Syn-QG: Syntactic and Shallow Semantic Rules for Question Generation

Kaustubh D. Dhole
Amelia Science
RnD, IPsoft
New York, NY 10004
kdhole@ipsoft.com

Christopher D. Manning
Department of Computer Science
Stanford University
Stanford, CA 94305
manning@stanford.edu

Abstract

Question Generation (QG) is fundamentally a simple syntactic transformation; however, many aspects of semantics influence what questions are good to form. We implement this observation by developing Syn-QG, a set of transparent syntactic rules leveraging universal dependencies, shallow semantic parsing, lexical resources, and custom rules which transform declarative sentences into question-answer pairs. We utilize PropBank argument descriptions and VerbNet state predicates to incorporate shallow semantic content, which helps generate questions of a descriptive nature and produce inferential and semantically richer questions than existing systems. In order to improve syntactic fluency and eliminate grammatically incorrect questions, we employ back-translation over the output of these syntactic rules. A set of crowd-sourced evaluations shows that our system can generate a larger number of highly grammatical and relevant questions than previous QG systems and that back-translation drastically improves grammaticality at a slight cost of generating irrelevant questions.

1 Introduction

Automatic Question Generation (QG) is the task of generating question-answer pairs from a declarative sentence. It has direct use in education and generating engagement, where a system automatically generates questions about passages that someone has read. A more recent secondary use is for automatic generation of questions as a data augmentation approach for training Question Answering (QA) systems. QG was initially approached by syntactic rules for question-generation, followed by some form of statistical ranking of goodness, e.g., (Heilman and Smith, 2009, 2010). In recent years, as in most areas of NLP, the dominant approach has been neural network generation (Du et al., 2017),

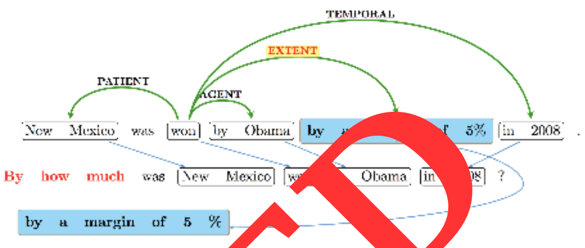

Figure 1: The SRL structure is leveraged to invoke a template, and a simple rearrangement of the modifying arguments is performed.

in particular using a sequence-to-sequence architecture which exploits the data in the rapidly growing number of large QA data sets.

Previous rule-based approaches suffer from a significant lack of variety in the questions they generate, sticking to a few simple and reliable syntactic transformation patterns. Neural architectures provide a pathway to solving this limitation since they can exploit QA datasets to learn the broad array of human question types, providing the usual neural network advantages of a data-exploiting, end-to-end trainable architecture. Nevertheless, we observe that the quality of current neural QG systems is still lacking: The generated questions lack syntactic fluency, and the models lack transparency and an easy way to improve them.

We argue that in essence QG can be governed by simple syntactic "question transformations" – while the implementation details vary, this is in accord with all major linguistic viewpoints, such as Construction Grammar and Chomskyan Generative Grammar, which emphasize grammatical rules and the existence of finite ways to create novel utterances. However, successful, fluent question generation requires more than just understanding syntactic question transformations, since felicitous questions must also observe various semantic and

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 752–765
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

pragmatic constraints. We approach these by making use of semantic role labelers (SRL), previously unexploited linguistic semantic resources like VerbNet's predicates (Figure 2) and PropBank's rolesets and custom rules like implications, allowing us to generate a broader range of questions of a descriptive and inferential nature. A simple transformation commonly used in rule-based QG is also displayed in Figure 1.

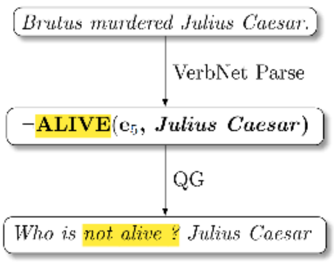

Figure 2: VerbNet Predicate Question Generation. Detailed intermediate steps are described in Figure 3.

We evaluate our QG framework, Syn-QG against three QG systems on a mixture of Wikipedia and commercial text sentences outperforming existing approaches in grammaticality and relevance in a crowd-sourced human evaluation while simultaneously generating more types of questions. We also notice that back-translated questions are grammatically superior but are sometimes slightly irrelevant as compared to their original counterparts. The Java code is publicly available at https://bitbucket.org/kaustubhdhole/syn-qg/.

2 Related Work

With the advent of large-scale QA datasets (Rajpurkar et al., 2016; Nguyen et al., 2016), recent work in QG (Du et al., 2017; Zhou et al., 2017) has primarily focused on training sequence-to-sequence and attention-based architectures. Dong et al. (2019) fine-tuned the question generation task by taking advantage of a large pre-trained language model. Success in reinforcement learning has inspired teacher-student frameworks (Wang et al., 2017; Tang et al., 2017) treating QA and QG as complementary tasks and performing joint training by using results from QA as rewards for the QG task. Yuan et al. (2017); Hosking and Riedel (2019); Zhang and Bansal (2019) used evaluation metrics like BLEU, sentence perplexity, and QA probability as rewards for dealing with exposure bias.

Chen et al. (2019) trained a reinforcement learning based graph-to-sequence architecture by embedding the passage via a novel gated bi-directional graph neural network and generating the question via a recurrent neural network. To estimate the positions of copied words, Liu et al. (2019) used a graph convolution network and convolved over the nodes of the dependency parse of the passage. Li et al. (2019) jointly modeled OpenIE relations along with the passage using a gated-attention mechanism and a dual copy mechanism.

Traditionally, question generation has been tackled by numerous rule-based approaches (Heilman and Smith, 2009; Mostow and Chen, 2009; Yao and Zhang, 2010; Lindberg et al., 2013; Labutov et al., 2015). Heilman and Smith (2009, 2010) introduced an overgenerate-and-rank approach that generated multiple questions via rule-based tree transformations of the constituency parse of a declarative sentence and then ranked them using a logistic-regression ranker with manually designed features. Yao and Zhang (2010) described transformations of Minimal Recursion Semantics representations guaranteeing grammaticality. Other transformations have been in the past defined in terms of templates (Mazidi and Nielsen, 2014, 2015; Mazidi and Tarau, 2016; Flor and Riordan, 2018), or explicitly performed (Heilman and Smith, 2009) by searching tree patterns via Tregex, followed by their manipulation using Tsurgeon (Levy and Andrew, 2006). Kurdi et al. (2020) provide a comprehensive summary of QG, analysing and comparing approaches before and after 2014.

Vis-à-vis current neural question generators, rule-based architectures are highly transparent, easily extensible, and generate well-formed questions since they perform clearly defined syntactic transformations like subject-auxiliary inversion and *WH-movement* over parse structures whilst leveraging fundamental NLP annotations like named entities, co-reference, temporal entities, etc.

However, most of the existing rule-based systems have lacked diversity, being mostly focused on generating *What*-type and boolean questions and have mainly exploited parse structures which are not semantically informed. Mazidi and Tarau (2016); Flor and Riordan (2018) use Dependency, SRL, and NER templates but do not handle modalities and negation in a robust manner. Moreover, there is plenty of availability of core linguistic resources like VerbNet and PropBank, which provide further unique ways to look at sentences and ask questions differently besides the generally well-established dependency and SRL parses.

3 Syn-QG

Syn-QG is a rule-based framework which generates questions by identifying potential short answers in 1) the nodes of crucial dependency relations 2) the modifying arguments of each predicate in the form of semantic roles 3) named entities and other generic entities 4) the states of VerbNet's thematic roles in the form of semantic predicates and 5) PropBank roleset specific natural language descriptions. Each of the five heuristics works independently, generating a combined set of question-answer pairs, which are eventually back-translated. We describe each of these five sources.

3.1 Dependency Heuristics

Dependency trees are syntactic tree structures, wherein syntactic units in the form of words are connected via directed links. The finite verb is considered as the structural root of the tree, and all other syntactic units are either directly (*nsubj, dobj,* etc.) or indirectly (*xcomp, iobj,* etc.) dependent on this finite verb.

We present rules over such dependency trees annotated according to the Universal Dependencies (UD) format (de Marneffe et al., 2014). To extract dependency structures, we use the parser of Gardner et al. (2018).

We make use of PropBank's predicate-argument structure (SRL) for clausal extraction of the verb headed by a select few dependency nodes which can serve as answers. These rules treat the clause as a combination of a subject, an object, the head verb and other non-core arguments. The clause is further refined with modals, auxiliaries and negations if found around the verb. Finally, we make use of a set of predefined handwritten templates, a few of which are described in Table 1.

In each of the templates, we convert *What* to *Who/Whom*, *When* or *Where* depending on the named entity of the potential answer and *do* to *does* or *did* according to the tense and number of the subject to ensure subject-verb agreement. The pseudo code is described in Algorithm 2 of the Appendix.

3.2 SRL Heuristics

While dependency representations are perhaps the most popular syntactic method for automatically extracting relationships between words, they lack sufficient semantic detail. Being able to answer *"Who* did *what* to *whom* and *how, why, when* and *where"* has been a central focus in understanding language. In recent decades, shallow semantic parsing has been a prominent choice in understanding these relationships and has been extensively used in question generation (Mazidi and Tarau, 2016; Flor and Riordan, 2018).

PropBank-style frames provide semantically motivated roles that arguments around a verb play. Moreover, highly accurate semantic role labeling models are being developed owing to corpora like PropBank and FrameNet. We take advantage of the SRL model of Gardner et al. (2018) for extracting the roles of each verb in the sentence.

Algorithm 1 SRL Heuristics

$\{SRL_1 \dots SRL_s\} \leftarrow SRL(w_0 \dots w_n)$
loop j = 0, until j = s:
if SRL_j *contains* A_0 or A_1 *and at least* 1 A_m
then
  $\{A_0 \dots A_{CAU}, A_{TMP}\} \leftarrow SRL_j$
  loop $A_x \in SRL_j\ if\ A_x = modifier$:
    $subj \leftarrow A_0$
    $A_x^- \leftarrow \sum(A_3, A_4, ...A_{TMP} - A_x)$
    $verb \leftarrow \{A_v, modals, negation\}$
    $template \leftarrow modifier_{type} \leftarrow A_x$
    $QA \leftarrow template(subj, A_x, verb, A_x^-)$
    close;

We succinctly describe the steps taken in Algorithm 1. We first filter out all the predicates which have an *Agent* or a *Patient* and at least one other modifier like *Extent, Manner, Direction,* etc. These modifiers would serve as our short answers. We make use of a set of predefined handwritten templates described in Table 2, which rearrange the arguments within the fact to convert it into an interrogative statement depending on the modifier.

In Figure 1, the predicate "won" is modified by a *Patient* "New Mexico", an *Agent* "Obama", an *Extent* modifier "by a margin of 5%" and a *Temporal* modifier "in 2008". For *Extent* as a short answer, we fill a pre-defined template "By how much mainAux nsubj otherAux verb obj modifiers ?" to get the above question-answer pair. We keep the order of arguments as they appear in the original

Potential Short Answer (Dependencies)	Question Template	Sample Fact	Generated Question
subject (nsubj)	**Wh** mainAux otherAux verb obj modifiers?	Ricky Ponting accepted captaincy during Australia's golden era.	**Who** accepted captaincy during Australia's golden era?
direct object(dobj)	**Wh** mainAux nsubj otherAux verb modifiers?	In monsoon, India receives large amounts of rain that can cause flooding.	**What** does India receive in monsoon?
open clausal complement (xcomp)	**Wh** mainAux nsubj verb modifiers?	The Sheriff did not try to eat the apples while the outlaws were fasting.	**What** did the Sheriff not try while the outlaws were fasting?
copula (cop)	**How would you describe** nsubj?	Comets are leftovers from the creation of our solar system about 4.5 billion years ago.	**How would you describe** comets ?

Table 1: A few templates to describe the construction of questions. Different word units are shown in unique colors to describe the filling of the template. All the short answers are highlighted in blue.

sentence. The templates are described in Table 2.

3.3 Named Entities, Custom Entities, and Hypernyms

We create separate templates when any numbered SRL argument contains common named entities like *Person*, *Location*, *Organization* etc. Like Flor and Riordan (2018), we add specific rules in the form of regexes to address special cases to differentiate between phrases like ***For how long*** and ***Till when*** instead of a generic ***When*** question type. Some of the templates are described in Table 7 in the Appendix. The approach is described in Algorithm 3 in the Appendix.

We also use WordNet (Miller, 1998) hypernyms of all potential short answers and replace ***What*** with the bigram ***Which hypernym***. So, for a sentence like "Hermione plays badminton at the venue", we generate a question "***Which sport*** does Hermione play at the venue?". For computing the hypernym, we use the sense disambiguation implementation of Tan (2014). While supersenses do display a richer lexical variety, sense definitions don't always fit well.

3.4 Handling modals and auxilliaries

During explicit inversion of the verb and arguments around it via our templates, we tried to ensure that the positions of auxiliaries are set, and negations are correctly treated. We define a few simple rules to ensure that.

- When there are multiple auxiliaries, we only invert the first auxiliary while the second and further auxiliaries remain as they are just before the main verb.
- We make the question auxiliary finite and agree with the subject.
- We ensure that the object is kept immediately after the verb.

For passive cases, *subj-verb-obj* is changed to *obj-verb-by-subj*.

3.5 Handling Factualness via Implicature

Previous rule-based approaches (Mazidi and Tarau, 2016; Flor and Riordan, 2018) have used the NEG dependency label to identify polarity. But such an approach would suffer whenever polarities would be hierarchically entailed from their parent clauses in cases like "Picard did not fail to X" where the entailed polarity of "X" is, in fact, positive. Moreover, in one-way implications like "Bojack hesitated to X", it would be best not to generate a question for unsure cases since it is open-ended if Bojack did or did not X. A similar example is displayed in Figure 5. For each verb representing a subordinate clause, we compute its entailed truth or falsity from its parent clause using the set of one-way and two-way implicative verbs, and verb-noun collocations provided by Karttunen (2012). For example, the two-way implicative construction "forget to X" entails that "X" did not happen, so it would be wrong to ask questions about "X". Karttunen (2012) provides simple implications in the form of 92 verbs and phrasal implications in the form of 9 sets of verbs and 8 sets of nouns making 1002 verb-noun collocations. The entailed polarity of a

Potential Short Answer (Verb Arguments)	Question Template	Sample Fact	Generated Question
Locative (LOC)	**Where** mainAux nsubj otherAux verb obj modifiers ?	Americans eat about 100 acres of pizza each day, with about 3 billion pizzas sold annually in the USA.	**Where** do about 3 billion pizzas sell annually ?
Manner (MNR)	**How** mainAux nsubj otherAux verb obj modifiers ?	Young Sheldon was caught unaware as the liquid was oozing out of the chamber in a zig-zag fashion.	**How** was the liquid oozing out of the chamber?
Purpose (PNC and PRP)	**For what purpose** mainAux nsubj otherAux verb obj modifiers ?	Collectively, South African women and children walk a daily distance equivalent to 16 trips to the moon and back to fetch water.	**For what purpose** do South African women and children walk a daily distance equivalent to 16 trips to the moon and back collectively ?
Cause (CAU)	**Why** mainAux nsubj otherAux verb obj modifiers ?	Since the average faucet releases 2 gallons of water per minute, you can save up to four gallons of water every morning by turning off the tap while you brush your teeth.	**Why** can you save up to four gallons of water by turning off the tap while you brush your teeth every morning ?
Temporal (TMP)	**When** mainAux nsubj otherAux verb obj modifiers ?	Stephen Hawking once in June 2009 threw a party for time-travelers but he announced the party the next day.	**When** did Stephen Hawking throw a party for time - travelers ? **When** did Stephen Hawking announce the party ?
	Till when mainAux nsubj otherAux verb obj modifiers?	Princess Sita traveled the whole town until the end of summer.	**Till when** did Princess Sita travel the whole town?
Extent (EXT)	**By how much** mainAux nsubj otherAux verb obj modifiers ?	New Mexico was won by Obama by a margin of 5% in 2008.	**By how much** was New Mexico won by Obama in 2008?

Table 2: The templates of temporal, direction, extent, etc. are leveraged to ask questions about different modifiers. Answer fragments are highlighted in blue. In passive cases like the last example, we change the template order from subj-verb-obj to obj-verb-by-subj.

clause can be either TRUE, FALSE, or UNSURE[1]. For FALSE clauses, we only generate a boolean question with a NO answer. For UNSURE clauses, we do not generate any question. For TRUE clauses and verbs and constructions not present in the above set, we rely on the NEG label.

3.6 VerbNet Predicate Templates

While SRL's event-based representations have permitted us to generate questions that talk about the roles participants of an event play, we exploit VerbNet's sub-event representation to ask questions on how participants' states change across the time frame of the event. In Figure 2, the event murder (VerbNet class *murder-42.1*) results in a final state in which the participant *Julius Caesar* is in a *not-alive* state.

Each class in VerbNet (Schuler, 2005; Brown et al., 2019) includes a set of member verbs, the thematic roles used in the predicate-argument structure, accompanied with flat syntactic patterns and their corresponding semantic predicates represented in neo-Davidsonian first-order-logic formulation. These semantic predicates bring forth a temporal sequencing of sub-events tracking how participants' states change over the course of the event. The advantage is to be able to ask questions

[1]Unsure clauses appear in one-way implicatives when it's unclear if the clause is true or false under either an affirmative or a negative parent clause.

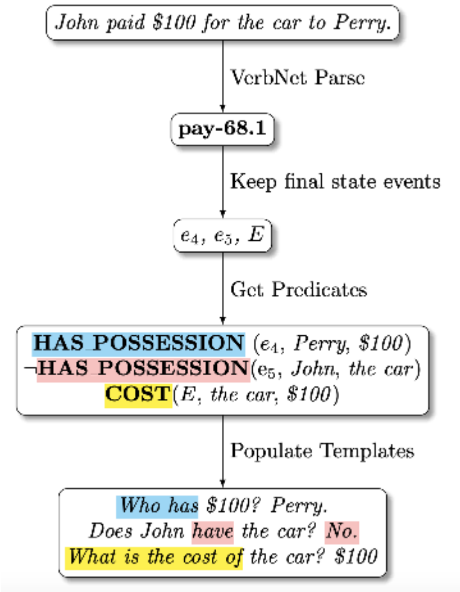

Figure 3: VerbNet Predicate Question Generation. All the predicates of the two sub-events e_4 and e_5 (HAS_POSSESSION) would be considered since e_3 possesses a process-oriented predicate TRANSFER. COST is the predicate of the main event E.

bearing a surface form different from the source sentence but which are driven by reasoning rather than just being paraphrastic. For example, in the sentence, "Brutus murdered Julius Caesar", the event *murder-42.1* entails a final state of "death" or the *Patient* participant not being alive at the end of the event. So, we construct a template "mainAux the Patient otherAux not alive?". Similarly, the event *pay-68-1* results in a final state in which the *Recipient* "Perry" has possession of "$100" and the *Agent* "John" has possession of "the car", against which we define the templates as shown in Figure 3.

We formulate two sets of questions: boolean type and wh-type questions asking specifically about these states. We create templates for VerbNet's stateful predicates like **has_location**, **has_possession**, **has_information**, **seem**, **has_state**, **cost**, **desire**, **harmed**, **has_organization_role**, **together**, **social_interaction**, **authority_relationship**, etc. which are present in 64.4% of the member verbs in VerbNet[2]. We outline a few of the templates in Table 3.

During inference time, we first compute the VerbNet sense, the associated thematic role mapping, and syntactic frame (along with the predicates) with the help of Brown et al. (2019)'s parser. VerbNet's predicates are governed by the sub-events in which they occur. Although VerbNet's representation lays out a sequence of sub-events, no sub-event is explicitly mentioned as the final one[3]. We choose all the predicates of those sub-events which are preceded by other sub-events which possess at least one process-oriented predicate.[4]

3.7 PropBank Argument Descriptions

PropBank rolesets' course-grained annotation of verb-specific argument definitions ("killer", "payer", etc.) to represent semantic roles offers robustly specific natural language descriptions to ask questions about the exact roles participants play. Nonetheless, not all descriptions are suitable to be utilized directly in rigid templates. So we incorporate back-translation to 1) get rid of grammatical errors propagated from incorrect parsing and template restrictions, and 2) eliminate rarely used Prop-Bank descriptions and generate highly probable questions.

While previous work in rule-based QG has used SRL templates and WordNet senses to describe the roles arguments around a verb play, previous SRL templates have always been verb-agnostic, and we believe there is a great deal of potential in PropBank descriptions. Moreover, WordNet supersenses do not always give rise to acceptable questions. On manual evaluation, question relevance decreased after incorporating templates with WordNet supersenses. Instead, we make use of PropBank's verb-specific natural language argument descriptions to create an additional set of templates. VerbNet senses have a one-to-one mapping with PropBank rolesets via the SemLink project (Palmer, 2009). We hence make use of Brown et al. (2019)'s parser to find the appropriate PropBank roleset for a sentence.

However, we observed that a lot of PropBank descriptions were noisy and made use of phrases which would be unarguably rare in ordinary parlance like "breather" or "truster". To eliminate such descriptions, we computed the mean Google N-gram probabilities (Lin et al., 2012) of all the PropBank phrases in the timespan of the last 100

[2] Out of 4854 member verbs, there are 3128 members whose syntactic frame contains at least one of these predicates.

[3] or a sub-event, which is an outcome of a process

[4] Out of 174 VerbNet predicates, we manually categorize 84 predicates like HAS_LOCATION, HAS_POSSESSION as stateful predicates and the remaining ones like DESCRIBE, TRANSFER, etc. as process-oriented predicates.

Triggering Predicate and Thematic Arguments	Question Template	Sample Fact & VerbNet Predicate	Generated Question
HAS_POSSESSION (Asset,Recipient)	**Who** has Asset ? Recipient	Robert paid $100 to Mary for the cycle. **HAS_POSSESSION**(Mary,$100)	**Who** has $100 ? Mary
HARMED (Patient)	**What** is harmed ? Patient	The terrorists bombed the building. **HARMED**(the building)	**What** is harmed ? the building
NOT_ALIVE (Patient)	Is Patient alive ? No.	According to epics, Vishnu killed the demon Kaitabh. **NOT_ALIVE** (the demon Kaitabh)	Is the demon Kaitabh alive ? No.

Table 3: VerbNet predicate templates (simplified) along with sample questions with the thematic roles highlighted. A question is created from the concept of "being alive" which is not synonymous with but is an outcome of "killing".

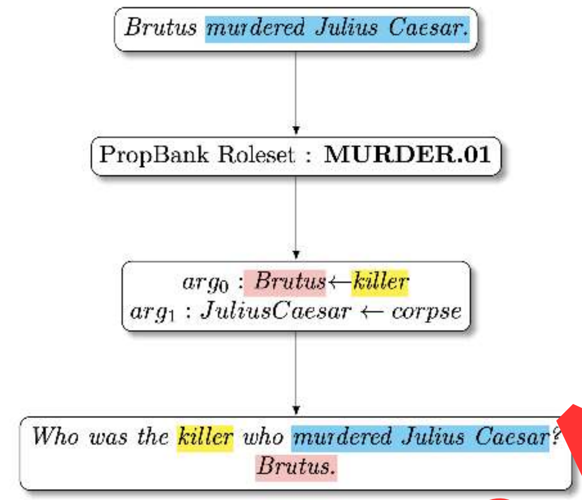

Figure 4: Here, "killer" is the natural language description of "Brutus" in the MURDER.01 roleset.

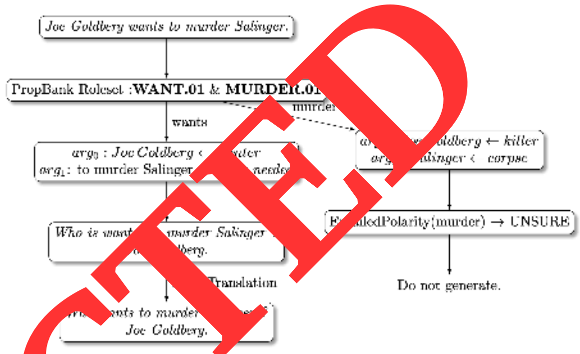

Figure 5: Back-translation and Implicature. Since the entailment polarity of "murder" is unsure, no questions are generated.

years and kept only those phrases which ranked in the top 50%.

3.8 Back-Translation

Back-translation has been used quite often in grammatical error correction (Xie et al., 2018) and is well known to translate noisy and ungrammatical sentences to their cleaner high probability counterparts. We exploit this observation to clean questions with noisy and inconsistent PropBank descriptions like "wanter" (Figure 5). We use two state-of-the-art (SOTA) pre-trained transformer models `transformer.wmt19.en-de` and `transformer.wmt19.de-en` from Ott et al. (2019) trained on the English-German and German-English translation tasks of WMT 2019.

Figure 6 in the Appendix shows the output of all the five sets of templates applied together over one sentence (along-with implicature).

4 Evaluation and Results

Most of the prior QG studies have evaluated the performance of the generated questions using automatic evaluation metrics used in the machine translation literature. We use the traditional BLEU scores (Papineni et al., 2002) and compare the performance of Syn-QG on the SQuAD (Rajpurkar et al., 2016) test split created by Zhou et al. (2017). BLEU measures the average n-gram precision on a set of reference sentences. A question lexically and syntactically similar to a human question would score high on such n-gram metrics. Despite not utilizing any training data, Syn-QG performs better than the previous SOTA on two evaluation metrics BLEU-3 and BLEU-4 and close to SOTA on BLEU-1 and BLEU-2 (Table 4) at the time of submission. The high scores obtained without conducting any training arguably shed a little light on the predictable nature of the SQuAD dataset too.

Besides SRL, Dependency, and NER templates,

Architecture	BLEU-1	BLEU-2	BLEU-3	BLEU-4
PCFG-Trans (Heilman and Smith, 2010)	28.77	17.81	12.64	9.47
SeqCopyNet (Zhou et al., 2018)				13.02
NQG++ (Zhou et al., 2017)	42.36	26.33	18.46	13.51
MPQG (Song et al., 2017)				13.91
Answer-focused Position-aware model (Sun et al., 2018)	43.02	28.14	20.51	15.64
To the Point Context (Li et al., 2019)	44.40	29.48	21.54	16.37
s2sa-at-mp-gsa (Zhao et al., 2018)	44.51	29.07	21.06	15.82
ASs2s (Kim et al., 2019)				16.17
CGC-QG (Liu et al., 2019)	46.58	30.9	22.82	17.55
Capturing Greater Context (Tuan et al., 2019)	46.60	**31.94**	23.44	17.76
Natural QG with RL based Graph-to-Sequence (Chen et al., 2019)	-	-	-	17.94
RefineNet (Nema et al., 2019)	**47.27**	31.88	23.65	18.16
QPP&QAP (Zhang and Bansal, 2019)	-	-	-	18.37
ACS-QG* (Liu et al., 2020)	**52.30***	**36.70***	**28.00***	22.05
UNILM* (Wang et al., 2020)	-	-	-	24.32
ERNIE-GEN* (Xiao et al., 2020)	-	-	-	25.57
UNILMv2* (Bao et al., 2020)	-	-	-	26.30
ProphetNet* (Yan et al., 2020)	-	-	-	**26.72***
Syn-QG	45.55	30.24	**23.84**	**18.72**

Table 4: Automatic Evaluation Results on SQuAD of different QG models. PCFG-TRANS and Syn-QG are two rule-based models. *Work contemporaneous with or subsequent to the submission of this paper.

System	#Questions Generated	Avg. #Questions Per Sentence	Grammaticality	Relevance
H&S	381	3.81	3.4[illegible]	4.23
NQG	100	1	[illegible]8	3.28
QPP&QAP	—	—	3.[illegible]	4.03
Syn-QG	654	**6.54**	**3.93**	**4.34**

Table 5: Comparison of human evaluation with H&S (Heilman and Smith, 2009), NQG (Du et al., 2017) and QPP&QAP (Zhang and Bansal, 2019)

System	Avg. novel unigrams	Avg. novel bigrams	Avg. novel trigrams
H&S	23.6	[illegible].64	52.22
Syn-QG (w/o BT)	26.8	43.93	53.4
Syn-QG	39.34	64.08	76.24
SQUAD	42.[illegible]	74.2	86.35
Syn-QG (BT vs w/o-BT)	28.7[illegible]	55.18	67.81

Table 6: The percentage of n-grams of the generated questions which are not present in the source sentence. The last row indicates the percentage of n-grams not present in the non-backtranslated questions.

Syn-QG's questions also arise from VerbNet's predicates and PropBank's descriptions, which indeed by nature describe events not mentioned explicitly within the fact. [illegible] in Figure 2, the sentence with the event "p[illegible]d" results in a question with a stateful event of "c[illegible]". Deducible questions like these have a good chance of having a distribution of n-grams quite different from the source sentences, possibly exposing the weakness of traditional n-gram metrics and rendering them less useful for a task like QG.

In order to have a complete and more reliable evaluation to gauge the system, we also carry out a human evaluation using two of the metrics used in QG-STEC Task B (Rus et al., 2012), namely grammaticality, and relevance which we define below. We compared the questions generated from our system against the constituency-based H&S (Heilman and Smith, 2009), a neural system NQG (Du et al., 2017) which does not depend on a separate answer extractor and QPP&QAP[5] (Zhang and Bansal, 2019) which has outperformed existing methods. We fed a total of 100 facts randomly picked from Wikipedia and 5 commercial domains (IT, Healthcare, Sports, Banking and Politics) combined, to each of the four systems. We then conducted a crowd-sourced evaluation over Amazon Mechanical Turk for the generated questions.

- **Grammatical Correctness**: Raters had to rate a question on how grammatically correct

[5] Since the QPP&QAP model does not have a separate answer extractor, we use the answer spans computed from Syn-QG (412 in total after discarding overlaps).

it is or how syntactically fluent it is, disregarding its underlying meaning.

- **Relevance Score**: Raters had to give a score on how relevant the generated question is to the given fact. The relevance score helps us gauge whether the question should have been generated or not irrespective of its grammaticality.[6]

Each question was evaluated by three people scoring grammaticality and relevance on a 5 point Likert scale. The inter-rater agreement (Krippendorff's co-efficient) among human evaluations was 0.72. The instructions given to the Mturk raters are provided in the Appendix Figure 7. The results of the evaluation are shown in Table 5. Syn-QG generates a larger number of questions than H&S and performs strongly on grammaticality ratings. Syn-QG is also able to generate highly relevant questions without the use of a ranker. Also, rule-based approaches seem to be much better at generating relevant questions than neural ones.

QG-STEC also used variety and question types as their evaluation criteria and rewarded systems to generate questions meeting a range of specific question types. Syn-QG's questions cover each of those question types.

Since many times, despite the ability to paraphrase (Table 6), back-translated outputs tend to change the meaning of the original sentence, we also measured back-translation's impact on the above QG metrics. We considered questions generated from 50 facts of Wikipedia measuring the grammaticality and relevance before and after back-translation. While grammaticality increased from 3.54 to 4.11, question relevance fell a bit from 3.96 to 3.88. This observation, along with the performance of QPP&QAP shown in Table 4, accentuates that while neural models are learning syntactic structures well, there is still some progress to be made to generate relevant questions.

5 Discussion

We introduced Syn-QG, a set of broad coverage rules leveraging event-based and sub-event based sentence views along with verb-specific argument descriptions. Automatic and manual evaluations show that Syn-QG is able to generate a large number of diverse and highly relevant questions with better fluency. Verb-focused rules help approach long-distance dependencies and reduce the need for explicit sentence simplification by breaking down a sentence into clauses while custom rules like implications serve a purpose similar to a re-ranker to discard irrelevant questions but with increased determinism. While our work focuses on sentence-level QG, it would be interesting to see how questions generated from VerbNet predicates would have an impact on multi-sentence or passage level QG, where the verb-agnostic states of the participants would change as a function of multiple verbs. The larger goal of QG is currently far from being solved. Understanding abstract representations, leveraging world knowledge, and reasoning about them is crucial. However, we believe that with an extensible and transparent architecture, it is very much possible to keep improving the system continuously in order to achieve this larger goal.

[6] In cases when the grammaticality is extremely low like 1 or 2, the relevance score will also tend to be low. Otherwise, we assume that minor grammatical variations can be ignored while gauging relevance.

Acknowledgments

We thank the three anonymous reviewers for their helpful comments and invaluable suggestions. We also thank the members of Amelia Science, RnD IPsoft, India - Manjunath Hegde, Anant Khandelwal, Ashish Shrivastava for their work in QG and especially Viswa Teja Ravi, for helping in replicating Mazidi and Tarau (2016)'s work. We also thank Uday Chinta and IPsoft, India, for supporting and providing access to Amazon Mechanical Turk.

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804*.

Susan Windisch Brown, Julia Bonn, James Gung, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2019. VerbNet representations: Subevent semantics for transfer verbs. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 154–163.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Natural question generation with reinforcement learning based graph-to-sequence model. *arXiv preprint arXiv:1910.08832*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming

Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

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.

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.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Michael Heilman and Noah A Smith. 2009. Question generation via overgenerating transformations and ranking. Technical Report CMU-LTI-09-013, Language Technologies Institute, Carnegie Mellon University.

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. Association for Computational Linguistics.

Tom Hosking and Sebastian Riedel. 2019. Evaluating rewards for question generation models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

Lauri Karttunen. 2012. Simple and phrasal implicatives. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 124–131. Association for Computational Linguistics.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6602–6609.

Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1):121–204.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 889–898.

Roger Levy and Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *LREC*, pages 2231–2234. Citeseer.

Jingjing Li, Yifan Gao, Lidong Bing, Irwin King, and Michael R Lyu. 2019. Improving question generation with to the point context. *arXiv preprint arXiv:1910.06036*.

Yuri Lin, Jean-Baptiste Michel, Erez Lieberman Aiden, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the Google books ngram corpus. In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174. Association for Computational Linguistics.

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. In *Proceedings of The Web Conference 2020*, pages 2032–2043.

Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. 2019. Learning to generate questions by learning what not to generate. *arXiv preprint arXiv:1902.10418*.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 4585–4592, Reykjavik, Iceland. European Languages Resources Association (ELRA).

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)*, volume 2, pages 321–326.

Karen Mazidi and Rodney D Nielsen. 2015. Leveraging multiple views of text for automatic question generation. In *International Conference on Artificial Intelligence in Education*, pages 257–266. Springer.

Karen Mazidi and Paul Tarau. 2016. Infusing NLU into automatic question generation. In *Proceedings of the 9th International Natural Language Generation conference*, pages 51–60.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Jack Mostow and Wei Chen. 2009. Generating instruction automatically for the reading strategy of self-questioning. In *AIED*, pages 465–472.

Preksha Nema, Akash Kumar Mohankumar, Mitesh M Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2019. Let's ask again: Refine network for automatic question generation. *arXiv preprint arXiv:1909.05355*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Martha Palmer. 2009. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the generative lexicon conference*, pages 9–15. GenLex-09, Pisa, Italy.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2012. A detailed account of the first question generation shared task evaluation challenge. *Dialogue & Discourse*, 3(2):177–204.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania.

Linfeng Song, Zhiguo Wang, and Wael Hamza. 2017. A unified query-based generative model for question generation and question answering. *arXiv preprint arXiv:1709.01058*.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939.

Liling Tan. 2014. Pywsd: Python implementations of word sense disambiguation (wsd) technologies [software]. https://github.com/alvations/pywsd.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.

Luu Anh Tuan, Darsh J Shah, and Regina Barzilay. 2019. Capturing greater context for question generation. *arXiv preprint arXiv:1910.10274*.

Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. *arXiv preprint arXiv:1706.01450*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*.

Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. *arXiv preprint arXiv:2001.11314*.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Y Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. 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 619–628.

Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*.

Xuchen Yao and Yi Zhang. 2010. Question generation with minimal recursion semantics. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 68–75. Citeseer.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on*

Empirical Methods in Natural Language Processing, pages 3901–3910.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2018. Sequential copying networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

A Appendices

Algorithm 2 Dependency Heuristics

$\{d_0 \dots d_n\} \leftarrow dependency(w_0 \dots w_n)$
loop i = 0, until i = n:
if *parent*$(d_i)! =$ *null* **then**
  $d_v \leftarrow parent(d_i)$
  $\{A_0 \dots A_{CAU}\} \leftarrow SRL(d_v)$
  $subj \leftarrow A_0$
  if $d_i \in A_1$ **then**
    $obj \leftarrow A_1$
  else
    $obj \leftarrow A_2$
  $A_x \leftarrow \sum(A_3, A_4, ...A_{TMP})$
  $verb \leftarrow \{d_v, modals, negation\}$
  $template \leftarrow dep_{type} \leftarrow d_i$
  $QA \leftarrow template(subj, obj, verb, A_x)$
  close;

Algorithm 3 Named Entity Heuristics

$\{SRL_1 \dots SRL_s\} \leftarrow SRL(w_0 \dots w_n)$
loop j = 0, until j = s:
if SRL_j *contains* A_0 or A_1 *and at least* 1 A_m **then**
  $\{A_0 \dots A_{CAU}, A_{TMP}\} \leftarrow SRL_j$
loop $A_x \in SRL_j$ *if* A_x *contains a* NE:
  $subj \leftarrow$ [illegible]
  $A_x^- \leftarrow \sum(A_3, A_4, ...A_{TMP} - A_x)$
  $verb \leftarrow \{A_v, modals, negation\}$
  $template \leftarrow NE_{type} \leftarrow A_x$
  $QA \leftarrow template(subj, A_x, verb, A_x^-)$
  close;

Potential Short Answer (Named Entities)	Question Template	Sample Fact	Generated Question
Location	**Where** mainAux subj otherAux verb obj modifiers ?	The event was organized at Times Square.	**Where** was the event organized?
Person	**Who** mainAux subj otherAux verb obj modifiers ? **Whom** mainAux obj otherAux verb modifiers	WestWorld brought back the life of the roboticist Craig Smith.	**Whom** did WestWorld bring back the life of?
Date	**When** mainAux subj otherAux verb obj modifiers ?	Donald Trump won the elections in the year 2016	**When** did Donald Trump win the elections?
Number	**How many** mainAux subj otherAux verb obj modifiers?	A thousand will not be enough for the event.	**How many** will not be enough for the event?
Phone Number	**At what number** mainAux subj otherAux verb obj modifiers ?	The pizza guy can be reached at +91-748-728-781	**At which phone number** can the pizza guy be reached?
Duration	**For how long** mainAux subj otherAux verb obj modifiers?	Lauren would be staying in the hut for around 10 minutes.	**For how long** would Lauren be staying at the hut?
Organization	**Which organization** mainAux subj otherAux verb obj modifiers?	Deepak joined the big firm, the United Nations.	**Which organization** did Deepak join?

Table 7: SRL arguments which contain a named entity are fully considered as a short answer "for around 10 minutes" rather than only the named entity span "10 minutes". SRL arguments are highlighted in blue.

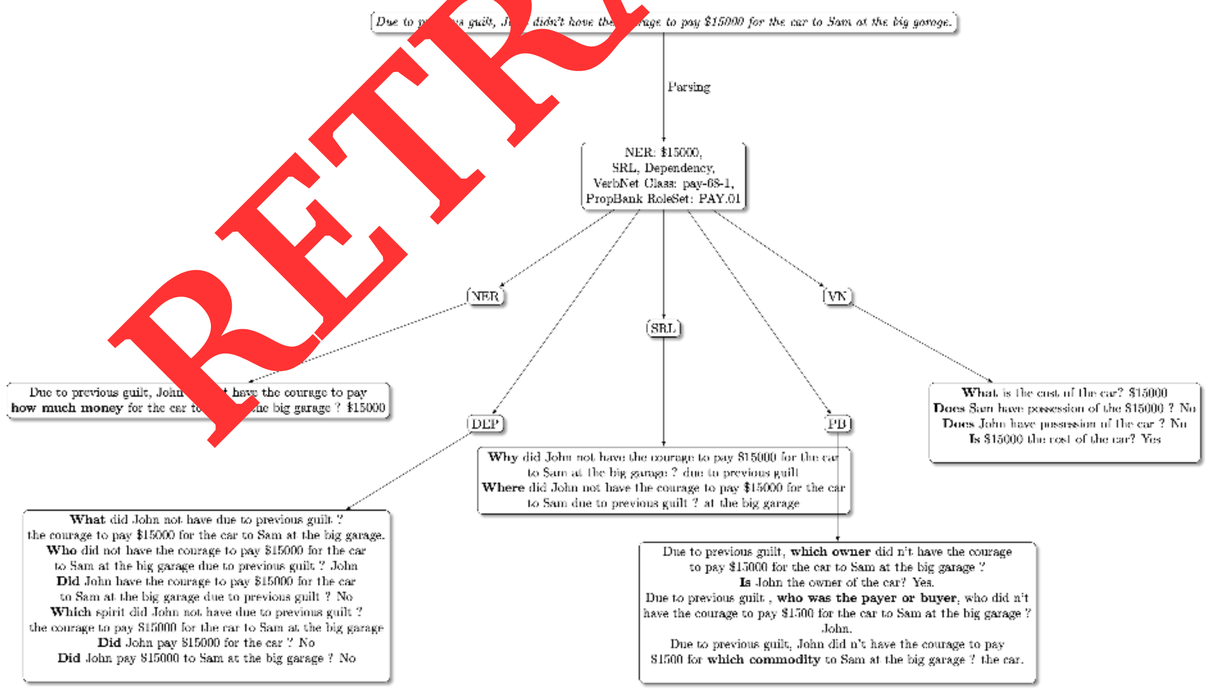

Figure 6: Questions generated by each set of heuristics for one sentence which are further sent for back-translation.

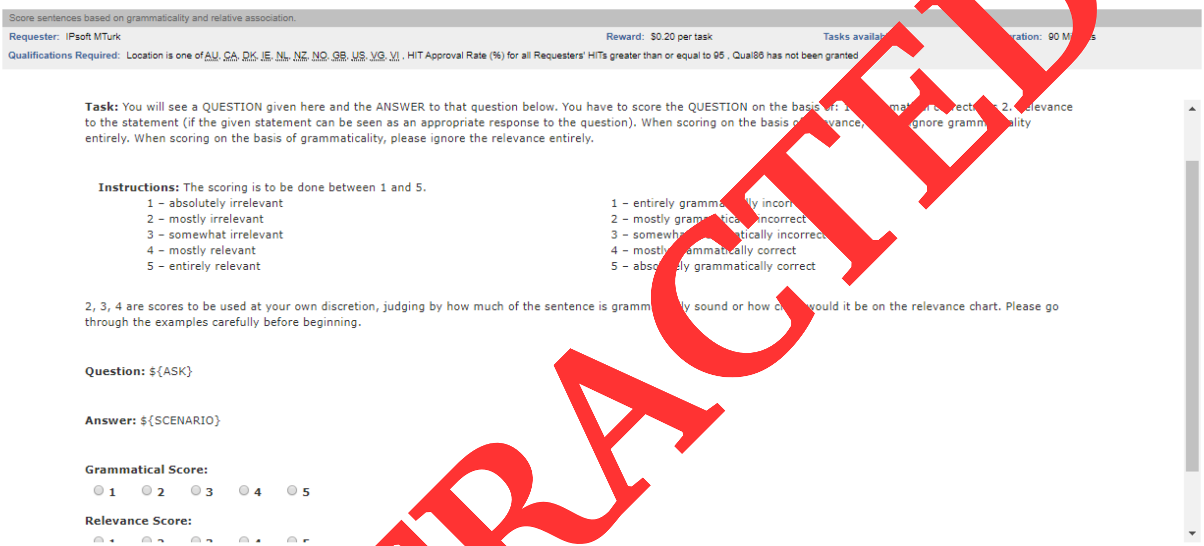

Figure 7: The MTURK template used for collecting responses for measuring question relevance and grammaticality.