# A Semantic Role-based Approach to Open-Domain Automatic Question Generation

**Michael Flor** and **Brian Riordan**
Educational Testing Service
660 Rosedale Road, Princeton, NJ 08541, USA
{mflor, briordan}@ets.org

## Abstract

We present a novel rule-based system for automatic generation of factual questions from sentences, using semantic role labeling (SRL) as the main form of text analysis. The system is capable of generating both *wh*-questions and yes/no questions from the same semantic analysis. We present an extensive evaluation of the system and compare it to a recent neural network architecture for question generation. The SRL-based system outperforms the neural system in both average quality and variety of generated questions.

## 1 Introduction

Automatic generation of questions (AQG) is an important and challenging research area in natural language processing. AQG systems can be useful for educational applications such as assessment of reading comprehension, intelligent tutoring, dialogue agents, and instructional games. Most of the research on AQG focuses on factoid questions – questions that are generated from reading passages and ask about information that is expressed in the text itself (as opposed to, e.g., readers' opinions of the text or external knowledge related to the text).

Traditional architectures for AQG involve syntactic and semantic analysis of text, with rule-based and template-based modules for converting linguistic analyses into questions. Many of these systems employ semantic role labeling (SRL) as an important analytic component (Mazidi and Tarau, 2016; Huang and He, 2016). Recently, neural network architectures have also been proposed for the AQG task (Du et al., 2017; Serban et al., 2016).

In this paper we present an automatic question generation system based on semantic role labeling. The system generates questions directly from semantic analysis, without templates. Our system includes two innovations. While previous SRL-based AQG systems generated only *wh*-questions, ours is the first reported system that also generates yes/no questions from SRL analysis. It is also the first system that generates questions for copular sentences from their SRL analysis (both yes/no and *wh*-questions).

To evaluate the performance of our system, we compare the quality of its output with that of a state-of-the-art neural network AQG system, over the same set of texts. To the best of our knowledge, ours is the first direct comparison of SRL-based and neural AQG systems.

The rest of this paper is structured as follows. Section 2 presents related work on AQG. Section 3 describes our SRL-based system and section 4 outlines the neural network AQG system. Section 5 describes the annotation study. Results are presented in section 6 and error analysis in section 7.

## 2 Related work

The bulk of research on automatic question generation from text takes one of two basic approaches: transforming sentences into questions using various intermediate representations, or generating questions from predefined templates, where the appropriate template for each question is selected based on analysis of the text. In both approaches, the analysis of text plays a major role. Text analysis is focused on primarily syntax-based methods or more semantics-based methods.

Syntax-based methods apply a parser to determine the syntactic structure of a sentence, then apply syntactic transformation rules and question word placement (e.g., "where"). The earliest such system was proposed by Wolfe (1976). Contemporary systems use constituent and dependency parsing (Heilman and Smith, 2010a; Varga and Ha, 2010; Kalady et al., 2010; Ali et al., 2010). Yao et al. (2012) proposed a system based on HPSG parsing with semantic analysis. A recent



*Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 254–263
New Orleans, Louisiana, June 5, 2018. 

example of the syntax-based approach is the system of Danon and Last (2017).

Semantics-based methods place greater emphasis on semantic analysis of texts, although they typically also use some syntactic analysis. Huang and He (2016) present an AQG system that uses the Lexical Functional Grammar representation, including syntactic and semantic layers. Araki et al. (2016) present a study of AQG from richly annotated sources. Many AQG systems rely on semantic role labeling as the main driver of linguistic analysis (Rodrigues et al., 2016; Mazidi and Tarau, 2016; Mazidi and Nielsen, 2015; Lindberg et al., 2013; Mannem et al., 2010), or as a supporting subsystem (Huang and He, 2016).

With recent advances in neural networks, some approaches forgo most linguistic analysis and train neural networks to generate questions from sequences of word tokens (Du et al., 2017; Serban et al., 2016). Using large quantities of paired texts and human-generated questions and the encoder-decoder neural network framework, these systems learn to map from sentences to questions in a manner similar to neural machine translation approaches. Further detail on neural network systems for question generation and the specific benchmark system we use is provided in Section 4.

### 2.1 Common issues in AQG research

Most research on AQG systems needs to address the following set of common issues: 1) content selection; 2) target identification; 3) simplification; 4) question formulation, and 5) evaluation.

*Content selection* refers to picking sections of the source text (typically single sentences) for which questions should be generated, i.e. what parts of the text are worth asking a question about (Vanderwende, 2008). Prior research embraced the working assumption that content selection should focus on the most important and salient information in a text. Hence, some AQG systems used automatic extractive summarization for sentence selection (Becker et al., 2012; Agarwal and Mannem, 2011). Recently, Du and Cardie (2017) described a neural architecture for the content selection task in AQG.

*Target selection* defines what exactly should be asked about the selected content. For example, given a sentence like *The executive arrived at 5pm in a black limousine*, we could ask who arrived, when, or in what kind of vehicle. Clearly, a variety of questions can be posed, and their selection may heavily depend on the educational task, e.g. assisting in reading comprehension (Gates, 2008), writing literature reviews (Liu et al., 2012), learning online (Lindberg et al., 2013).

*Simplification of text* has two aspects. Texts often use complex and long sentences, but questions are rarely very long. For a human reader, shorter questions are easier to process. From the perspective of AQG, simplification of the original text is sometimes necessary for applying transformation or matching to predefined templates (Lindberg et al., 2013; Yao et al., 2012; Heilman and Smith, 2010a).

*Question formulation* involves the actual process of generating a question and producing the final surface-form realization. Systems differ widely in this respect. For factoid questions, syntactic transformations or semantic analysis are often sufficient for question formulation. Template-based methods allow asking questions that can go beyond the explicit information in a text (Mazidi and Tarau, 2016; Lindberg et al., 2013).

*Evaluation* of AQG systems is a complex task in itself. Common criteria for sentence-based questions are grammaticality (syntactic correctness), relevance to the input sentence, the variety of question types produced, and semantic appropriateness (Godwin and Piwek, 2016; Chali and Golestanirad, 2016; Heilman and Smith, 2010b). Lindberg et al. (2013) add the notion of learning value (pedagogical usefulness) for question evaluation. However, the pedagogical value of questions is tightly related to the goals of the question use (Mazidi and Nielsen, 2014).

## 3 SRL-based system

Our SRL-based AQG system uses a mostly standard NLP pipeline structure with the following steps: 1) tokenization and sentence boundary detection; 2) POS tagging; 3) detection of verbal groups; 4) semantic role labeling; 5) postprocessing; 6) question generation.

For POS-tagging we use OpenNLP.[1] We use the SENNA system (Collobert et al., 2011) for semantic role labeling, similar to some previous research in AQG (Mazidi and Nielsen, 2015; Lindberg et al., 2013).

[1] https://opennlp.apache.org

Given a sentence, SENNA produces semantic role labels according to the Propbank 1.0 specifications (Palmer et al., 2005). Verbs in a sentence are considered as predicates. Semantic roles include the generalized core arguments of verbs – labeled A0, A1, etc. – and a set of adjunct modifiers. Table 1 provides an overview.

| Label | Role |
|---|---|
| A0 | proto-agent (often grammatical subject) |
| A1 | proto-patient (often grammatical object) |
| A2 | instrument, attribute, benefactive, amount, etc. |
| A3 | start point or state |
| A4 | end point or state |
| AM-LOC | location |
| AM-DIR | direction |
| AM-TMP | time |
| AM-CAU | cause |
| AM-PNC | purpose |
| AM-MNR | manner |
| AM-EXT | extent |
| AM-DIS | discourse |
| AM-ADV | adverbial |
| AM-MOD | modal verb |
| AM-NEG | negation |

Table 1: Semantic roles per PropBank 1.0 specification.

**Detection of verbal groups**. In an English language clause, a verbal group consists of the main lexical verb and its related modifiers – negation, auxiliary verbs, and modals (Palmer, 1987). A sentence with multiple clauses may have several verbal groups. The verbal group does not include the semantic roles or their fillers, although there is some overlap with the Propbank definitions, since Propbank includes Modal and Negation as semantic arguments. Our question generation system includes a rule-based module for detection and analysis of verbal groups in sentences. The module uses POS and lexical patterns to identify verbal groups and analyze tense, grammatical aspect, verb negation, modality, and grammatical voice (passive/active). All of this information is necessary for adequate formulation of questions.

**Postprocessing**. In the postprocessing step, we correct several issues in the SRL output. The SENNA system tends to assign the A1 role for subjects instead of A0. For example, for *John arrived today*, 'John' is assigned A1. This also often happens for copula sentences, e.g. SENNA produces: *[$_{A1}$John] is [$_{A1}$ a painter]*. Since we want to treat A1 assignments as direct objects, we automatically remap A1 in objectless clauses to a specially devised category, A01, which, for question generation, is treated the same as A0 arguments (i.e., as grammatical subjects).

Another step in postprocessing is linking the verbal group to the verb of the detected predicate. In the presence of auxiliary verbs, SENNA produces multiple analyses for the same chunk of text, and some of them are systematically incorrect. We are able to correct this by utilizing the separately detected verbal group. For example, for *Joe has sold his house*, SENNA produces both *[$_{A0}$ Joe] [$_{Predicate}$ has] [$_{A1}$ sold his house]* and *[$_{A0}$ Joe] has [$_{Predicate}$ sold] [$_{A1}$ his house]* A verbal group would indicate that 'has' is an auxiliary of 'sold', and our system would pick up the second analysis.

### 3.1 Generating constituent questions

Constituent questions (CQ, a.k.a. *wh*-questions) are the most common type of question in AQG research. Semantic role labeling is a natural choice for CQ generation, since SRL basically analyzes a sentence into *who did what to whom, how and when...*. Producing CQ from SRL involves three main steps: a) focusing, b) producing the question word(s), and c) formulating the question.

**Focusing.** To generate a question for a predicate, we need to choose the focal argument – the argument about which the question will be asked. We create questions from all of the major arguments, and also for the following adjunct arguments:[2] AM-TMP, AM-MNR, AM-CAU, AM-LOC, AM-PNC, AM-DIR. The text of the chosen focal argument becomes the expected answer to the question.

**Producing question words** involves some intricate decisions. There are at least three broad issues: 1) selecting the appropriate question word for the semantic argument, 2) deciding on *What* vs. *Who*, and 3) handling prepositions.

Selecting the appropriate *wh*-word is aided by the identity of the focused argument. Manner (AM-MNR) invites *How* and location (AM-LOC) invites *Where*. However, the situation is not quite so simple. Consider, for example, semantic role

[2]Selecting question focus by semantic roles may be useful for user customization. For example one may wish to focus questions only on manner arguments, cause and purpose, etc.

A4, which is often used for the 'end point' of complex locative constructions. A sentence like *They can fly from here [$_{A4}$ to any country]*, should generate a question with *Where*. However, a similar construction in *Antarctica doesn't belong [$_{A4}$ to any country]* should not produce a *Where* question.

A major issue is deciding on whether to use *Who* or *What* (for subject, direct object, and some other cases). Currently we make a rule-based decision, based on examining the POS of the argument, presence of pronouns, a check in a large gazetteer of first and last person names (about 130K entries), and a lookup into a list of person-denoting words derived from WordNet supersenses[3] (Fellbaum, 1998) (e.g., *king, senator*, etc.). If the argument is a whole phrase, a careful analysis is required. For example, *king of the land* is a *Who*, but *a hat for a lady* is a *What*.

The complexity of generating adequate question words is well illustrated with the case of temporal arguments. It is not the case that everything tagged as AM-TMP can have a question with *When* generated for it. Essentially, an SRL designation of AM-TMP is too general. It does not distinguish between time points, durations, and sets (repetitive temporal specifications). (For detailed temporal nomenclature, see, for example, Verhagen et al. (2010)). This is the minimal distinction that is necessary for *When*-questions, as opposed to *How long* and *How often*. As an illustration, consider the following sentences:

1. [$_{A0}$Peter] *called* [$_{AM-TMP}$ on Monday].
2. [$_{A0}$Peter] *called* [$_{AM-TMP}$ for six hours].
3. [$_{A0}$Peter] *called* [$_{AM-TMP}$ every day].

Their corresponding proper questions are: 1) *When did Peter call?* (A: on Monday); 2) *For how long did Peter call?* (A: for six hours); 3) *How often did Peter call?* (A: every day).

Inspired by research on rule-based handling of time-expressions (Chang and Manning, 2013; Strotgen and Gertz, 2010), we designed a rule-based algorithm for subclassification of temporal expressions. Prepositions in time expressions are major clues in this task. For example, 'every' and 'each' hint at *How often*, 'for' hints at *Duration*, while many other prepositions hint at a time point (or time range) description, which is asked about with *When*. Some prepositions of temporal expressions are retained to be used in the questions, for example *from/until Monday* → `from/until when?`, *for five minutes* → `for how long?`.

Prepositions are sometimes retained for the formation of question word-sequences also for non-temporal semantic arguments. For example *The bird sat on the branch* → `On what did the bird sit?`. The who/what distinction can appear in this context as well. For example: *They rely on him/it* → `On whom/what do they rely?`

For **question formation** we need to select and rearrange the remaining arguments of the predicate. While SRL is a type of semantic analysis, for question formulation we need at least approximate grammatical information, such as the subject and direct object of the clause. For example, for *[$_{A0}$Danny] dropped [$_{A1}$ the package]*, with a focus on 'the package', we need to introduce do-support: `What did Danny drop?`. In the current implementation, we presume A0 arguments are subjects and A1 arguments are direct objects. Question formation also checks whether the verbal group is in active or passive voice, to adjust the placing of auxiliary verbs. Presently we do not convert passive sentences into active-voice questions.

### 3.2 Generating Yes/No questions

We generate a simple yes/no question (YNQ) for every predicate that has a finite verb (thus excluding bare and to-infinitives, and gerunds). If a sentence contains multiple predicates, we generate multiple yes/no questions – one for each predicate.

First, the system selects from a clause all chunks that are role-fillers for the current predicate. Next, the sequential position of SRL arguments may need to be rearranged. For yes/no questions, the standard declarative word order (usually SOV) is preserved. Do-support is provided when needed, based on the analysis of the verbal group (constructions that do not require do-support include copular, modals, and cases when an auxiliary *be/have/do* is already present). Adjunct arguments may be moved relative to the main verb (e.g. *he quickly ate* → `did he eat quickly ?`).

**Positivize.** For the current application, yes/no questions are always posed in positive mode.

[3] Supersenses were also used for this pupose by prior systems, e.g., Huang and He (2016), Heilman and Smith (2010a).

The analyzed verbal group of the predicate will have information about explicit negation of the main verb, including contracted negation, such as 'didn't' and 'couldn't'. The question generation process then avoids transferring the negation into the question, but it also registers that the correct answer is flipped from 'yes/no' to 'no/yes'. For example, from *Johnny didn't know the song*, we derive `Did Johnny know the song?` + Answer='no'. For the copula *The tea isn't sweet enough*, we derive `Is the tea sweet enough?` + Answer='no'.

## 4 Neural network benchmark system

The neural network system we used for comparison during evaluation is the LSTM-based system described by Du et al. (2017)[4]. The system is trained on a large corpus of question-answer pairs from Wikipedia. Given an input sentence, the system generates a question based on the encoded input and what the model has learned from the training data about plausible question content and form.

The network employs the encoder-decoder framework. An encoder network encodes an input sentence with a bidirectional LSTM. The network uses the encoded sentence to initialize a decoder network for question generation. The decoder generates a question token-by-token. At each time step $t$, the decoder employs a global bilinear attention mechanism (Luong et al., 2015) over the encoder representation, allowing the network to focus the encoded representation on tokens that are more salient for that time step. The network generates the next token using the decoder's state and the attention-weighted encoding of the input at $t$.

We use the sentence-oriented model[5] from Du et al. (2017), where only the input sentence is encoded. We use their code without modification.

We trained the network on the preprocessed version of the SQuAD dataset (Rajpurkar et al., 2016) provided by Du et al. (2017). SQuAD consists of 536 articles with more than 100,000 question-answer pairs generated by crowd workers. The corpus was processed with Stanford CoreNLP, and question-answer pairs without any non-stop words in common were filtered out. The model is trained on 80% of the data split at the article level.

The source vocabulary is 45,000 tokens and the target vocabulary is 28,000 tokens. Pretrained GloVe embeddings (Pennington et al., 2014) are used to initialize the token representations and are fixed during network training. The LSTM encoder used 2 layers with 600 dimensions. The decoder uses beam search with a beam size of 3. Of note, in Du et al.'s implementation, generated questions are post-processed by replacing any <UNK> tokens with the token from the input with the highest attention score.

## 5 Evaluation study

We evaluate the SRL and neural network systems' capacity to produce generally good questions, focusing only on question-generation capabilities. In this respect, our evaluation study differs from some previous studies in which the AQG system was tasked with performing both content selection (picking the 'important' sentences for which questions would be generated) and the question generation process itself. We believe that content selection depends very much on the goals of the educational task for which questions are generated, and should be seen as a separate task. For a similar perspective, see Rus et al. (2010).

Since our focus is on question generation for educational applications, we selected five expository texts. Three of the texts consist of several initial paragraphs from Wikipedia articles[6] and two are complete short articles from an educational website[7]. From those texts, we retained only declarative sentences that have at least five words and do not include conditional (if. . . then) clauses. Our corpus consists of 171 sentences, with a maximum sentence length of 50 tokens and an average of 17.

Both the neural and the SRL-based systems were tasked with generating questions for each of the 171 sentences.

The SRL-based system generated at least one question for 165 sentences and failed to provide an output for 6 sentences. Overall, the SRL-based system generated 890 questions, with an average of 5.4 questions per sentence. There are two reasons for this abundance. First, the system attempts to generate a yes/no question for each predicate in each sentence. As a result, it generated 236 yes/no questions. Next, the system attempts to generate a constituent question for almost every argument

[4] https://github.com/xinyadu/nqg

[5] Du et al. (2017) also propose a paragraph-oriented model.

[6] Fox (animal), Helsinki (city), and Celsius (scientist).

[7] beyondpenguins.ehe.osu.edu/stories-for-students

| Sentence | Generated Question | Sys | GR | SEM | REL |
|---|---|---|---|---|---|
| The Senate was relocated to Vaasa, although some senators and officials remained in hiding in the capital. | Who remained in hiding in the capital? | SRL | 5 | 5 | 3 |
| The Sun's temperature is much too hot for anything alive. | What the Sun's temperature is for anything alive? | SRL | 2.5 | 2.5 | 2 |
| Deep below is a place called the magma chamber. | Did a place call the magma chamber? | SRL | 4.5 | 1.5 | 2.5 |
| The particles from the Sun also carry an electric charge. | Do the particles from the Sun carry an electric charge? | SRL | 5 | 5 | 3 |
| Dr. Philip Kyle is one of these scientists. | Who is one of the scientists that is one of the scientists? | NN | 2 | 2 | 1.5 |
| But the real wonder is inside the volcano's crater itself. | What is the real view of the planet? | NN | 4.5 | 2 | 1 |
| Other foxes such as fennec foxes, are not endangered. | What are some other animals that are not endangered? | NN | 5 | 5 | 3 |

Table 2: Examples of sentences, generated questions and evaluation ratings (average of two raters).

of every predicate. If a sentence contains multiple predicates, even more questions are generated. The system generated 654 constituent questions.

The neural system generated one question for each of 169 sentences (and failed for two sentences). All questions generated by the system resemble constituent questions because the SQuAD dataset does not contain yes/no questions. We investigated whether it was possible to generate more than one question per sentence by retrieving hypotheses from the beam search, but the hypotheses are not fully formed and are small variants of the best question for each sentence.

### 5.1 Annotation

In total 1,060 questions were automatically generated for evaluation. The questions were annotated by two annotators with expertise in linguistic annotation of English Learning Arts materials and student-produced writing. Each question was rated on three scales: grammar, semantics and relevance.

The *grammar* scale is a five-point scale: 5) grammatically well-formed; 4) mostly well-formed, with slight problems; 3) has grammatical problems; 2) seriously disfluent; 1) severely mangled. The five-point *semantic* scale was intended to check to what extent the question 'understood' the semantics of the original sentence: 5) semantically adequate; 4) mostly semantically adequate, with slight problems; 3) has semantic problems; 2) serious misunderstanding of the original sentence; 1) severely mangled and makes no sense. The *relevance* scale was designed to check to what extent the generated question is about information that was conveyed in the original sentence. This scale had just four levels: 3) is about the sentence; 2) goes beyond the information in the sentence; 1) veers away, is unrelated to the sentence; 0) too mangled to make a reasonable judgment.

The annotators completed a training session with 272 questions that were generated from a separate set of texts.

Upon completion of training, the annotators received the 1060 questions of the main data set (with corresponding sentences, and access to original texts). Each annotator completed annotations individually. We measured inter-annotator agreement with Quadratically-weighted Kappa (QWK). Agreement was high: grammar = 0.75, semantics = 0.77, relevance = 0.48[8].

In our analysis we used the average ratings on each question for each of the categories. In addition, for each question we also computed a *total* rating, which is the sum of grammar, semantics, and relevance ratings. Samples of sentences with corresponding generated questions and ratings are presented in Table 2.

[8]The low agreement on relevance stemmed from the tendency of one of the annotators to lower the relevance rating to 0 when a question was 'mangled'.

## 6 Results

To estimate the quality of the various questions, we compared the average ratings for three groups of questions: yes/no and constituent questions from the SRL-based system (SRL-YNQ and SRL-CQ), and questions from the neural system (NN). We conducted ANOVA analyses for each of the three rating scales and for the total score (with Bonferrroni adjustment for pairwise contrasts). Results are presented in Table 3 and in Figure 1.

SRL-YNQ questions (n=236), are rated significantly higher than SRL-CQ (n=654), which, in turn, are rated significantly higher than questions from the neural system (n=169). All comparisons are statistically significant ($p < .001$), except for SRL-CQ vs. NN on grammar. In other words, the neural system-generated questions achieved a similar level of grammaticality judgment as the SRL system's constituent questions.

| Scale | SRL-YNQ | SRL-CQ | NN |
|---|---|---|---|
| Grammar | 4.32 | 3.89 | 3.75 |
| Semantics | 4.34 | 3.79 | 2.61 |
| Relevance | 2.75 | 2.52 | 1.65 |
| Total | 11.41 | 10.20 | 8.01 |

Table 3: Average ratings for SRL system yes/no questions (SRL-YNQ), constituent questions (SRL-CQ), and neural network questions (NN). *Total* is the sum of grammar, semantics, and relevance.

We also looked at the 163 sentences that have both a NN question and at least one SRL-CQ question. We picked the best scoring SRL-CQ question for each sentence (using total score values). The mean rating of the best SRL-CQ question per sentence is 12.2, while the mean rating of NN questions is 8.1. The difference is statistically significant (t-test, $p < .0001$). Thus, if we had to pick just one CQ question for each sentence, SRL-based questions are on average much better than NN-generated questions.

We also investigated to what extent the automatically generated questions might be potentially usable in a learning context (e.g. for reading comprehension assessment). We consider a *potentially useful* question to be one that has reasonably good grammar (rating $\geq 4$), is semantically sensible in context (rating $\geq 4$) and is relevant to the information conveyed in the text (rating $\geq 2$). We operationalize these criteria with two measures. First, we look at what proportion of questions have a total rating $\geq 10$. Among the SRL-YNQ questions, 81% are potentially useful, compared to 64% among SRL–CQ questions, and 29% among questions generated by the neural network. Our second, more stringent, measure is to require that a question meet the criteria above on each of the three scales, i.e. grammar $\geq$4, semantics $\geq$4, and relevance $\geq$2. With this measure, the proportion of potentially useful questions is 71% for SRL-YNQ questions, 50% for SRL-CQ questions, and 15% for the neural network-generated questions.

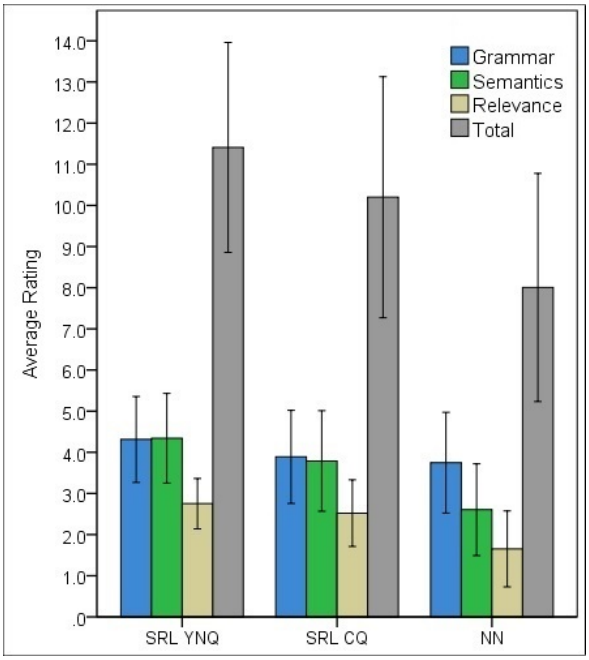


Figure 1: Average ratings and standard deviations for automatically generated questions, by system and question type. Note that score range is 1-5 for Grammar and Semantics, 0-3 for Relevance and 2-13 for Total.

## 7 Error analysis

We analyzed patterns of errors in SRL-based questions that received ratings below 4 on grammar and semantics and below 2 on relevance.

Among the constituent questions generated by the SRL-based system, we randomly sampled 30 questions. The most common reason for errors (33%) was incorrect handling of longer and more complicated sentences, including incorrect handling of arguments in subordinate clauses. For example, for the sentence *Red foxes have been introduced into Australia, which lacks similar carnivores...*, one of the generated questions was `What lacks?` This question misses the subject, *Australia*, which only appears in the matrix clause.

Incorrect handling of subordinate clauses is also one of the common reasons for errors among the SRL-based yes/no questions. For example, for the sentence *It's a little like the sound waves bats and dolphins use to find objects in the air and water.*, the system generated `Do bats and dolphins use to find objects in the air and water?`. The proper question should have been: `Do bats...use sound waves...to find...?`. The necessary direct object, *sound waves*, is outside the reduced relative clause and was missed in question generation.

### 7.1 Analysis of NN system errors

The patterns of ratings for errorful questions from the neural system differed from the SRL system. One pattern, of high grammaticality but low semantic coherence and relevance (22.7%), was attributable to strange substitutions of words in the original sentence. For example, for the sentence *Greater Helsinki has eight universities and six technology parks*, the generated question was: `How many universities does greater Strasbourg have?` Another common pattern was repetition of a word or phrase in the question. For example: `What type of birds do birds usually live?` Word repetition caused poor ratings on all scales. Another notable pattern was high grammaticality but low semantic coherence and relevance. This pattern is sometimes characterized by word substitutions but more generally a lack of analysis of the original sentence. For example, for the sentence *Despite the tumultuous first half of the 20th century, Helsinki continued to develop steadily*, the system generated: `When did the first half of the 20th century occur?`

We also analyzed a sample of sentences that were rated highly across all categories. Many of these sentences were simple declarative sentences. For the most part, the network reused words from the original sentence and created grammatical questions. In a few instances, the network gave hints of an ability to generalize lexical items. For example, for the sentence, *In fact, as the inside walls of the igloo start to melt, they come into contact with...*, the generated question was: `What do the walls of the igloo begin to do?`

## 8 Discussion

The SRL-based system generates a relatively high percentage of questions that are potentially usable as-is in an application, achieving good ratings for grammaticality, semantic coherence, and relevance. The SRL system was able to generate particularly high quality yes/no questions, as demonstrated by the strong scores from the human raters. Another strength demonstrated by the SRL-based system was the ability to systematically generate multiple constituent questions by focusing on each argument of a predicate in a clause.

The average quality of yes/no questions generated by the SRL system is significantly higher than the average quality of the generated constituent questions. The reason for this is mostly due to the fact that, while both types of questions are generated based on the same SRL analysis, yes/no questions require less complicated processing for generation.

While the questions produced by the SRL system show a promising level of quality, one area where the system falters is in handling long and complicated sentences, particularly those that involve subordinated clauses.

Although we did not focus on augmenting the neural network system for this study, our results demonstrate that the basic neural architecture of LSTM and attention already shows a surprising ability to produce readable questions, as indicated by reasonably high average grammaticality ratings. At the same time, the neural system had difficulty producing semantically adequate and relevant questions. These results point to the need for improved semantic analysis in neural AQG systems.

## 9 Conclusions

In this work, we described a novel rule-based system for automatic generation of factual questions from sentences that leverages semantic role labeling for text analysis and is capable of generating both *wh*-questions and yes/no questions from the same semantic analysis. Both of these capabilities are likely to prove useful in practical applications, for example to limit generated questions to only certain types of constituents or to generate questions of only certain forms. Another practical advantage of SRL-based AQG is that this approach produces questions with corresponding answers. This can be very useful for downstream applica-

tions such as quiz generators or automated scoring of responses.

We presented a detailed evaluation of the system and compared it to a state-of-the-art neural network architecture for question generation. The SRL-based system produced questions with greater variety and higher average quality than the neural system. In future work, we will explore methods for combining the strengths of rule-based and neural methods for text analysis and question generation.

## Acknowledgments

Many thanks to our raters, Jennifer Wain and Jeremy Lee. The paper benefited much from the comments of three anonymous BEA reviewers and comments by Aoife Cahill, Ikkyu Choi, Beata Beigman Klebanov, and David Pautler.

## References

Manish Agarwal and Prashanth Mannem. 2011. Automatic gap-fill question generation from text books. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–64, Portland, OR, USA. Association for Computational Linguistics.

Husam Ali, Yllias Chali, and Sadid A. Hasan. 2010. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67.

Jun Araki, Dheeraj Rajagopal, Sreecharan Sankaranarayanan, Susan Holm, Yukari Yamakawa, and Teruko Mitamura. 2016. Generating questions and multiple-choice answers using semantic analysis of texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1125?–1136. Association for Computational Linguistics.

Lee Becker, Sumit Basu, and Lucy Vanderwende. 2012. Mind the gap: Learning to choose gaps for question generation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 742–751, Montréal, Canada. Association for Computational Linguistics.

Yllias Chali and Sina Golestanirad. 2016. Ranking automatically generated questions using common human queries. In *Proceedings of The 9th International Natural Language Generation conference*, pages 217–?231. Association for Computational Linguistics.

Angel Chang and Christopher D. Manning. 2013. SUTime: Evaluation in TempEval-3. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 78–82, Atlanta, Georgia, USA. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 2011:2493–2537.

Guy Danon and Mark Last. 2017. A syntactic approach to domain-specific automatic question generation. In *linguisarXiv:1712.09827v1*.

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Copenhagen, Denmark. 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.

Chrisiane Fellbaum, editor. 1998. *WorNet: an electronic lexical database*. The MIT Press, Cambridge, MA, USA.

Donna M. Gates. 2008. Generating look-back strategy questions from expository texts. In *The Workshop on the Question Generation Shared Task and Evaluation Challenge*, Arlington, VA, USA.

Keith Godwin and Paul Piwek. 2016. Collecting reliable human judgements on machine-generated language: The case of the qg-stec data. In *Proceedings of The 9th International Natural Language Generation conference*, pages 212?–216. Association for Computational Linguistics.

Michael Heilman and Noah A Smith. 2010a. Good question! Statistical ranking for question generation. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–?617. Association for Computational Linguistics.

Michael Heilman and Noah A Smith. 2010b. Rating computer-generated questions with mechanical turk. In *In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon?s Mechanical Turk*, pages 35?–40. Association for Computational Linguistics.

Yan Huang and Lianzhen He. 2016. Automatic generation of short answer questions for reading comprehension assessment. *Natural Language Engineering*, 22(3):457–489.

Saidalavi Kalady, Ajeesh Elikkottil, and Rajarshi Das. 2010. Natural language question generation using syntax and keywords. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 1–10.

David Lindberg, Fred Popowich, John Nesbit, and Philip 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. Association for Computational Linguistics.

Ming Liu, Rafael A. Calvo, and Vasile Rus. 2012. G-asks: an intelligent automatic question generation system for academic writing support. *Dialogue and Discourse*, 3(2):101–124.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. 2010. Question generation from paragraphs at UPenn: QGSTEC system description. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 84–91.

Karen Mazidi and Rodney D Nielsen. 2014. Pedagogical Evaluation of Automatically Generated Questions. In Stefan Trausan-Maru, Kristy Elizabeth Boyer, Martha Crosby, and Kitty Panourgia, editors, *Proceededngs of the 12th international conference on Intelligent Tutoring Systems*, pages 294–299. Springer, Honolulu, HI, USA.

Karen Mazidi and Rodney D Nielsen. 2015. Leveraging multiple views of text for automatic question generation. In TBD, editor, *Artificial Intelligence in Education, LNCS*, pages 0–0. Springer, TBD.

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. Association for Computational Linguistics.

Frank R. Palmer. 1987. *The English Verb*, 2nd edition. Longman, London, UK.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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.

Hugo Rodrigues, Luisa Coheur, and Eric Nyberg. 2016. Qgasp: a framework for question generation based on different levels of linguistic information. In *Proceedings of The 9th International Natural Language Generation conference*, pages 242?–243. Association for Computational Linguistics.

Vasile Rus1, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. Overview of The First Question Generation Shared Task Evaluation Challenge. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 45–57.

Iulian Vlad Serban, Alberto Garcia-Duran, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questionswith recurrent neural networks: The 30m factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 588?–598. Association for Computational Linguistics.

Jannik Strotgen and Michael Gertz. 2010. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

Lucy Vanderwende. 2008. The importance of being important: Question generation. In *Workshop on the Question Generation, Shared Task and Evaluation Challenge*, pages 1342?–1352.

Andrea Varga and Le An Ha. 2010. WLV: A question generation system for the QGSTEC 2010, task b. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 80–83.

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

John H. Wolfe. 1976. Automatic question generation from text - an aid to independent study. *ACM SIGCUE Outlook*, 10:104–112.

Xuchen Yao, Gosse Bouma, and Yi Zhang. 2012. Semantics-based question generation and implementation. *Dialogue and Discourse*, 3(2):11–42.