SNOW and RAINFALL are correct answers to the question above).

Second, there are many events in a typical passage of text, so temporal relation questions typically query more than one relationship at the same time. This means that a question can have multiple answers (e.g., *"what happened after the landslide?"*), or no answers, because the question may be beyond the time scope (e.g., *"what happened before the snow started?"*).

Third, temporal relations queried by natural language questions are often sensitive to a few key words such as *before*, *after*, and *start*. Those questions can easily be changed to make contrasting questions with dramatically different answers. Models that are not sensitive to these small changes in question words will perform poorly on this task, as shown in Table 1.

In this paper, we present TORQUE, the first reading comprehension benchmark that targets these challenges. We trained crowd workers to label events in text, and to write and answer questions that query temporal relationships between these events. We also had workers write questions with contrasting changes to the temporal keywords, to give a comprehensive test of a machine's temporal reasoning ability and minimize the effect of any data collection artifacts (Gardner et al., 2020). We annotated 3.2k text snippets randomly selected from the TempEval3 dataset (UzZaman et al., 2013). In total, TORQUE has 25k events and 21k user-generated and fully answered temporal relation questions. 20% of TORQUE was further validated by additional crowd workers to be used as test data. Results show that RoBERTa-large (Liu et al., 2019) achieves 51% in exact-match on TORQUE after fine-tuning, about 30% behind human performance, indicating that more investigation is needed to better solve this problem.

2 Definitions

2.1 Events

As temporal relations are relationships between events, we first define *events*. Generally speak-

Events in different modes

The lion had a large meal and slept for 24 hours.

[Negated] The lion didn't sleep after having a large meal.

[Uncertain] The lion may have had a large meal before sleeping.

[Hypothetical] If the lion has a large meal, it will sleep for 24 hours.

[Repetitive] The lion used to sleep for 24 hours after having large meals.

[Generic] After having a large meal, lions may sleep longer.

Figure 3: Various modes of events that prior work needed to categorize. Section 3 shows that they can be handled naturally without explicit categorization.

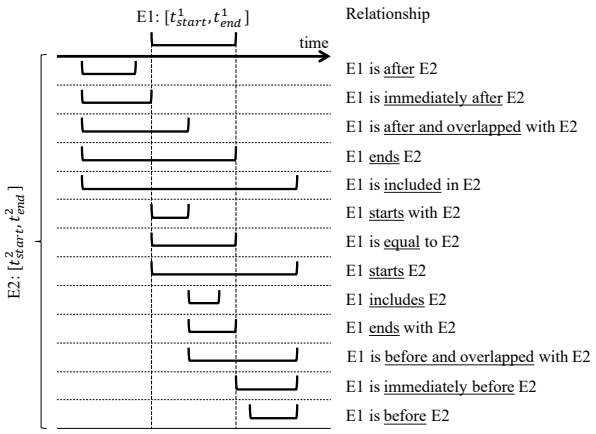

Figure 4: Thirteen relations between two time intervals $[t^1_{start}, t^1_{end}]$ and $[t^2_{start}, t^2_{end}]$.

ing, an event involves a predicate and its arguments (ACE, 2005; Mitamura et al., 2015). When studying *time*, events were defined as actions/states triggered by verbs, adjectives, and nominals (Pustejovsky et al., 2003). Later works on event and time have largely followed this definition, e.g., TempEval (Verhagen et al., 2007), TimeBank-Dense (Chambers et al., 2014), RED (O'Gorman et al., 2016), and MATRES (Ning et al., 2018b).

This work follows this line of event definition and uses *event* and *event trigger* interchangeably. We define an event to be either a verb or a noun (e.g., TRAPPED and LANDSLIDE in Fig. 1). Specifically, in copular constructions, we choose to label the verb as the event, instead of an adjective or preposition. This allows us to give a consistent treatment of *"she was on the east coast yesterday"* and *"she was happy"*, which we can easily teach to crowd workers. Note that from the perspective of data collection, labeling the copula does not lose information as one can always do post-processing using dependency parsing or semantic role labeling to recover the connection between "was" and "happy."

Note that events expressed in text are not always factual. They can be negated, uncertain, hypothetical, or have other associated modalities (see Fig. 3). Prior work dealing with events often tried to categorize and label these various aspects because they were crucial for determining temporal relations. Sometimes certain categories were even dropped due to annotation difficulties (Pustejovsky et al., 2003; O'Gorman et al., 2016; Ning et al., 2018b). In this work, we simply have people label all events, irrespective of their modality, and use natural language to describe relations between them, as discussed in Sec. 3.

2.2 Temporal Relations

Temporal relations describe the relationship between two events with respect to time, or between one event and a fixed time point (e.g., *yesterday*).[2] We can use a triplet, (A, r, B), to represent this relationship, where A and B are events or time points, and r is a temporal relation. For example, the first sentence in Fig. 3 expresses a temporal relation (HAD, *happened before*, SLEPT).

In previous works, every event is assumed to be associated with a time interval $[t_{start}, t_{end}]$. When comparing two events, there are 13 possible relation labels (see Fig. 4) (Allen, 1984). However, there are still many relations that cannot be expressed because the assumption that every event has a time interval is inaccurate: The time scope of an event may be fuzzy, an event can have a non-factual modality, or events can be repetitive and invoke multiple intervals (see Fig. 5). To better handle these phenomena, we move away from the fixed set of relations used in prior work and instead use natural language to annotate the relationships between events, as described in the next section.

3 Natural Language Annotation of Temporal Relations

Motivated by recent works (He et al., 2015; Michael et al., 2017; Levy et al., 2017; Gardner et al., 2019b), we propose using natural language question answering as an annotation for-

[2] We could also include relationships between two fixed time points (e.g., compare 2011-03-24 with 2011-04-05), but these are mostly trivial, so we do not discuss them further.

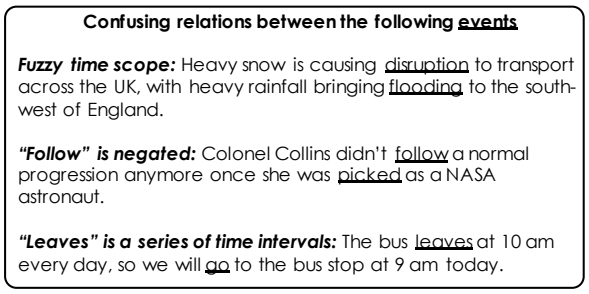

Figure 5: It is confusing to label these relations using a fixed set of relations: they are not simply *before* or *after*, but they can be fuzzy, can have modalities as events, and/or need multiple time intervals to represent.

mat for temporal relations. Recalling that we denote a temporal relation between two events as (A, r, B), we use $(?, r, B)$ to denote a temporal relation question. We instantiate these temporal relation questions using natural language. For instance, (?, *happened before*, SLEPT) means *"what happened before a lion slept?"* We then expect as an answer the set of all events A in the passage such that (A, r, B) holds, assuming for any deictic expression A or B the time point when the passage was written, and assuming that the passage is true.

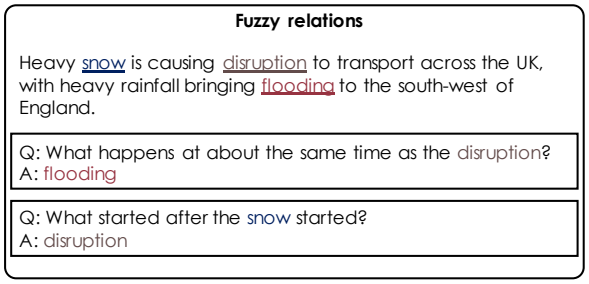

Figure 6: Fuzzy relations that used to be difficult to represent using a predefined label set can be captured naturally in a reading comprehension task.

3.1 Advantages

Studying temporal relations as a reading comprehension task gives us the flexibility to handle many of the aforementioned difficulties. First, fuzzy relations can be described by natural language questions (after all, the relations are expressed in natural language in the first place). In Fig. 6, DISRUPTION and FLOODING happened at about the same time, but we do not know for sure which one is earlier, so we have to choose *vague*. Similarly for SNOW and DISRUPTION, we do not know which one ends earlier and have to choose *vague* again. In contrast, the question-answer (QA) pairs

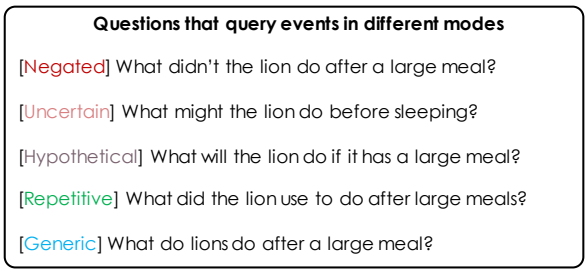

Figure 7: Events in different modes can be distinguished using natural language questions.

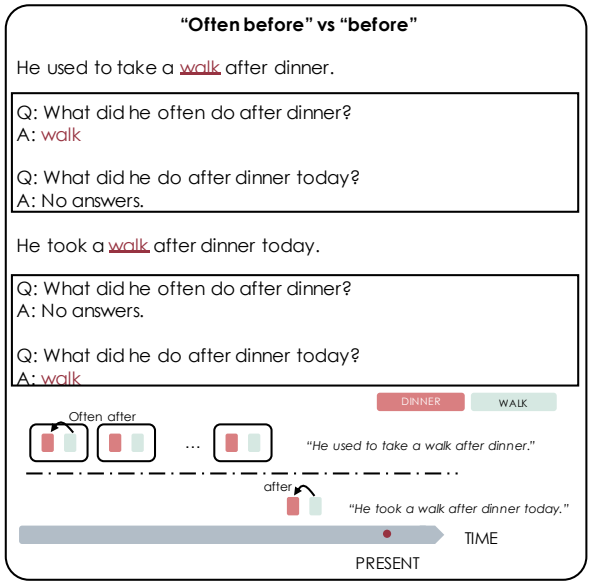

Figure 8: A repetitive event needs multiple time intervals and conveys very different semantics.

in Fig. 6 can naturally capture these fuzzy relations.

Second, natural language questions can conveniently incorporate different modes of events. Figure 7 shows how to accurately query the relation between *"having a meal"* and *"sleeping"* in different modes (original sentences can be found in Fig. 3). In contrast, if we could only choose one label, we must choose *before* for all these relations, although these relations are actually different. For instance, a repetitive event may be a series of intervals rather than a single one, and *often before* is very different from *before* (Fig. 8).

Third, a major issue that prior works wanted to address was deciding when two events should have a relation (Cassidy et al., 2014; Mostafazadeh et al., 2016; O'Gorman et al., 2016; Ning et al., 2018b). To avoid asking for relations that do not exist, prior works needed to explicitly annotate certain properties of events as a preprocessing step, but it still remains difficult to have a the-

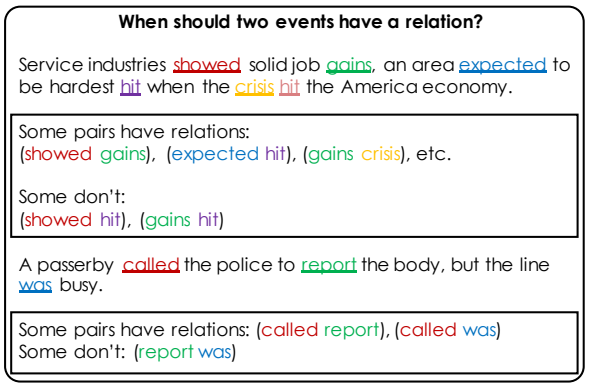

Figure 9: It remains unclear how to determine if two events should have a temporal relation.

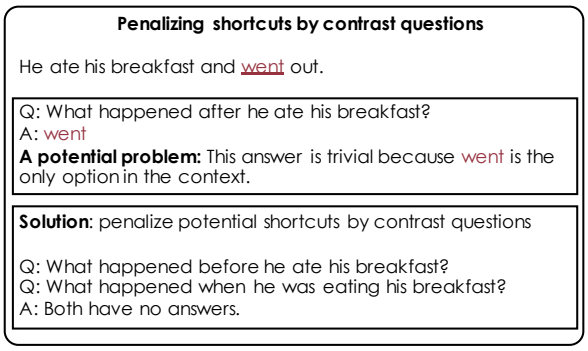

Figure 10: Penalize potential shortcuts by providing contrast questions.

ory explaining, for instance, why hit can compare to expected and crisis, but not to gains. Interestingly, when we annotate temporal relations in natural language, the annotator naturally avoids event pairs that do not have relations. For instance, for the sentences in Fig. 9, one will not ask questions like *"what happened after the service industries are hardest hit?"* or *"what happened after a passerby reported the body?"* Instead, natural questions will be *"what was expected to happen when the crisis hit America?"* and *"what was supposed to happen after a passerby called the police?"* The format of natural language questions bypasses the need for explicit annotation of properties of events or other theories.

While using QA as the format gives us many benefits in describing fuzzy relations and incorporating various temporal phenomena, we want to note that it may also lead to potential difficulties in transferring the knowledge to downstream tasks that are not in a QA format, and some special treatment in modeling may be needed (e.g., He et al. (2020)). This paper focuses on constructing this QA dataset covering new phenomena, and the problem of successful transfer learning is beyond our scope here.

3.2 Penalize Shortcuts by Contrast Sets

An important problem in building datasets is to avoid trivial solutions (Gardner et al., 2019a). As Fig. 10 shows, there are two events ATE and WENT in the text. Since ATE is already mentioned in the question, the answer of WENT seems a trivial option without the need to understand the underlying relationship. To address this issue, we create *contrast questions* which slightly modify the original questions, but dramatically change the answers, so that shortcuts are penalized. Specifically, for an existing question $(?, r, B)$ (e.g., *"what happened after he ate his breakfast?"*), one should keep using B and change r (e.g., *"what happened **before/shortly after/**... he ate his breakfast?"*), or modify it to ask about the start/end time (e.g., *"what happened after he **started** eating his breakfast?"* or *"what would **finish** after he ate his breakfast?"*). We also instructed workers to make sure that the answers to the new question are different from the original one to avoid trivial modifications (e.g., changing *"what happened"* to *"what occurred"*).

4 Data Collection

We used Amazon Mechanical Turk to build TORQUE. Following prior work, we focus on passages that consist of two contiguous sentences, as this is sufficient to capture the vast majority of non-trivial temporal relations (Ning et al., 2017). We took all the articles used in the TempEval3 (TE3) workshop (2.8k articles) (UzZaman et al., 2013) and created a pool of 26k two-sentence passages. Given a random passage from this pool, the annotation process for crowd workers was:

1. Label all the events
2. Repeatedly do the following[3]
   (a) Ask a temporal relation question and point out all the answers from the list of events
   (b) Modify the temporal relation to create one or more new questions and answer them

The annotation guidelines[4] and interface[5] are public. In the following sections, we further discuss issues of quality control and crowdsourcing cost.

[3] The stopping criterion is discussed in Sec. 4.2.
[4] https://qatmr-qualification.github.io/
[5] https://qatmr.github.io/

4.1 Quality Control

We used three quality control strategies: qualification, pilot, and validation.

Qualification We designed a separate qualification task where crowd workers were trained and tested on 3 capabilities: labeling events, asking temporal relation questions, and question-answering. They were tested on problems randomly selected from a pool we designed. Crowd workers were considered level-1 qualified if they could pass the test within 3 attempts. In practice, about 1 out of 3 workers passed our qualification.

Pilot We then asked level-1 crowd workers to do a small amount of the real task. We manually checked the annotations and gave feedback to them. Those who passed this inspection were called level-2 workers, and only they could work on the large-scale real task. Roughly 1 out of 3 pilot submissions received a level-2 qualification. In the end, there were 63 level-2 annotators, and 60 of them actually worked on our large-scale task.

Validation We randomly selected 20% of the articles from TORQUE for further validation. We first validated the events by 4 different level-2 annotators (with the original annotator, there were in total 5 different humans). We also intentionally added noise to the original event list so that the validators must carefully identify wrong events. The final event list was determined by aggregating all 5 humans using majority vote. Second, we validated the answers in the same portion of the data. Two level-2 workers were asked to verify the initial annotator's answers; we again added noise to the answer list as a quality control for the validators. Instead of using majority vote as we did for events, the final answers from all workers are considered correct. We did not do additional validation for the questions themselves, as a manual inspection found the quality to be very high already, with no bad questions in a random sample of 100.

4.2 Cost

In each job of the main task, we presented 3 passages. The crowd worker could decide to use some or all of them. For each passage a worker decided to use, they needed to label the events, answer 3 hard-coded warm-up questions, and then ask and answer at least 12 questions (including contrast questions). The final reward is a base pay of \$6 plus \$0.5 for each extra question. Crowd workers thus had the incentive to (1) use fewer passages so that they can do event labeling and warm-up questions fewer times, (2) modify questions instead of asking from scratch, and (3) ask extra questions in each job. All these incentives were for more coverage of the temporal phenomena in each passage. In practice, crowd workers on average used 2 passages in each job. Validating the events in each passage and the answers to a specific question both cost \$0.1. In total, TORQUE cost \$15k for an average of \$0.70/question.

5 TORQUE Statistics

TORQUE has 3.2k passage annotations (∼50 tokens/passage),[6] 24.9k events (7.9 events/passage), and 21.2k user-provided questions (half of them were labeled by crowd workers as modifications of existing ones). Every passage comes with 3 hard-coded warm-up questions asking which events in the passage had already happened, were ongoing, or were still in the future. Table 3 shows some basic statistics of TORQUE. Note the 3 warm-up questions form a contrast set, and we treat the first as "original" and the others "modified."

In a random sample of 200 questions in the test set of TORQUE, we found 94 questions querying about relations that cannot be directly represented by the previous single-interval-based labels. Table 2 gives example questions capturing these phenomena. More analysis of the event, answer, and workload distributions are in Appendix A-D.

5.1 Quality

To validate the event annotations, we took the events provided by the initial annotator, added noise, and asked different workers to validate. We also trained an auxiliary event detection model using RoBERTa-large and added its predictions as event candidates. This tells us about the quality of events in TORQUE in two ways. First, the Worker Agreement with Aggregate (WAWA) F_1 here is 94.2%; that is, compare the majority-vote with all annotators, and perform micro-average on all instances. Second, if an event candidate is labeled by both the initial annotator and the model, then almost all of them (99.4%) are kept by the validators; if neither the initial annotator nor the model labeled a candidate, the candidate is almost surely removed (0.8%). As validators did not know which ones were noise or not before-

[6] Since the passages were selected randomly with replacement, there are 2.9k unique passages in total.

Type	Subtype	Example	%
Standard		*"What happened before Bush gave four key speeches?"*	53%
Fuzzy	begin only	*"What started before Mr. Fournier was prohibited from organizing his own defense?"*	15%
	overlap only	*"What events were occurring during the competition?"*	10%
	end only	*"What will end after he is elected?"*	1%
Modality	uncertain	*"What might happen after the FTSE 100 index was quoted 9.6 points lower?"*	10%
	negation	*"What has not taken place before the official figures show something?"*	5%
	hypothetical	*"What event will happen if the scheme is broadened?"*	2%
	repetitive	*"What usually happens after common shares are acquired?"*	1%
Misc.	participant	*"What did Hass do before he went to work as a spy?"*	4%
	opinion	*"What should happen in the future according to Obama's opinion?"*	3%
	intention	*"What did Morales want to happen after Washington had a program to eradicate coca?"*	1%

Table 2: Temporal phenomena in TORQUE. "Standard" are those that can be directly captured by the previous single-interval-based label set, while other types cannot. Percentages are based on manual inspection of a random sample of 200 questions from TORQUE; some questions can have multiple types.

	Q	Q/P	A	A/Q
Overall	30.7k	9.7	65.0k	2.1
Warm-up	9.5k	3	21.6k	2.3
* *Original*	3.2k	1	12.8k	4.0
* *Modified*	6.3k	2	8.8k	1.4
User-provided	21.2k	6.7	43.4k	2.1
* *Original*	10.6k	3.4	25.1k	2.4
* *Modified*	10.6k	3.3	18.3k	1.7

Table 3: Columns from left to right: questions, questions per passage, answers, and answers per question. *Modified* is a subset of questions that is created by slightly modifying an *original* question.

hand, this indicates that the validators could identify noise terms reliably.

Similarly, the WAWA F_1 of the answer annotations is 84.7%, slightly lower than that for events, which is expected because temporal relation QA is intuitively harder. Results show that 12.3% of the randomly added answer candidates were labeled as correct answers by the validators. We manually inspected 100 questions and found 11.6% of the added noise terms were correct answers (very close to 12.3%), indicating that the validators were actually doing a good job in answer validation. More details of the metrics and the quality of annotations can be found in Appendix E.

6 Experiment

We split TORQUE into train (80% of all the questions), dev (5%), and test (15%) and these three parts do not have the same articles. To solve TORQUE in an end-to-end fashion, the model here takes as input a passage and a question, then looks at every token in the passage and makes a binary classification of whether this token is an answer to the question or not. Specifically, the model has a one-layered perceptron on top of BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019), and the input to the perceptron layer is the transformers' output corresponding to the token we're looking at. We fine-tuned BERT/RoBERTa (both "base" and "large") on the training set of TORQUE. We fixed batch size = 6 (each instance is a tuple of one passage, one question, and all its answers) with gradient accumulation step = 2 in all experiments. We selected the learning rate (from $(1e^{-5}, 2e^{-5})$), the training epoch (within 10), and the random seed (from 3 arbitrary ones) based on performance on the dev set of TORQUE.[7] To compute an estimate of human performance, one author answered 100 questions from the test set and compared with crowd workers' annotations.

Both the human performance and system performances are shown in Table 4. We report the standard macro F_1 and exact-match (EM) metrics in question answering, and also EM consistency, the percentage of contrast question sets for which a model's predictions match exactly to all questions in a group (Gardner et al., 2020). We see warm-up questions are easier than user-provided ones because warm-up questions focus on easier phenomena of past/ongoing/future events. In addition, RoBERTa-large is expectedly the best system, but still far behind human performance, trailing by about 30% in EM.

We further downsampled the training data to test the performance of RoBERTa. We find that with 10% of the original training data, RoBERTa fails to learn anything meaningful and simply pre-

[7] More reproducibility information in Appendix F.

	Dev			Test								
				Overall			Warm-up questions			User-provided		
	F_1	EM	C	F_1	EM	C	F_1	EM	C	F_1	EM	C
Human	-	-	-	*95.3*	*84.5*	*82.5*	*95.7*	*89.7*	*90.9*	*95.1*	*82.4*	*79.3*
BERT-base	67.6	39.6	24.3	67.2	39.8	23.6	72.9	46.2	28.8	64.8	37.1	21.3
BERT-large	72.8	46.0	30.7	71.9	45.9	29.1	75.0	50.1	30.3	70.6	44.1	28.5
RoBERTa-base	72.2	44.5	28.7	72.6	45.7	29.9	75.4	48.8	32.3	71.4	44.4	28.8
RoBERTa-large	**75.7**	**50.4**	**36.0**	**75.2**	**51.1**	**34.5**	**77.3**	**54.3**	**36.1**	**74.3**	**49.8**	**33.8**

Table 4: Human/system performance on the test set of TORQUE. System performance is averaged from 3 runs; all std. dev. were $\leq 4\%$ and those in $[1\%, 4\%]$ are underlined. C (consistency) is the percentage of contrast groups for which a model's predictions have $F_1 \geq 80\%$ for all questions in a group (Gardner et al., 2020).

dicts "not an answer" for all tokens. With 50% of the training data, RoBERTa is slightly lower than but already comparable to that of using the entire training set. This means that the learning curve on TORQUE is already flat and the current size of TORQUE may not be the bottleneck for its low performance. Our data and code are public to facilitate more investigations into TORQUE.[8]

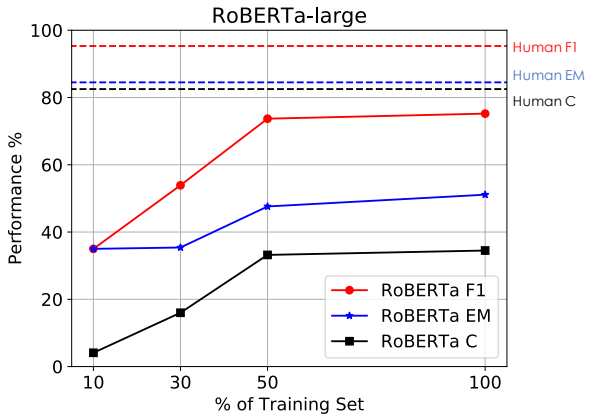

Figure 11: RoBERTa-large with different percentage of training data. Human performance in dashed lines.

7 Related Work

The study of *time* is to understand *when, how long, and how often* things happen. While *how long* and *how often* usually require temporal common sense knowledge (Vempala et al., 2018; Zhou et al., 2019, 2020), the problem of *when* often boils down to extracting temporal relations.

Modeling. Research on temporal relations often focuses on algorithmic improvement, such as structured inference (Do et al., 2012; Chambers et al., 2014; Ning et al., 2018a), structured learning (Leeuwenberg and Moens, 2017; Ning et al., 2017), and neural networks (Dligach et al., 2017; Lin et al., 2017; Tourille et al., 2017; Cheng and Miyao, 2017; Meng and Rumshisky, 2018; Leeuwenberg and Moens, 2018; Ning et al., 2019).

Formalisms. The approach that prior works took to handle the aforementioned temporal phenomena was to define formalisms such as the different modes of events (Fig. 3), a predefined label set (Fig. 4), different time axes for events (Ning et al., 2018b), and specific rules to follow when there is confusion. For example, Bethard et al. (2007); Ning et al. (2018b) focused on a limited set of temporal phenomena and achieved high inter-annotator agreements (IAA), while Cassidy et al. (2014); Styler IV et al. (2014); O'Gorman et al. (2016) aimed at covering more phenomena but suffered from low IAAs even between NLP researchers.

QA as annotation. A natural choice is then to cast temporal relation understanding as a machine reading comprehension (MRC) problem. TORQUE is motivated by the philosophy in QA-SRL (He et al., 2015) and QAMR (Michael et al., 2017), where QA pairs were used as representations for predicate-argument structures. In zero-shot relation extraction (RE), they reduced relation slot filling to an MRC problem so as to build very large distant training data and improve zero-shot learning performance (Levy et al., 2017). However, our work differs from zero-shot RE since it centers around entities, while TORQUE is about events; the way to ask and answer questions, and the way to design a corresponding crowdsourcing pipeline, are thus significantly different between us.

The QA-TempEval workshop (Llorens et al., 2015), desipte its name, is actually not studying temporal relations in an RC setting. The differences between TORQUE and QA-TempEval

[8] https://allennlp.org/torque.html

are as follows. First, QA TempEval is an evaluation approach for systems that generate TimeML annotations and actually is not a QA task. For instance, QA TempEval is to evaluate whether a system can answer questions like "Is <ENTITY_1> <RELATION> <ENTITY_2>?", where one clearly knows which event that <ENTITY> is referring to and where RELATION is selected from a predefined label set. Second, QA-TempEval's annotation relies on the existence of a TimeML corpus. From the perspective of data collection for studying a particular phenomenon, TORQUE has done more on defining the task and developing a scalable crowdsourcing pipeline. As a result, TORQUE is also much larger than QA-TempEval and the annotation pipeline of TORQUE can be easily adopted to collect even more data.

8 Conclusion

Understanding temporal ordering of events is critical in reading comprehension, but existing works have studied very little about it. This paper presents TORQUE, a new English machine reading comprehension (MRC) dataset of temporal ordering questions. TORQUE has 3.2k news snippets, 9.5k hard-coded questions asking which events had happened, were ongoing, or were still in the future, and 21.2k human-generated questions querying more complex phenomena. We argue that an MRC setting allows for more convenient representation of these temporal phenomena than conventional formalisms. Results show that even a state-of-the-art language model, RoBERTa-large, falls behind human performance by a large margin, necessitating more investigation on improving MRC on temporal relationships in the future.

Acknowledgments

This work was partly supported by contract FA8750-19-2-1004 and contract W911NF-15-1-0543, both with the US Defense Advanced Research Projects Agency (DARPA), and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the BETTER Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

2005. The ACE 2005 (ACE 05) Evaluation Plan. Technical report.

James F Allen. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.

Steven Bethard, James H Martin, and Sara Klingenstein. 2007. Timelines from text: Identification of syntactic temporal relations. In *IEEE International Conference on Semantic Computing (ICSC)*, pages 11–18.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 Task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062, San Diego, California. Association for Computational Linguistics.

Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. Semeval-2017 task 12: Clinical tempeval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572. Association for Computational Linguistics.

Taylor Cassidy, Bill McDowell, Nathanel Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 501–506.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics (TACL)*, 2:273–284.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 1–6.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5925–5932.

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 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. In *Proceedings of the Conference of the European Chapter of the Association*

for Computational Linguistics (EACL), volume 2, pages 746–751.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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

Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. 2019a. On making reading comprehension more comprehensive. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*.

Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. 2019b. Question answering is a format; when is it useful? *arXiv preprint arXiv:1909.11291*.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hangfeng He, Qiang Ning, and Dan Roth. 2020. QuASE: Question-answer driven sentence encoding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8743–8758.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 643–653.

Egoitz Laparra, Dongfang Xu, Ahmed Elsayed, Steven Bethard, and Martha Palmer. 2018. SemEval 2018 task 6: Parsing time normalizations. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 88–96.

Artuur Leeuwenberg and Marie-Francine Moens. 2017. Structured learning for temporal relation extraction from clinical records. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 333–342.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2017. Representations of time expressions for temporal relation extraction with convolutional neural networks. *BioNLP 2017*, pages 322–327.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015. SemEval-2015 Task 5: QA TEMPEVAL - evaluating temporal information understanding with question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 792–800.

Yuanliang Meng and Anna Rumshisky. 2018. Context-aware neural model for temporal information extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 527–536.

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2017. Crowdsourcing question-answer meaning representations. *arXiv preprint arXiv:1711.05885*.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, Ruben Urizar, and Fondazione Bruno Kessler. 2015. SemEval-2015 Task 4: TimeLine: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786.

T. Mitamura, Y. Yamakawa, S. Holm, Z. Song, A. Bies, S. Kulick, and S. Strassel. 2015. Event nugget annotation: Processes and issues. In *Proceedings of the Workshop on Events at NAACL-HLT*.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016. CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the 4th Workshop on Events: Definition, Detection, Coreference, and Representation*, pages 51–61.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1038–1048, Copenhagen, Denmark. Association for Computational Linguistics.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. Joint reasoning for temporal and causal relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2278–2288. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An Improved Neural Baseline for Temporal Relation Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qiang Ning, Hao Wu, and Dan Roth. 2018b. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1318–1328. Association for Computational Linguistics.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer Event Description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56, Austin, Texas. Association for Computational Linguistics.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The TIMEBANK corpus. In *Corpus Linguistics*, page 40.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 784–789.

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

Jessica Semega, Melissa Kollar, John Creamer, and Abinash Mohanty. 2019. *Income and Poverty in the United States: 2018*. U.S. Department of Commerce.

William F Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics (TACL)*, 2:143.

Julien Tourille, Olivier Ferret, Aurelie Neveol, and Xavier Tannier. 2017. Neural architecture for temporal relation extraction: A Bi-LSTM approach for detecting narrative containers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 224–230.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 Task 1: TEMPEVAL-3: Evaluating time expressions, events, and temporal relations. *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM)*, 2:1–9.

Alakananda Vempala, Eduardo Blanco, and Alexis Palmer. 2018. Determining event durations: Models and error analysis. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, volume 2, pages 164–168.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80. Association for Computational Linguistics.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "Going on a vacation" takes longer than "Going for a walk": A Study of Temporal Commonsense Understanding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal Common Sense Acquisition with Minimal Supervision. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Appendix

A Event Distribution

As we mentioned in Sec. 5, TORQUE has 24.9k events over 3.2k passages. Figure 12 shows the histogram of the number of events in all these passages. We can see it roughly follows a Gaussian distribution with the mean at around 7-8 events per passage.

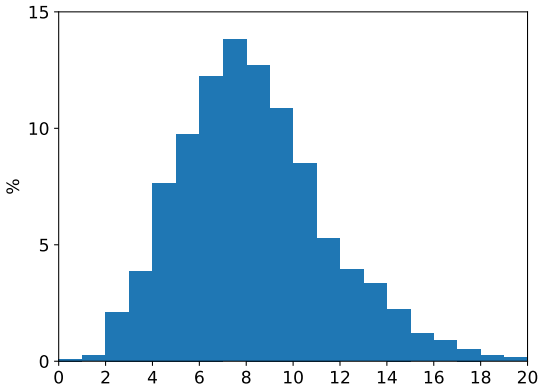

Figure 12: Histogram of the number of events in all passages in TORQUE.

Figure 13 further shows the 50 most common events in TORQUE. Unsurprisingly, the most common events are reporting verbs (e.g., "say", "tell", "report", and "announce") and copular verbs. Other common events such as "meeting", "killed", "visit", and "war" are also expected given that the passages of TORQUE were taken from news articles.

B Question Prefix Distribution

Figure 14 shows a sunburst visualization of the questions provided by crowd workers in TORQUE, including both their original questions and their modifications. Specifically, Fig. 14a shows that almost all of the questions start with "what." The small portion of questions that do not start with "what" are cases where crowd workers switch the order of how they ask. One example of these was *"Before making his statement to the Sunday Mirror, what did the author do?"* Figure 14a also shows the most common following words of "what."

Figures 14b-c further show the distribution of questions starting with "what happened" and "what will." We can see that when asking things in the past, people ask more about "what happened before/after" than "what happened while/during," while when asking things in the future, people ask much more about "what will happen after" than "what will happen before."

C Answer Distribution

The distribution of the number of answers to each question is shown in Fig. 15, where we divide the questions into 4 categories: the original warm-up questions, the modified warm-up questions, the original user questions, and the modified user questions. Note for each passage, there are 3 warm-up questions and they were all hard-coded when crowd workers worked on them. We are treating the first one (i.e., *"What events have already finished?"*) as the original and the other two as modified (i.e., *"What events have begun but have not finished?"* and *"What will happen in the future?"*).

We can see that in both the warm-up and the user questions, "modified" has a larger portion of questions with no answers at all as compared to the "original." This effect is very significant for warm-up questions because in news articles, most of the events were in the past. As for the user-provided questions, the percentage of no-answer questions is higher in "modified," but it is not as drastic as for the warm-up question. This because we only required that the modified question should have different answers from the original one; many of those questions sill have answers after modification.

D Workload Distribution Among Workers

As each annotator may be biased to only ask questions in a certain way, it is important to make sure that the entire dataset is not labeled by only a few annotators Geva et al. (2019). Figure 16a shows the contribution of each crowd worker to TORQUE and we can see even the rightmost worker only provided 5%. Figure 16b further adopts the notion of *Gini Index* to show the dispersion.[9] The Gini index of TORQUE is 0.42.

[9] A high Gini Index here means the data were provided by a small group of workers. The Gini Index of family incomes in the United States was 0.49 in 2018 (Semega et al., 2019).

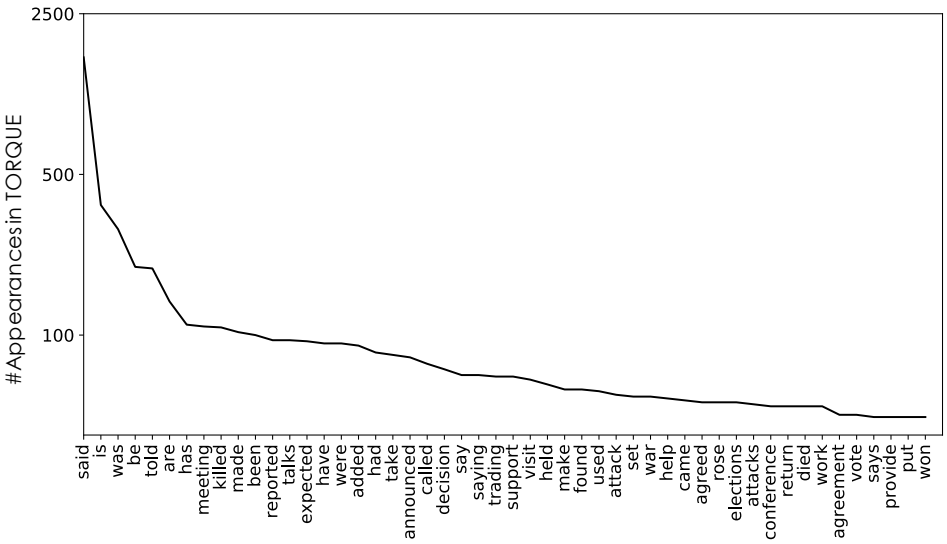

Figure 13: Fifty most common event triggers in TORQUE. Note the y-axis is in log scale.

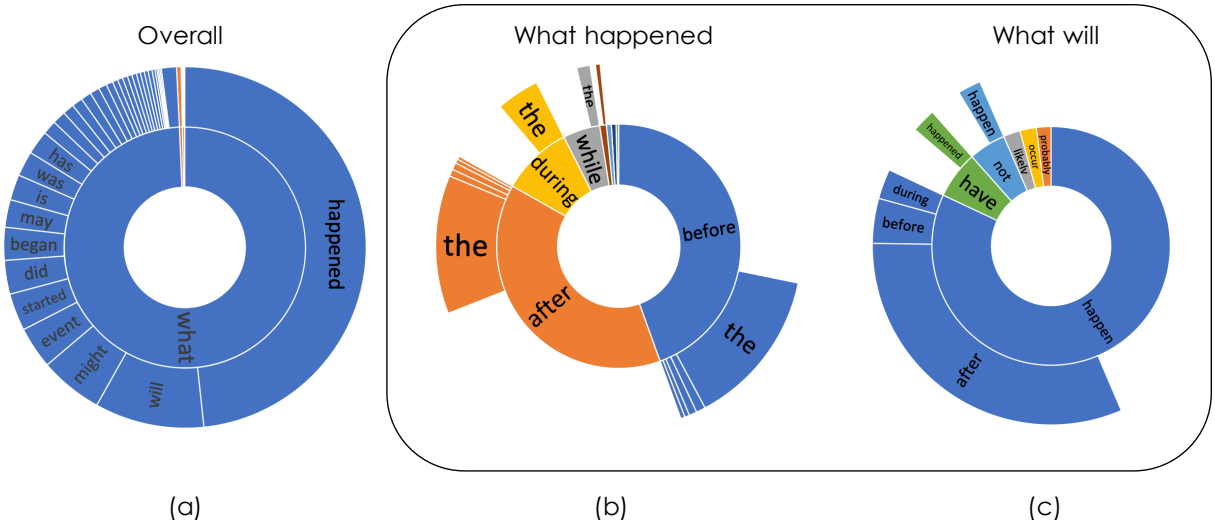

Figure 14: Prefix distribution of user-provided questions.

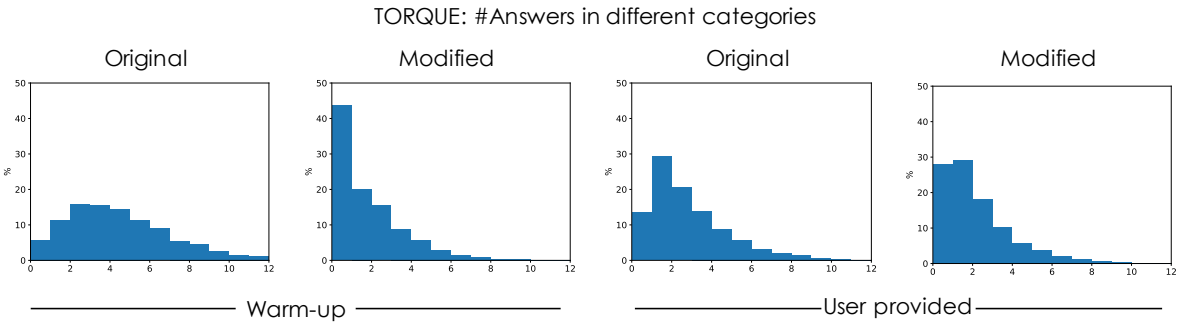

Figure 15: Distribution of the number of answers to each question.

E Worker Agreement With Aggregate

In Sec. 5 we described the worker agreement with aggregate (WAWA) metric for measuring the inter-annotator agreement (IAA) between crowd workers of TORQUE. This WAWA metric is explained in the figure below. It is to first get an aggregated

Figure 16: If every annotator provided the same number of passages (i.e., perfect equality), the curve would be the straight dashed line and the Gini Index would be 0. If one person provided all the annotations, the Gini Index is 1.

answer set from multiple workers (we used majority vote as the aggregate function), then compare each worker with the aggregated answer set, and finally compute the micro-average across multiple workers and multiple questions.

Tables 5 and 6 show the quality of event annotations and question-answering annotations, respectively. In both of them, the IAA are using the WAWA metric explained above; the "Init Annotator" rows are a slight modification of WAWA, which means that all workers are used when aggregating those answers, but only the first annotator is compared against the aggregated answer set. Table 5 further shows the agreement between the init annotator and an event detection model, which we have described in Sec. 5.

	P	R	F
IAA (WAWA)	94.3%	94.1%	94.2%
Init Annotator	94.9%	89.8%	92.3%

		Init Annotator	
		Yes	No
Model	Yes	99.4%	82.0%
	No	64.1%	0.8%

Table 5: Inter-annotator agreement (IAA) of the event annotations in TORQUE. Above: compare the aggregated event list with either all the annotators or the initial annotator. Below: how many candidates in each category were successfully added into the aggregated event list.

	P	R	F
IAA (WAWA)	82.3%	87.3%	84.7%
Init Annotator	91.3%	82.2%	86.5%

Table 6: IAA of the answer annotations in TORQUE.

F Reproducibility

- We ran our experiments on PyTorch 1.3.1. Pre-trained language models are implemented in the Huggingface transformers library.
- A single GeForce RTX 2080 GPU was used to finetune a model. CUDA Version 10.2. The average time to run an epoch was 38 minutes for the full training section of TORQUE using RoBERTa-large.
- The best performing model consist of RoBERTa-large + final MLP layer, and the number of parameters is 355M + 1024 * 64 + 64 * 2.

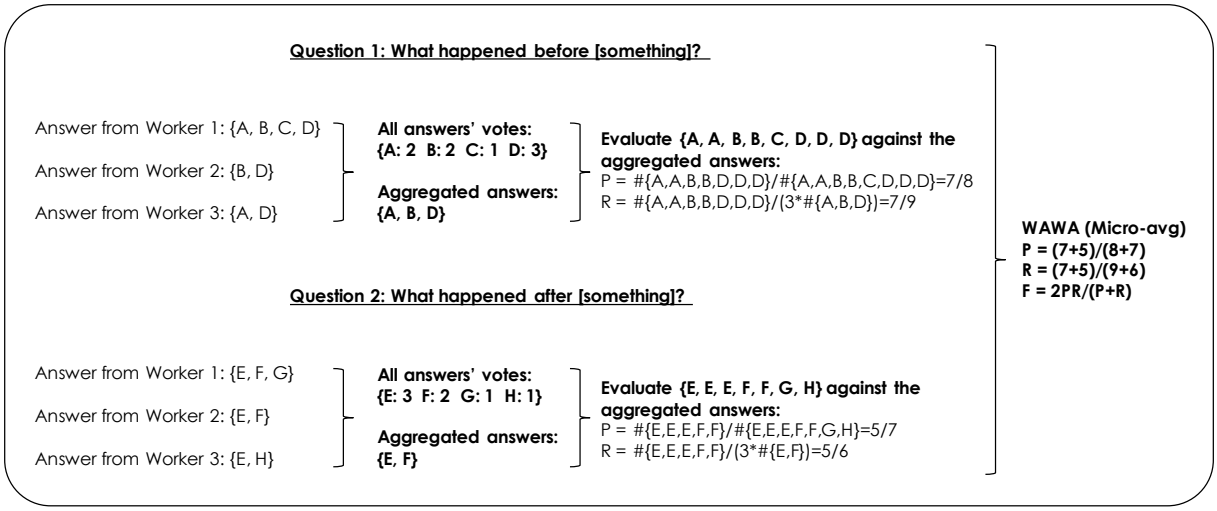

Figure 17: Explanation of the worker agreement with aggregate (WAWA) metric.