Asking It All: Generating Contextualized Questions for any Semantic Role

Valentina Pyatkin[*,1] **Paul Roit**[*,1] **Julian Michael**[2]
Reut Tsarfaty[1,3] **Yoav Goldberg**[1,3] **Ido Dagan**[1]
[1]Computer Science Department, Bar Ilan University
[2]Paul G. Allen School of Computer Science & Engineering, University of Washington
[3]Allen Institute for Artificial Intelligence
{valpyatkin,plroit,yoav.goldberg}@gmail.com
julianjm@cs.washington.edu, reut.tsarfaty@biu.ac.il, dagan@cs.biu.ac.il

Abstract

Asking questions about a situation is an inherent step towards understanding it. To this end, we introduce the task of role question generation, which, given a predicate mention and a passage, requires producing a set of questions asking about all possible semantic roles of the predicate. We develop a two-stage model for this task, which first produces a context-independent question prototype for each role and then revises it to be contextually appropriate for the passage. Unlike most existing approaches to question generation, our approach does not require conditioning on existing answers in the text. Instead, we condition on the type of information to inquire about, regardless of whether the answer appears explicitly in the text, could be inferred from it, or should be sought elsewhere. Our evaluation demonstrates that we generate diverse and well-formed questions for a large, broad-coverage ontology of predicates and roles.

1 Introduction

Soliciting information by asking questions is an essential communicative ability, and natural language question-answer (QA) pairs provide a flexible format for representing and querying the information expressed in a text. This flexibility has led to applications in a wide range of tasks from reading comprehension (Rajpurkar et al., 2016) to information seeking dialogues (Qi et al., 2020).

Automatically generating questions can potentially serve as an essential building block for such applications. Previous work in question generation has either required human-curated templates (Levy et al., 2017; Du and Cardie, 2020), limiting coverage and question fluency, or generated questions for answers already identified in the text (Heilman and Smith, 2010; Dhole and Manning, 2020). Open-ended generation of information-seeking questions, where the asker poses a question inquiring about a certain type of information, remains a significant challenge.

The plane took off in Los Angeles. The tourists will **arrive** in Mexico at noon.

entity in motion	Who will arrive in Mexico?
end point	Where will the tourists arrive?
start point	Where will the tourists arrive from?
manner	How will the tourists arrive?
cause	Why will the tourists arrive?
temporal	When will the tourists arrive?

Figure 1: Example role questions for "arrive". Some questions are for explicit arguments (*entity in motion*,*end point*, *temporal*), some for implicit (*start point*, *manner*) ones, and some for arguments that do not appear at all (*cause*).

In this work, we propose to resolve these difficulties by generating *role questions*. In particular, for any predicate expressed in a text and semantic role associated with that predicate, we show how to generate a contextually-appropriate question whose answer—if present—corresponds to an argument of the predicate bearing the desired role. Some examples are shown in Figure 1. Since the set of possible questions is scoped by the relations in the underlying ontology, this gives us the ability to *ask it all*: pose information-seeking questions that exhaustively cover a broad set of semantic relations that may be of interest to downstream applications.

Concretely, we generate questions derived from QA-SRL (He et al., 2015) for the semantic role ontology in PropBank (Palmer et al., 2005) using a two-stage approach. In the first stage (§4.1), we leverage corpus-wide statistics to compile an ontology of simplified, context-independent *prototype questions* for each PropBank role. In the second

[*]Equal contribution

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1429–1441
November 7–11, 2021.

WH	AUX	SBJ	VERB	OBJ	PREP	MISC	?
Who	might		bring	something	to	someone	?
Where	would	someone	arrive		at		?
What	was	something	sold		for		?

Table 1: QA-SRL slot format. WH and VERB are mandatory, and AUX and VERB encode tense, modality, negation, and active/passive voice.

stage (§4.2), we contextualize the question using a learned translation from prototypes to their contextualized counterparts, conditioned on a sentence. To that end we present a new resource of *frame-aligned QA-SRL questions* which are grounded in their source context. This setup decouples posing a question that captures a semantic role from fitting that question to the specific context of the passage. As we show in §5, the result is a system which generates questions that are varied, grammatical, and contextually appropriate for the passage, and which correspond well to their underlying role.[1]

The ability to exhaustively enumerate a set of questions corresponding to a known, broad-coverage underlying ontology of relations allows for a comprehensive, interpretable, and flexible way of representing and manipulating the information that is contained in—or missing from—natural language text. In this way, our work takes an essential step towards combining the advantages of formal ontologies and QA pairs for broad-coverage natural language understanding.

2 Background

Question Generation Automatic question generation has been employed for use cases such as constructing educational materials (Mitkov and Ha, 2003), clarifying user intents (Aliannejadi et al., 2019), and eliciting labels of semantic relations in text (FitzGerald et al., 2018; Klein et al., 2020; Pyatkin et al., 2020). Methods include transforming syntactic trees (Heilman and Smith, 2010; Dhole and Manning, 2020) and SRL parses (Mazidi and Nielsen, 2014; Flor and Riordan, 2018), as well as training seq2seq models conditioned on the question's answer (FitzGerald et al., 2018) or a text passage containing the answer (Du et al., 2017). By and large, these approaches are built to generate questions whose answers are already identifiable in a passage of text.

However, question generation has the further potential to seek *new* information, which requires asking questions whose answers may only be implicit, inferred, or even absent from the text. Doing so requires prior specification of the kind and scope of information to be sought. As a result, previous work has found ways to align existing relation ontologies with questions, either through human curation (Levy et al., 2017) — which limits the approach to very small ontologies — or with a small set of fixed question templates (Du and Cardie, 2020) — which relies heavily on glosses provided in the ontology, sometimes producing stilted or ungrammatical questions. In this work, we generate natural-sounding information-seeking questions for a broad-coverage ontology of relations such as that in PropBank (Bonial et al., 2014).

QA-SRL Integral to our approach is QA-SRL (He et al., 2015), a representation based on question-answer pairs which was shown by Roit et al. (2020) and Klein et al. (2020) to capture the vast majority of arguments and modifiers in PropBank and NomBank (Palmer et al., 2005; Meyers et al., 2004). Instead of using a pre-defined role lexicon, QA-SRL labels semantic roles with questions drawn from a 7-slot template, whose answers denote the argument bearing the role (see Table 1 for examples). Unlike in PropBank, QA-SRL argument spans may appear outside of syntactic argument positions, capturing *implicit* semantic relations (Gerber and Chai, 2010; Ruppenhofer et al., 2009).

QA-SRL is useful to us because of its close correspondence to semantic roles and its carefully restricted slot-based format: it allows us to easily transform questions into context-independent *prototypes* which we can align to the ontology, by removing tense, negation, and other information immaterial to the semantic role (§4.1). It also allows us to produce *contextualized* questions which sound natural in the context of a passage, by automatically aligning the syntactic structure of different questions for the same predicate (§4.2).

3 Task Definition

Our task is defined with respect to an ontology of semantic roles such as PropBank. Given a passage of text with a marked predicate and a chosen role of that predicate, we aim to generate a naturally-phrased question which captures that semantic relation. For example, consider Figure 1. For the predicate `arrive.01` and role A0 (defined in PropBank as *entity in motion*), we want to gener-

[1] Our code and resources can be found here: https://github.com/ValentinaPy/RoleQGeneration

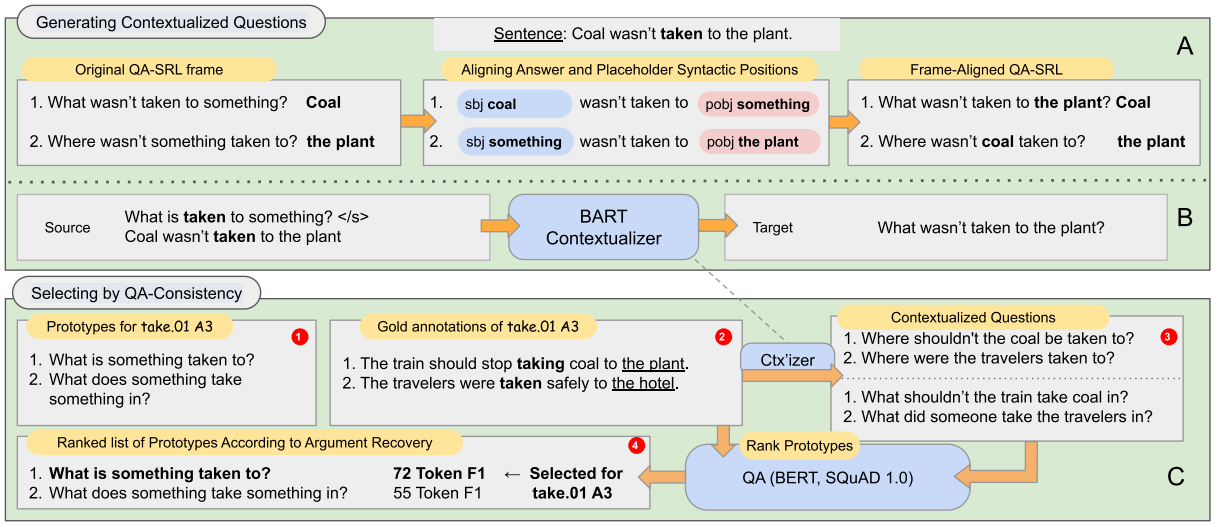

Figure 2: **A: Construction of Frame-Aligned QA-SRL** (§4.2) using syntactic information inferred by the autocomplete NFA from FitzGerald et al. (2018). **B: Contextualizing questions.** The contextualizer takes in a prototype question and a context, and outputs a Frame-Aligned QA-SRL question. Note how Frame-Aligned QA-SRL questions preserve the role expressed in their respective prototypes. **C: Selecting prototype questions** according to their ability to recover explicit arguments (see §4.1). We test each prototype (1) against a sample of arguments for each role (2). Contextualized versions of each prototype question for each sampled sentence (3) are fed (with the sentences) to a QA model, and the predicted answer is evaluated against the gold argument. The highest ranking prototype (4) is selected for that role.

ate the question *Who will arrive in Mexico?*. In this task, the lack of an answer in the text should not preclude the system from generating a question that pertains to a role. Rather, the question should be such that *if* an answer exists, it denotes an argument bearing that role.

4 Method

Generating role questions requires mapping from an input sentence, a predicate, and role to a natural language question, regardless of whether the question is answerable in the text. However, we don't have training data of this nature: existing question-answer driven approaches to semantic annotation (FitzGerald et al., 2018) only elicit *answerable* questions from annotators, and existing resources with unanswerable questions (Rajpurkar et al. (2018), *inter alia*) do not systematically align to semantic roles in any clear way. Indeed, we will show in §5.2 that training a seq2seq model to directly generate role questions from question-answer pairs annotated with explicit semantic roles yields poor results for implicit or missing arguments, because the model overfits towards only asking answerable questions.

Instead, we adopt a **two-stage approach**: first, we leverage corpus-wide statistics to map each role in an ontology to a context-independent **prototype question** which provides the high-level syntactic structure of any question for that role. For example, `bring.01`'s *destination* argument A2 may receive the prototype question *where does something bring something?* Second, we employ a **question contextualization** step which aligns the tense, negation/modality, and entities in the question with those in the particular source text, e.g., *where did the pilot bring the plane?*. The result is a question about the source text which is amenable to traditional QA systems and whose semantically correct answer should relate to the predicate according to the given role. Figure 3 shows example prototype and contextualized questions for the predicate `change.01`.[2]

4.1 Generating Prototype Questions

In our first step, we introduce *prototype questions* to serve as intermediate, *context-independent* representations of each role in the ontology. We obtain these prototypes by automatically producing a large dataset of QA-SRL questions jointly labelled with

[2]In this work, we target both verbal predicates (*e.g.*, *eat*, *give*, *bring*) and deverbal nouns (such as *sale*, *presentation*, and *loss*), using the frame index associated with the OntoNotes corpus (Weischedel et al., 2017). In addition to the core roles for each predicate, we generate questions for the adjunct roles LOCATIVE, TEMPORAL, MANNER, CAUSAL, EXTENT, and GOAL.

causer of transformation	Who changes something?	Who might have changed their minds?
thing changing	What is changed ?	What might have been changed?
end state	What is something changed to?	What might their minds be changed to?
start state	What is changed into something?	What might have been changed into something?

Figure 3: Inference for change.01 given the sentence: *"The only thing that might've changed their minds this quickly I think is money"*. For each role (left) we look-up a context-independent question (center) and apply a contextualizing transformation to produce a sound question (right).

PropBank roles. Each question is then stripped of features which are irrelevant to the underlying semantic role, and the resulting prototypes are aggregated for each role. Finally, we choose the prototype which allows a question answering model to most accurately recover the arguments corresponding to that role. The end result is the assignment of a single prototype question to each semantic role in the ontology, to be fed into the second step (§4.2).

Joint PropBank and QA-SRL We begin by aligning PropBank roles with QA-SRL questions. We do this in two ways: First, we run the SRL parser[3] of Shi and Lin (2019) on the source sentences of the QA-SRL Bank 2.0 (FitzGerald et al., 2018) and QANom (Klein et al., 2020). To be aligned, the answer must have significant ($\geq$0.4 intersection-over-union) overlap with the predicted SRL argument and the question must target the same predicate. Second, we run the QA-SRL question generator from FitzGerald et al. (2018) on the gold predicates and arguments in the OntoNotes training set to produce QA-SRL questions aligned to each argument. Altogether, this produces a total of 543K instances of jointly-labeled PropBank arguments and QA-SRL questions. We manually check 200 sampled instances of both verbal and nominal aligned QA pairs, split equally, and find that 93% had no errors.

Aggregating Prototype Candidates For each role, we enumerate the full set of QA-SRL questions appearing for that role in our jointly labeled data.[4] We post-process the QA-SRL questions into what we call *question prototypes* using the same process as Michael and Zettlemoyer (2021): we remove negations, replace animate pronouns (*who*/*someone*) with inanimate ones (*what*/*something*), and convert all questions to the simple present tense (which can be easily done in QA-SRL's slot-based format). This eliminates aspects of a question which are specific of a particular text, while retaining aspects that are potentially indicative of the underlying semantic role, such as the question's syntactic structure, active/passive distinction, prepositions, and *wh*-word (*e.g.*, *when/where*). For example, the questions *What will be fixed?* and *What won't be fixed?* have conflicting meanings, but both target the same semantic role (the THEME of `fix`) and they receive the same prototype (*What is fixed?*). Full details of this method are described in Appendix B.

Selecting by QA Consistency Of the prototype candidates for each role, we want to choose the one with the right specificity: for example, *where does something win?* is a more appropriate prototype for an `AM-LOC` modifier role than than *what does someone win at?*, even if the latter appears for that role. This is because *what does someone win at?* is at once too specific (inappropriate for locations better described with *in*, *e.g.*, *in Texas*) and too general (also potentially applying to `win.01`'s `A2` role, the contest being won).

To choose the right prototype, we run a consistency check using an off-the-shelf QA model (see Figure 2, Bottom). We sample a set of gold arguments[5] for the role from OntoNotes (Weischedel et al., 2017) and instantiate each prototype for each sampled predicate using the question contextualizer described in the next section (§4.2). We then select the prototype for which a BERT-based QA model (Devlin et al., 2019) trained on SQuAD 1.0 (Rajpurkar et al., 2016) achieves the highest token-wise F1 in recovering the gold argument from the contextualized question.

4.2 Generating Contextualized Questions

For our second stage, we introduce a *question contextualizer* model which takes in a prototype question and passage, and outputs a contextualized ver-

[3]We use AllenNLP's (Gardner et al., 2018) public model for verbal SRL, and train our own parser on the original NomBank data.

[4]To increase coverage, we share prototypes for the same role label between different senses of the same predicate.

[5]For core roles we sample 50 argument instances, and for adjunct roles we take 100 but select samples from any predicate sense.

*Air molecules move a lot and **bump** into things.*

QA-SRL:
What bumps into something?
↪ **Air molecules**
What does something bump into?
↪ **things**

Syntactic Alignment:
tense: *present*
SUBJ: **Air molecules**
OBJ: ∅
PP: into **things**

Frame-Aligned QA-SRL:
What bumps into **things**?
↪ **Air molecules**
What do **air molecules** bump into?
↪ **things**

Figure 4: Example QA-SRL contextualization. The autocomplete NFA from FitzGerald et al. (2018) keeps track of the syntactic position of the gap produced by wh-movement, which allows us to substitute the answer of one question in place of the placeholder pronouns in another. We also use simple heuristics to correct capitalization and a masked language model to correct verb agreement.

sion of the question, designed to sound as natural as possible and match the semantics of the passage (see Figure 2, Top). In particular, our model adjusts the tense, negation, and modality of the question and fills in the placeholder pronouns with their corresponding mentions in the surrounding context.

To train the contextualizer, we automatically construct a new resource — the Frame-Aligned QA-SRL Bank — which pairs QA-SRL question prototypes with their contextualized forms (see Figure 4 for an example). The latter are constructed on the basis of syntactic alignments between different questions about the same predicate instance. We then train a BART model (Lewis et al., 2020) on this data to directly perform contextualization.

Recovering Syntax from QA-SRL The first step to constructing contextualized QA-SRL questions is identifying the questions' underlying syntactic structure, which can be used to align each question's answers with the placeholder positions of other questions in the same frame. We use the autocomplete NFA from FitzGerald et al. (2018), which constructs a simple syntactic representation of QA-SRL questions with three grammatical functions: subject, object, a third argument which may be an object, prepositional phrase, or complement. This recovers the declarative form of a QA-SRL question: for example, *What bumps into something?* corresponds to the declarative clause *Something bumps into something*, and asks about the argument in the subject position. Questions with adverbial *wh*-words, like *When did something bump into something?*, are mapped to a declarative clause without the adverbial (*i.e.*, *Something bumped into something*).

In some cases, the syntax is ambiguous: for example, the question *What does something bump into?* may correspond to either *Something bumps something into* or *Something bumps into something*. To resolve ambiguities, we choose the interpretation of each question which is shared with the most other questions in the same frame. To handle ties, we fall back to heuristics listed in §C.1.

Aligning Answers with Placeholders Where multiple questions share their underlying syntax, we replace the placeholder pronouns (e.g., *something* or *someone*) in each with the answers corresponding to their syntactic position (see Figure 2, Top, and Figure 4). To increase coverage of placeholder pronouns, we extend the correspondences to hold between slightly different syntactic structures, e.g., the passive subject and transitive object. Finally, we correct capitalization with simple heuristics and fix subject-verb agreement using a masked language model. Full details are in §C.2. Altogether, this method fills in 91.8% of the placeholders in the QA-SRL Bank 2.0 and QANom.

Model We train a BART model (Lewis et al., 2020) taking a passage with a marked predicate and a prototype question as input, where the output is the contextualized version of the same question provided in the new Frame-Aligned QA-SRL Bank. This setup ensures that the basic structure indicative of the semantic role is left unchanged, since both the source and target questions are derived from the same original using role preserving transformations. As a result, the model learns to preserve the underlying semantic role, and to retain the tense, modality, negation and animacy information of the original question. Full training details are in §C.3.

Equipped with a question contextualizer, we can perform the full task: given a predicate in context and desired role, we retrieve the question prototype for the role (§4.1), pair it with the context, and run it through the contextualizer to produce the final role question.

5 Evaluation

To assess our system, we perform an intrinsic evaluation against a seq2seq baseline (§5.2) as well as comparisons to existing question generation systems (§5.3).

5.1 Metrics

For our evaluations, we measure three properties of role questions. First, **grammaticality**: is the question well-formed and grammatical? Second, **adequacy**: does the question make sense in context? For this property, a question's presuppositions must be satisfied; for example, the question *Who will bring the plane back to base?* is only adequate in a context where a plane is going to be brought to base. (Note that the answer does not necessarily have to be expressed.) Third, we measure **role correspondence**: does the question correspond to the correct semantic role?

For all of these measures, we source our data from existing SRL datasets and use human evaluation by a curated set of trusted workers on Amazon Mechanical Turk.[6] Automated metrics like BLEU or ROUGE are not appropriate for our case because our questions' meanings can be highly dependent on minor lexical choices (such as with prepositions) and because we lack gold references (particularly for questions without answers present).

We assess grammaticality and adequacy on a 5-point Likert scale, as previous work uses for similar measures (Elsahar et al., 2018; Dhole and Manning, 2020). We measure role correspondence with two metrics: *role accuracy*, which asks annotators to assign the question a semantic role based on PropBank role glosses, and *question answering accuracy*, which compares annotators' answers to the question against the gold SRL argument (or the absence of such an argument).[7]

5.2 Main Evaluation

Data We evaluate our system on a random sample of 400 predicate instances (1210 questions) from Ontonotes 5.0 (Weischedel et al., 2017) and 120 predicate instances (268 questions) from two small implicit SRL datasets: Gerber and Chai (2010, G&C) and Moor et al. (2013, ON5V). We generate questions for all core roles in each instance. For comparison to the baseline, we use all predicates from the implicit SRL datasets and a subsample of 100 from OntoNotes. We also evaluate questions for 5 modifier roles[8] on 100 OntoNotes instances.

End-to-End Baseline As a baseline, we use a BART-based seq2seq model trained to directly generate role questions from a text passage with a marked target predicate, PropBank role label, and role description; for example: `Some geologists ⟨p⟩ study ⟨/p⟩ the Moon . ⟨/s⟩ student A0 study.01`, where *student* is the gloss provided in PropBank for the `A0` role of `study.01`. To train the model, we use the contextualized questions in the Frame-Aligned QA-SRL dataset (§4.2) as outputs while providing their aligned predicted PropBank roles (as detailed in §4.1) in the input.

Main Results Results are in Table 2. We stratify on whether the argument matching the role was *explicit* (syntactically related to the predicate), *implicit* (present in the text, but not a syntactic argument), or *None* (missing from the text entirely). We combine *implicit* and *None* for OntoNotes since it only marks explicit arguments.

Role Correspondence On role accuracy and QA accuracy, our model showed its strongest performance on explicit arguments, as well as implicit arguments in the G&C/ON5V datasets. It significantly outperforms the baseline on these metrics, with a 26 point gain in role accuracy and 11 point gain in QA accuracy on average. The difference is much greater for implicit and missing arguments, with a 38 point gain for RC and a 23 point gain for QA, showing how our approach excels at producing questions with the correct semantics for these arguments, despite their unavailability during training. Our results complement previous work (Moryossef et al., 2019) showing the utility of micro-planning with an intermediate representation to handle cases with little or no training data (*e.g.*, implicit arguments).

Grammaticality & Adequacy Average grammaticality and adequacy scores are shown in Table 2. In addition, for each measure we record the percentage of questions that both got a score $\geq$4 and were assigned to the correct role. Our questions were deemed grammatical and adequate

[6] Screenshots of the task interfaces are in Appendix D.

[7] Where an argument is present, we consider an annotator's answer correct if it includes its syntactic head.

[8] LOCATIVE, TEMPORAL, MANNER, CAUSAL, EXTENT, and GOAL

		Role Accuracy				QA Accuracy				Grammaticality		Adequacy		
		All	Expl.	Impl.	None	All	Expl.	Impl.	None	All	GRM+RC	All	ADQ+RC	Freq.
Onto*	RoleQ	0.72	0.81		0.64	-	0.75	-	-	4.25	60%	4.05	56%	1210
Onto	RoleQ	0.78	0.93		0.64	-	0.81	-	-	4.22	61%	4.07	58%	289
	e2e	0.51	0.88		0.25	-	0.85	-	-	4.34	49%	4.20	46%	289
G&C	RoleQ	0.76	0.82	0.80	0.63	0.50	0.36	0.58	0.57	4.45	69%	4.20	58%	120
	e2e	0.54	0.66	0.57	0.32	0.37	0.33	0.43	0.32	4.74	50%	4.53	51%	120
ON5V	RoleQ	0.85	0.83	0.92	0.79	0.50	0.59	0.40	0.39	4.57	78%	4.24	69%	148
	e2e	0.63	0.86	0.44	0.31	0.46	0.74	0.19	0.09	4.81	61%	4.69	59%	148

Table 2: Analysis of our questions on multiple splits by SRL argument types (Explicit, Implicit, and None / not present) over a sample of predicates in OntoNotes, G&C and ON5V. GRM+RC and ADQ+RC are the percentage of questions that were rated with GRM (resp. ADQ) ≥ 4 and also aligned to the correct description. RoleQ is our model, and e2e is the baseline. Onto* is the full OntoNotes set of 400 predicates.

[...] Jordan's King Abdullah II **pardoned** (JUSTICE.PARDON/pardon.01) *the former legislator known for her harsh criticism of the state* (DEFENDANT/A1) .
EEQ Who is the defendant? **RoleQ** Who did Jordan's King Abdullah II pardon?

[...] gun crime incidents are averaging about 29 a day in England and Wales, more than twice the level of when *the Labour Government*(ENTITY/A1) **came** (PERSONNEL.ELECT/come.01) to power in 1997.
EEQ Who voted? **RoleQ** What came?

About 160 workers at a factory that **made** *paper* (A1) for the Kent filters were exposed to asbestos in the 1950s.
Syn-QG What materials did a factory produce ? **RoleQ** What did 160 workers make?

But you **believe** *the fact that the U.S. Supreme Court just decided to hear this case is a partial victory for both Bush and Gore.* (A1)
Syn-QG What do you believe in ? **RoleQ** What is being believed?

Table 3: Examples of our role questions, Event Extraction Questions (Du and Cardie, 2020), and Syn-QG questions (Dhole and Manning, 2020)

overall, with average scores above 4, but the baseline scored even better on all datasets. However, the percentage of questions that were assigned both the correct role and high grammaticality/adequacy were significantly higher for our model (around 10–20% absolute). As we will see in the error analysis below, these results follow in part from the baseline model overfitting to natural-sounding questions for explicit arguments (which are easier to make grammatical due to an abundance of training data), even when they are not appropriate for the role. We also find that adequacy takes a hit for implicit or None roles, as our model has seen few such examples during training and since often the placeholder-filler arguments are also implicit for those instances.

Finding Implicit Arguments For explicit arguments in OntoNotes (69% of questions), annotators selected a mismatched answer in 9% of cases and marked the question unanswerable in 11%. For the non-explicit arguments, annotators quite often chose to answer the questions (50% of cases). 36% of these answers, which is 9% of all questions, were deemed to be plausible implicit arguments via manual inspection. For example, consider the sentence "It is only recently that the residents of Hsiachuotsu, unable to stand the smell, have begun to protest." Here annotators answered "the smell" when asked *Why did the residents of Hsiachuotsu protest?* (`A1`). Such implicit annotations could thus conceivably increase annotation coverage by about 10%, indicating that our Role Questions may be appealing as a way to help annotate implicit arguments.

Error analysis To understand the models' role correspondence errors, we check for cases where each model produced identical questions for different roles of the same predicate. 64% of predicate instances had at least one duplicate question under the baseline, as opposed to 6% for our model. Upon further examination, we found that the baseline systematically repeats questions for explicit arguments when prompted with a role whose argument is implicit or missing. For example, for the predicate `give.01`, in a context where only `A1` (thing given) was explicit (*the position was given*), it predicted *What was given?* for all core roles. This shows that our prototype generation stage is essential for handling these phenomena.

While RC accuracy is good for both explicit and implicit Role Questions, QA accuracy is lower on ON5V and G&C. On one hand, this may be due to the financial domain of G&C and the fact that it targets nominal predicates, making it harder

for annotators to understand. We also notice that the contextualizer sometimes fills the placeholders with the wrong argument from context, either because it is implicit or ambiguous. In such cases the annotators could mark the correct role, but do not answer the question properly.

In general, contextualization works well: The BART model is able to correctly identify tense, modality, negation and animacy in most cases. We inspected 50 randomly sampled instances of questions with average adequacy below 3, finding that the most common error is due to the placeholder being wrongly filled. Other errors are mainly due to an incorrect animacy judgment (*who* vs. *what*) or preposition or verb sense mistakes.

Modifiers We also evaluate results on 5 modifier roles for 100 predicate instances in OntoNotes. On these, grammaticality (4.20), adequacy (4.29), and role accuracy (81%) are comparable to results on core arguments, but QA accuracy (45%) is much lower. However, this number is not very reliable: of 500 modifier role questions, <10% corresponded to explicit arguments, because modifiers are relatively sparse in OntoNotes.

5.3 Comparison to Related Systems

To understand how our system fits in the landscape of existing work, we compare to two recently published question generation methods: Syn-QG (Dhole and Manning, 2020) and Event Extraction Questions (Du and Cardie, 2020, EEQ). These comparisons require some care, as the systems differ from ours in scope and inputs/outputs: Syn-QG only generates questions for arguments detected in the text, and EEQ uses fixed, template-based phrases for roles in an ontology of event types (rather than broad-coverage semantic roles).

Comparison to Syn-QG Syn-QG (Dhole and Manning, 2020) uses several techniques, including off-the-shelf syntax, SRL, and NER models, to identify potential answers in a sentence and generate questions which ask about them (examples in Table 3). Reusing their published code, we validate the model's output with the authors[9] and apply it to a sample of 100 sentences from OntoNotes. We run their model on these sentences and collect 143 QA pairs where the answer in Syn-QG overlaps significantly with a gold SRL argument, and assign the gold role label to the paired question. Then we use our system to generate questions for these role labels and evaluate both sets of questions according to our metrics.

	RA	QA	ADQ	GRM
RoleQ	0.85	0.75	4.43	4.49
SynQG	0.60	0.59	3.57	4.19

Table 4: Comparison between our (RoleQ) approach and Syn-QG (Dhole and Manning, 2020) on 100 random frames in OntoNotes, covering 143 core arguments. *RA* is Role Accuracy, *QA* is answer accuracy, *ADQ* is adequacy, and *GRM* is grammaticality.

The system's output is evaluated using our evaluation criteria, where results are shown in Table 4. Our model has better role accuracy (85%) than Syn-QG (60%), though this may be unsurprising since ours conditions on the gold role. Perhaps more surprisingly, our model also strongly wins on QA accuracy, with 75% to Syn-QG's 59%, despite Syn-QG conditioning directly on an answer that highly overlaps with the gold argument. Furthermore, our Role Questions are deemed more grammatical and adequate on average. These results suggest that our model has significant advantages over Syn-QG for fluently and correctly capturing aspects of semantic structure related to semantic roles.

Comparison to EEQ Du and Cardie (2020) generate questions by applying fixed templates to argument descriptions in the ACE (Doddington et al., 2004) event ontology, in order to facilitate the use of a QA system to extract arguments. For example (Table 3), the JUSTICE.PARDON event has a DEFENDANT argument which receives the question *Who is the defendant?* Event detection and extraction, in comparison to SRL, deals with more abstract events which may be expressed by a variety of predicates: for example, the PERSONNEL.ELECT event may be triggered by a phrase like "came to power" (Table 3, second row), where the verbal predicate is the more general `come.04`.[10] For comparison, we randomly extracted two arguments from ACE for each event and role covered by their questions, for 198 total arguments. Two of the authors then manually mapped the participant types to the corresponding PropBank roles of the predicate denoted by the annotated trigger word in

[9] Following their advice, we exclude questions generated from the templates for WordNet supersenses, as they were a source of noise.

[10] In PropBank, `come.04` includes such expressions as *come to fruition* and *come under scrutiny*.

	GRM	ADQ	QA	QA SQuAD
RoleQ	4.40	4.30	0.59	0.70
EEQ	3.98	3.72	0.57	0.56

Table 5: Grammaticality, adequacy, and QA scores for our Role Questions (RoleQ) and EEQ (Du and Cardie, 2020). We also report the QA Accuracy score of a SQuAD model.

ACE.[11] We then evaluated both sets of questions according to our metrics, with the exception of role accuracy, since the EEQ questions are not specific to the trigger word.

Results are shown in Table 5. Our questions score higher in grammaticality and adequacy, as EEQ's template-based approach often results in awkward or ungrammatical questions like *What declare bankruptcy?* (which, besides the subject agreement error, might need to ask *who* in order to be adequate, depending on the context). QA accuracy, on the other hand, is roughly comparable between the two systems for human annotators, showing that both do a similar job of capturing question semantics. However, we also measure QA accuracy with an automated QA model trained on SQuAD 1.0 (QA SQuAD, Table 5), and we find that our contextualized questions produce much higher QA accuracy. We suspect this is due to our contextualization step producing natural-sounding questions which are similar to those in other other QA datasets, aiding transfer.

6 Conclusion

We presented an approach to produce fluent natural language questions targeting any predicate and semantic role in the context of a passage. By leveraging the syntactic structure of QA-SRL questions in a two-stage approach, we overcome a lack of annotated data for implicit and missing arguments and produce questions which are highly specific to the desired roles. This enables the automatic generation of information-seeking questions covering a large, broad-coverage set of semantic relations, which can bring the benefits of QA-based representations to traditional SRL and information extraction tasks.

[11] For example, in the context of "coming to power," the ELECTED-ENTITY in ACE is mapped to COME.04-A0 in PropBank.

Acknowledgments

We would like to thank Daniela Brook-Weiss for helping in the initial stages of this project and the anonymous reviewers for their insightful comments. The work described herein was supported in part by grants from Intel Labs, Facebook, the Israel Science Foundation grant 1951/17. This project has received funding from the Europoean Research Council (ERC) under the Europoean Union's Horizon 2020 research and innovation programme, grant agreements No. 802774 (iEXTRACT) and No. 677352 (NLPRO).

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 475–484. ACM.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: Semantics of new predicate types. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3013–3019, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaustubh D Dhole and Christopher D Manning. 2020. Syn-qg: Syntactic and shallow semantic rules for question generation. *arXiv preprint arXiv:2004.08694*.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*,

pages 671–683, Online. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Hady Elsahar, Christophe Gravier, and Frederique Laforest. 2018. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 218–228, New Orleans, Louisiana. Association for Computational Linguistics.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-scale QA-SRL parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2060, Melbourne, Australia. Association for Computational Linguistics.

Michael Flor and Brian Riordan. 2018. A semantic role-based approach to open-domain automatic question generation. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 254–263, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Matthew Gerber and Joyce Chai. 2010. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden. Association for Computational Linguistics.

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 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, California. Association for Computational Linguistics.

Ayal Klein, Jonathan Mamou, Valentina Pyatkin, Daniela Stepanov, Hangfeng He, Dan Roth, Luke Zettlemoyer, and Ido Dagan. 2020. QANom: Question-answer driven SRL for nominalizations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3069–3083, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Karen Mazidi and Rodney D. Nielsen. 2014. Linguistic considerations in automatic question generation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 321–326, Baltimore, Maryland. Association for Computational Linguistics.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Julian Michael and Luke Zettlemoyer. 2021. Inducing semantic roles without syntax. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4427–4442, Online. Association for Computational Linguistics.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17–22.

Tatjana Moor, Michael Roth, and Anette Frank. 2013. Predicate-specific annotations for implicit role binding: Corpus annotation, data analysis and evaluation experiments. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 369–375, Potsdam, Germany. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. 2020. QADiscourse - Discourse Relations as QA Pairs: Representation, Crowdsourcing and Baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2804–2819, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, and Christopher D. Manning. 2020. Stay hungry, stay focused: Generating informative and specific questions in information-seeking conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 25–40, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2020. Controlled crowdsourcing for high-quality QA-SRL annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7008–7013, Online. Association for Computational Linguistics.

Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2009. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 106–111, Boulder, Colorado. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

R. Weischedel, E. Hovy, M. Marcus, and Martha Palmer. 2017. Ontonotes : A large training corpus for enhanced processing.

A Aligning QA-SRL to SRL roles

On the QA-SRL Bank 2.0, we aligned 190K QA-pairs with predicted SRL labels out of 271K gold questions. For QANom, we aligned 8.3K out of 21.5K gold questions. We use the verbal SRL parser in AllenNLP (Gardner et al., 2018) which re-implements Shi and Lin (2019)'s SRL parser. For nominal predicates, we re-train the same model on NomBank (Meyers et al., 2004), achieving 81.4 CoNLL-F1 score on the development set.

B Converting QA-SRL Questions to Prototypes

To transform a QA-SRL question into its prototype, we replace the AUX and VERB slot values with either *is* and the past participle form (for passive voice), a blank and the present form (for active voice when SUBJ is blank), or *does* and the stem form (for active voice when SUBJ is present). We also replace all occurences of *who* and *someone* with *what* or *something*. This effectively removes all modality, aspect, negation, and animacy information, while converting all questions to the simple present tense. However, it preserves the active/passive distinction and other elements of the question's syntactic structure, which are relevant to the semantic role.

C Contextualizing QA-SRL Questions

Here we provide extra details on the algorithm used to contextualize the questions provided by annotators in the QA-SRL Bank 2.0 and QANom.

C.1 Resolving Syntactic Ambiguity

As written in §4.2, we first try to choose the syntactic structure that is shared with the greatest number of other questions. If there is a tie (for example, if we only have one question for an instance), then we fall back to a few rules, depending on the type of ambiguity as follows:

- **Preposition/Particle:** in a question like *What does something give up?*, there is ambiguity over whether the object should be placed before or after the preposition (*up*). Here we default to after the preposition, as any time the object position before the preposition is valid (e.g., *something gives something up*), that means the preposition is acting as a particle, which generally admits the object after it as well (*something gives up something*).
- **Locative arguments:** QA-SRL allows the placeholder *somewhere* in the MISC slot. As a result, a question like *Where does something put something?* is ambiguous between treating *where* as denoting an adverbial (which would lead to the clause *Someone put something*) or a locative argument (which would result in *Someone put something somewhere*). We default to the adverbial interpretation.
- **Ditransitives:** the last type of ambiguity is for questions over ditransitive verbs like *What did someone give someone?* which are ambiguous over which of the two objects is extracted, *i.e.*, whether it should be *Someone gave something someone* or (*Someone gave someone something*). We default to resolving *who* questions to the first object and *what* questions to the second, to match English's tendency to put (generally animate) recipients/benefactors in the first object position and (generally inanimate) themes in the second.

C.2 Aligning Answers to Placeholders

After resolving ambiguities, every placeholder position in a QA-SRL question and every answer to a QA-SRL question is associated with a syntactic function (SUBJ for the SUBJ slot, OBJ for the OBJ slot, or PP/XCOMP/OBJ2/LOC for the PREP/MISC slots — see Table 1) and syntactic structure (which we may denote by the question's declarative form). To produce Frame-Aligned QA-SRL questions, we replace the placeholders in each question with any answers bearing the same syntactic function in the same syntactic structure (see Figure 4). For the purpose of encoding these syntactic structures, we ignore the same factors as we strip out in the question prototyping step (§4.1): tense, modality, negation, and animacy (but we keep the active/passive distinction).

To further increase coverage of our placeholder alignments, we add extra correspondences between syntactic structures which correspond to the same semantic role in many cases:

- We align the OBJ of a transitive clause (with a LOC or no MISC argument) with the SUBJ of passive clauses (with a LOC, *by*-PP, or no MISC argument).
- We align the SUBJ of a transitive clause (no MISC argument) with the prepositional object of a *by*-PP in a passive clause.

- We align the LOC argument of a transitive clause with the *where*-adverbial of a transitive clause with no MISC.

- Finally, we align any SUBJ arguments as long as their syntactic structures agree after stripping the PREP/MISC argument from both (and similarly for OBJ). For example, if we have *Who brought something? / John*, then *Who did someone bring something to?* would align with the answer to produce *Who did John bring something to?*

Without these extra correspondences, our method populates 83.7% of placeholders in the QA-SRL Bank 2.0 and QANom, while with them, we cover 91.8%.

Grammar Correction We add two extra post-processing steps to improve the quality of the questions. First, before substituting answers for placeholders, we attempt to undo sentence-initial capitalization by decapitalizing the first word of sentences in the source text if both the second character of the first word is lowercase (ruling out acronyms) and the first character of the second word is lowercase (ruling out many proper nouns). Second, we fix subject/verb agreement for questions with plural subjects (as QA-SRL questions always have singular agreement) by masking out the auxiliary verb (e.g., *does*) and replacing it with the form that receives higher probability under a masked language model (e.g., either *do* or *does*).

C.3 Training the Contextualizer

We fine-tune BART on the Frame-Aligned QA-SRL dataset for 3 epochs on 4 GeForce GTX 1080Ti GPUs with an effective batch size of 32 and maximum target sequence length of 20. We use the standard separator token between the question and the passage. We also surround the predicate token with text markers *PREDICATE-START* and *PREDICATE-END* (but without using new embeddings in the vocabulary), and insert the predicate lemma again right after the question. The full text input may look like: Some geologists PREDICATE-START study PREDICATE-END the Moon . ⟨/s⟩ study [SEP] what studies something ?, with the output question *Who studies the moon?*.

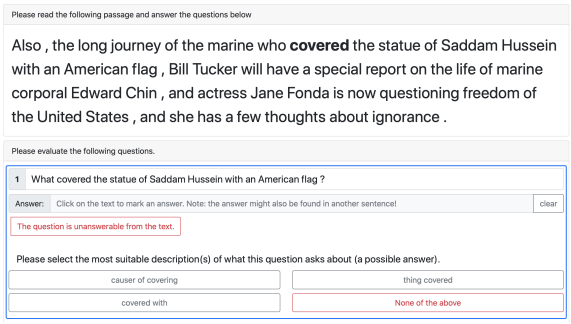

Figure 5: Interface for QA and RC annotation.

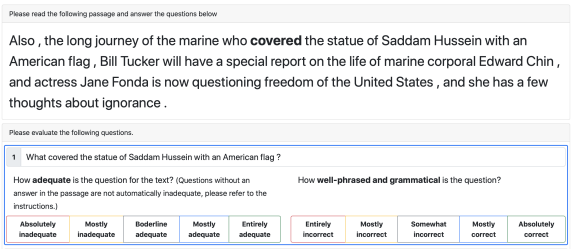

Figure 6: Interface for grammaticality and adequacy annotation.

D Annotation Interfaces

E Coverage

OntoNotes includes 530K core argument instances with 11K distinct roles. Our question lexicon contains prototypical questions for almost all arguments except 4K (<1%) instances. When filtering out questions with less than 50% SQuAD-F1 accuracy, the leftover prototypical questions cover 83.5% of all argument instances.