Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language

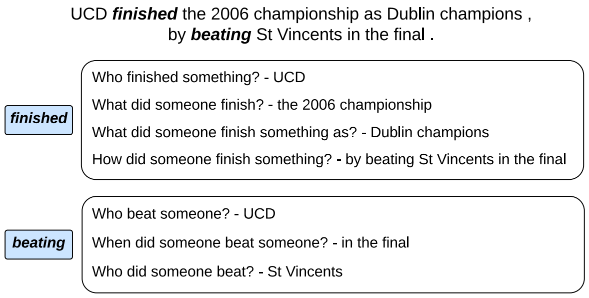
Luheng He **Mike Lewis** **Luke Zettlemoyer**
Computer Science & Engineering
University of Washington
Seattle, WA
{luheng,mlewis,lsz}@cs.washington.edu

Abstract

This paper introduces the task of question-answer driven semantic role labeling (QA-SRL), where question-answer pairs are used to represent predicate-argument structure. For example, the verb "introduce" in the previous sentence would be labeled with the questions "What is introduced?", and "What introduces something?", each paired with the phrase from the sentence that gives the correct answer. Posing the problem this way allows the questions themselves to define the set of possible roles, without the need for predefined frame or thematic role ontologies. It also allows for scalable data collection by annotators with very little training and no linguistic expertise. We gather data in two domains, newswire text and Wikipedia articles, and introduce simple classifier-based models for predicting which questions to ask and what their answers should be. Our results show that non-expert annotators can produce high quality QA-SRL data, and also establish baseline performance levels for future work on this task.

1 Introduction

Semantic role labeling (SRL) is the widely studied challenge of recovering predicate-argument structure for natural language words, typically verbs. The goal is to determine "who does what to whom," "when," and "where," etc. (Palmer et al., 2010; Johansson and Nugues, 2008). However, this intuition is difficult to formalize and fundamental aspects of the task vary across efforts, for example FrameNet (Baker et al., 1998) models a large set of interpretable thematic roles (AGENT, PATIENT, etc.) while PropBank (Palmer et al., 2005) uses a small set of verb-specific roles (ARG0, ARG1, etc.). Existing task definitions can be complex and require significant linguistic expertise to understand,[1] causing challenges for data annotation and use in many target applications.



UCD ***finished*** the 2006 championship as Dublin champions , by ***beating*** St Vincents in the final .

finished

Who finished something? - UCD
What did someone finish? - the 2006 championship
What did someone finish something as? - Dublin champions
How did someone finish something? - by beating St Vincents in the final

beating

Who beat someone? - UCD
When did someone beat someone? - in the final
Who did someone beat? - St Vincents

Figure 1: QA-SRL annotations for a Wikipedia sentence.

In this paper, we introduce a new question-answer driven SRL task formulation (QA-SRL), which uses question-answer pairs to label verbal predicate-argument structure. For example, for the sentence in Figure 1, we can ask a short question containing a verb, e.g. *"Who finished something?"*, and whose answer is a phrase from the original sentence, in this case *"UCD."* The answer tells us that *"UCD"* is an argument of "finished," while the question provides an indirect label on the role that *"UCD"* plays. Enumerating all such pairs, as we will see later, provides a relatively complete representation of the original verb's arguments and modifiers.

The QA-SRL task formulation has a number of advantages. It can be easily explained to non-expert annotators with a short tutorial and a few examples. Moreover, the formulation does not depend on any pre-defined inventory of semantic roles or frames, or build on any existing gram-

[1] The PropBank annotation guide is 89 pages (Bonial et al., 2010), and the FrameNet guide is 119 pages (Ruppenhofer et al., 2006). Our QA-driven annotation instructions are 5 pages.

mar formalisms. Nonetheless, as we will show, it still represents the argument and modifier attachment decisions that have motivated previous SRL definitions, and which are of crucial importance for semantic understanding in a range of NLP tasks, such as machine translation (Liu and Gildea, 2010) and coreference resolution (Ponzetto and Strube, 2006). The annotations also, perhaps surprisingly, capture other implicit arguments that cannot be read directly off of the syntax, as was required for previous SRL approaches. For example, in *"It was his mother's birthday, so he was going to play her favorite tune"*, annotators created the QA pair *"When would someone play something? His mother's birthday"* which describes an implicit temporal relation. Finally, QA-SRL data can be easily examined, proofread, and improved by anyone who speaks the language and understands the sentence; we use natural language to label the structure of natural language.

We present a scalable approach for QA-SRL annotation and baseline models for predicting QA pairs. Given a sentence and target word (the verb), we ask annotators to provide as many question-answer pairs as possible, where the question comes from a templated space of wh-questions[2] and the answer is a phrase from the original sentence. This approach guides annotators to quickly construct high quality questions within a very large space of possibilities. Given a corpus of QA-SRL annotated sentences, we also train baseline classifiers for both predicting a set of questions to ask, and what their answers should be. The question generation aspect of QA-SRL is unique to our formulation, and corresponds roughly to identifying what semantic role labels are present in previous formulations of the task. For example, the question *"Who finished something"* in Figure 1 corresponds to the AGENT role in FrameNet. Table 1 also shows examples of similar correspondences for PropBank roles. Instead of pre-defining the labels, as done in previous work, the questions themselves define the set of possibilities.

Experiments demonstrate high quality data annotation with very little annotator training and establish baseline performance levels for the task. We hired non-expert, part-time annotators on Upwork (previously oDesk) to label over 3,000 sentences (nearly 8,000 verbs) across two domains (newswire and Wikipedia) at a cost of approximately $0.50 per verb. We show that the data is high quality, rivaling PropBank in many aspects including coverage, and easily gathered in non-newswire domains.[3] The baseline performance levels for question generation and answering reinforce the quality of the data and highlight the potential for future work on this task.

In summary, our contributions are:

- We introduce the task of question-answer driven semantic role labeling (QA-SRL), by using question-answer pairs to specify verbal arguments and the roles they play, without predefining an inventory of frames or semantic roles.
- We present a novel, lightweight template-based scheme (Section 3) that enables the high quality QA-SRL data annotation with very little training and no linguistic expertise.
- We define two new QA-SRL sub-tasks, question generation and answer identification, and present baseline learning approaches for both (Sections 4 and 5). The results demonstrate that our data is high-quality and supports the study of better learning algorithms.

2 Related Work

The success of syntactic annotation projects such as the Penn Treebank (Marcus et al., 1993) has led to numerous efforts to create semantic annotations for large corpora. The major distinguishing features of our approach are that it is not tied to any linguistic theory and that it can be annotated by non-experts with minimal training.

Existing SRL task formulations are closely related to our work. FrameNet (Baker et al., 1998) contains a detailed lexicon of verb senses and thematic roles. However, this complexity increases the difficulty of annotation. While the FrameNet project is decades old, the largest fully annotated corpus contains about 3,000 sentences (Chen et al., 2010). We were able to annotate over 3,000 sentences within weeks. PropBank (Kingsbury and Palmer, 2002), NomBank (Meyers et al., 2004) and OntoNotes (Hovy et al., 2006) circumvent the need for a large lexicon of roles, by defin-

[2] Questions starting with a wh-word, such as *who*, *what*, *when*, *how*, etc.

[3] Our hope is that this approach will generalize not only across different domains in English, as we show in this paper, but also to other languages. We will leave those explorations to future work.

Sentence	CoNLL-2009		QA-SRL	
(1) Stock-fund managers , meantime , went into October with less cash on hand than they **held** earlier this year .	A0	they	Who had held something?	Stock-fund managers / they
	AM-TMP	year	When had someone held something?	earlier this year
			What had someone held?	less cash on hand
			Where had someone held something?	on hand
(2) Mr. Spielvogel **added** pointedly : " The pressure on commissions did n't begin with Al Achenbaum . "	A0	Spielvogel	Who added something?	Mr. Spielvogel
	A1	did	What was added?	" The pressure on commissions did n't begin with Al Achenbaum . "
	AM-MNR	pointedly	How was something added?	pointedly
(3) He claimed losses totaling $ 42,455 – and the IRS **denied** them all .	A0	IRS	Who denied something?	IRS
	A1	them	What was denied?	losses / them
(4) The consumer - products and newsprint company said net **rose** to $ 108.8 million , or $ 1.35 a share , from $ 90.5 million , or $ 1.12 a share , a year ago .	A1	net	What rose?	net
	A3	$/ago	What did something rise from?	$ 90.5 million , or $ 1.12 a share
	A4	to	What did something rise to?	$ 108.8 million , or $ 1.35 a share
			When did something rise?	a year ago
(5) Mr. Agnew was vice president of the U.S. from 1969 until he **resigned** in 1973 .	A0	he	Who resigned from something?	Mr. Agnew
	AM-TMP	in	When did someone resign from something?	1973
			What did someone resign from?	vice president of the U.S.
(6) Mr. Gorbachev badly **needs** a diversion from the serious economic problems and ethnic unrest he faces at home .	A0	Gorbachev	Who needs something?	Gorbachev/he
	A1	diversion	What does someone need?	a diversion from the serious economic problems and ethnic unrest he faces at home
	AM-ADV	badly	How does someone need something?	badly
			What does someone need something from?	the serious economic problems and ethnic unrest he faces at home .
(7) Even a federal measure in June allowing houses to **add** research fees to their commissions did n't stop it .	A0	houses	What added something? houses	
	A1	fees	What was added?	research fees
	A2	to		
			When was something added?	June
(8) This year , Mr. Wathen says the firm will be able to **service** debt and still turn a modest profit .	A0	firm	Who will service something?	the firm
	A1	debt	What will be serviced?	debt
			When will something be serviced?	this year
(9) Clad in his trademark black velvet suit , the soft - spoken clarinetist announced that . . . and that it was his mother 's birthday , so he was going to **play** her favorite tune from the record .	A0	he	Who would play something?	the soft - spoken clarinetist / he
	A1	tune	What would be played?	her favorite tune from the record
			When would someone play something?	his mother 's birthday

Table 1: Comparison between CoNLL-2009 relations and QA-SRL annotations. While closely related to PropBank predicate-argument relations, QA pairs also contain information about within-sentence co-reference (Ex 3, 5, 6, 9), implicit or inferred relations (Ex 4, 7, 8, 9) and roles that are not defined in PropBank (Ex 1, 5). Annotation mistakes are rare, but for example include missing pronouns (Ex 5) and prepositional attachment errors (Ex 6).

ing the core semantic roles in a predicate-specific manner. This means that frames need to be created for every verb, and it requires experts to distinguish between different senses and different roles.

Our work is also related to recent, more general semantic annotation efforts. Abstract Meaning Representation (Banarescu et al., 2013) can be viewed as an extension of PropBank with additional semantic information. Sentences take 8-13 minutes to annotate—which is slower than ours, but the annotations are more detailed. Universal Cognitive Conceptual Annotation (UCCA) (Abend and Rappoport, 2013) is an attempt to create a linguistically universal annotation scheme by using general labels such as *argument* or *scene*. The UCCA foundational layer does not distinguish semantic roles, so *Frogs eat herons* and *Herons eat frogs* will receive identical annotation — thereby discarding information which is potentially useful for translation or question answering. They report similar agreement with PropBank to our approach (roughly 90%), but annotator training time was an order-of-magnitude higher (30-40 hours). The Groningen Meaning Bank (Basile et al., 2012) project annotates text by manually correcting the output of existing semantic parsers. They show that some annotation can be crowdsourced using "games with a purpose" — however, this does not include its predicate-argument structure, which requires expert knowledge of their syntactic and semantic formalisms. Finally, Reisinger et al. (2015) study crowdsourcing semantic role labels based on Dowty's proto-roles, given gold predicate and argument mentions. This work directly complements our focus on labeling predicate-argument structure.

The idea of expressing the meaning of natural language in terms of natural language is related to natural logic (MacCartney and Manning, 2007), in which they use natural language for logical inference. Similarly, we model predicate-argument structure of a sentence with a set of question-

Field	Description	Example of Values	No. Values
WH*	Question words (wh-words)	who, what, when, where, why, how, how much	7
AUX	Auxiliary verbs	is, have, could, is n't	36
SBJ	Place-holding words for the subject position	someone, something	2
TRG*	Some form of the target word	built, building, been built	≈ 12
OBJ1	Place-holding words for the object position	someone, something	2
PP	Frequent prepositions (by, to, for, with, about) and prepositions (unigrams or bigrams) that occur in the sentence	to, for, from, by	≈ 10
OBJ2	Similar to OBJ1, but with more options	someone, something, do, do something, doing something	9

Table 2: Fields in our question annotation template, with descriptions, example values, and the total number of possible values for each. **WH*** and **TRG*** are required; all other fields can be left empty.

WH*	AUX	SBJ	TRG*	OBJ1	PP	OBJ2	
Who			built	something			?
What	had	someone	said				?
What	was	someone	expected		to	do	?
Where	might	something	rise		from		?

Table 3: Four example questions written with our question annotation template.

answer pairs. While existing work on natural logic has relied on small entailment datasets for training, our method allows practical large-scale annotation of training data.

Parser evaluation using textual entailment (Yuret et al., 2010) is a method for evaluating syntactic parsers based on entailment examples. In a similar spirit to our work, they abstract away from linguistic formalisms by using natural language inference. We focus on semantic rather than syntactic annotation, and introduce a scalable method for gathering data that allows both training and evaluation. Stern and Dagan (2014) applied textual entailment to recognize implicit predicate-argument structure that are not explicitly expressed in syntactic structure.

3 QA-based Semantic Dataset

This section describes our annotation process in more detail, and discusses agreement between our annotations and PropBank. Table 1 shows examples provided by non-expert annotators.[4]

3.1 Annotation Task Design

We annotate verbs with pairs of questions and answers that provide information about predicate-argument structure. Given a sentence s and a verbal predicate v in the sentence, annotators must produce a set of wh-questions that contain v and whose answers are phrases in s.

To speed annotation and simplify downstream processing, we define a small grammar over possible questions. The questions are constrained with a template with seven fields, $q \in \textbf{WH} \times \textbf{AUX} \times \textbf{SBJ} \times \textbf{TRG} \times \textbf{OBJ1} \times \textbf{PP} \times \textbf{OBJ2}$, each associated with a list of possible options. Descriptions for each field are shown in Table 2. The grammar is sufficiently general to capture a wide-range of questions about predicate-argument structure—some examples are given in Table 3.

The precise form of the question template is a function of the verb v and sentence s, for two of the fields. For the **TRG** field, we generate a list of inflections forms of v using the Wiktionary dictionary. For the **PP** field, the candidates are all the prepositions that occurred in the sentence s, and some frequently-used prepositions - *by*, *to*, *for*, *with*, and *about*. We also include preposition bigrams (e.g., *out for*) from s.

Answers are constrained to be a subset of the words in the sentence but do not necessarily have to be contiguous spans. We also allow questions to have multiple answers, which is useful for annotating graph structured dependencies such as those in examples 3 and 6 in Table 1.

3.2 Data Preparation

We annotated over 3000 sentences (nearly 8,000 verbs) in total across two domains: newswire (PropBank) and Wikipedia. Table 4 shows the full data statistics. In the newswire domain, we sampled sentences from the English training data of CoNLL-2009 shared task (Hajič et al.,

[4] Our dataset is freely available at: `https://dada.cs.washington.edu/qasrl`.

Dataset	Sentences	Verbs	QAs
newswire-train	744	2020	4904
newswire-dev	249	664	1606
newswire-test	248	652	1599
Wikipedia-train	1174	2647	6414
Wikipedia-dev	392	895	2183
Wikipedia-test	393	898	2201

Table 4: Annotated data statistics.

2009), excluding questions and sentences with fewer than 10 words. For the Wikipedia domain, we randomly sampled sentences from the English Wikipedia, excluding questions and sentences with fewer than 10 or more than 60 words.

In each sentence, we need to first identify the candidates for verbal predicates. In principle, a separate stage of annotation could identify verbs—but for simplicity, we instead used POS-tags. We used gold POS-tags for newswire, and predicted POS-tags (using Stanford tagger (Toutanova et al., 2003)) in Wikipedia. Annotators can choose to skip a candidate verb if they are unable to write questions for it. Annotators skipped 136 verbs (3%) in Wikipedia data and 50 verbs (1.5%) in PropBank data.

3.3 Annotation Process

For annotation, we hired 10 part-time, non-expert annotators from Upwork (previously oDesk) and paid $10 per hour for their work. The average cost was $0.58 per verb ($1.57 per sentence) for newswire text and $0.45 per verb ($1.01 per sentence) on the Wikipedia domain. The annotators are given a short tutorial and a small set of sample annotations (about 10 sentences). Annotators were hired if they showed good understanding of English and our task. The entire screening process usually took less than 2 hours.

Writing QA pairs for each sentence takes 6 minutes on average for Wikipedia and 9 minutes on newswire, depending on the length and complexity of the sentence and the domain of the text.

3.4 Agreement with Gold PropBank Data (CoNLL-2009)

PropBank is the most widely used annotation of predicate-argument structure. While our annotation captures different information from PropBank, it is closely related. To investigate the similarity between the annotation schemes, we measured the overlap between the newswire domain (1241 sentences) of our QA-SRL dataset and the PropBank dataset.

	All Roles	Core	Adjuncts
Precision	81.4	85.9	59.9
Recall	86.3	89.8	63.6

Table 5: Agreement with gold PropBank (CoNLL-2009) for all roles, core roles, and adjuncts. Precision is the percentage of QA pairs covering exactly one PropBank relation. Recall is the percentage of PropBank relations covered by exactly one QA pair.

For each PropBank predicate that we have annotated with our scheme, we compute the agreement between the PropBank arguments and the QA-SRL answers. We ignore modality, reference, discourse and negation roles, as they are outside the scope of our current annotation. An annotated answer is judged to match the PropBank argument if either (1) the gold argument head is within the annotated answer span, or (2) the gold argument head is a preposition and at least one of its children is within the answer span.

We measure the macro-averaged precision and recall of our annotation against PropBank, with the proportion of our QA-pairs that are match a PropBank relation, and the proportion of PropBank relations covered by our annotation. The results are shown in Table 5, and demonstrate high overall agreement with PropBank. Agreement for core arguments [5] is especially strong, showing much of the expert linguist annotation in PropBank can be recovered with our simple scheme. Agreement for adjuncts is lower, because the annotated QAs often contain inferred roles, especially for *why*, *when* and *where* questions (See examples 4, 7 and 8 in Table 1). These inferred roles are typically correct, but outside of the scope of PropBank annotations; they point to exciting opportunities for future work with QA-SRL data. On the other hand, the adverbial arguments in PropBank are sometimes neglected by annotators, thus becoming a major source of recall loss.

Table 6 shows the overlap between our annotated question words and PropBank argument labels. There are many unsurprising correlations—*who* questions are strongly associated with Prop-

[5] In PropBank, A0-A5 are the core arguments. In QA-SRL, the core arguments include QA pairs with a question that starts with *Who* or *What*.

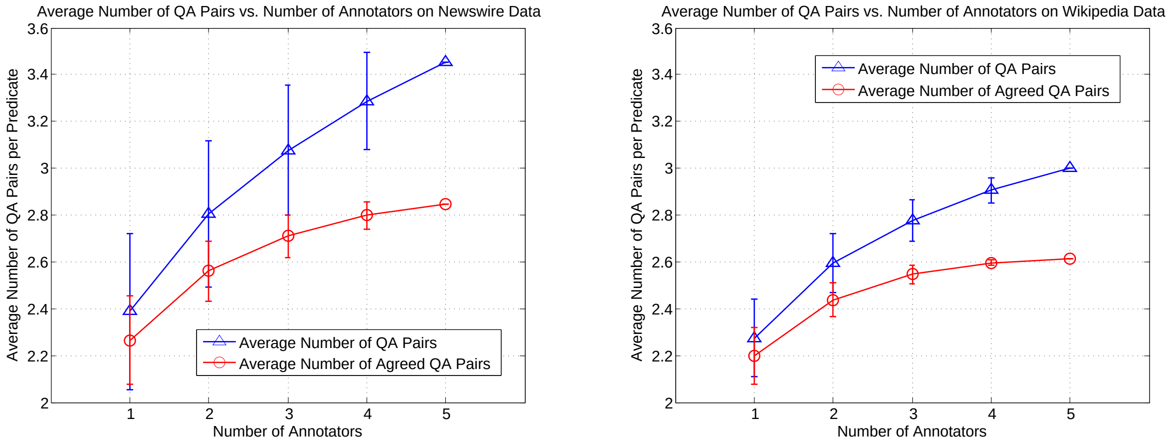


Figure 2: Inter-annotator agreement measured on 100 newswire sentences and 108 Wikipedia sentences, comparing the total number of annotators to the number of unique QA pairs produced and the number of agreed pairs. A pair is considered agreed if two or more annotators produced it.

	WHO	WHAT	WHEN	WHERE	WHY	HOW	HOW MUCH
A0	1575	414	3	5	17	28	2
A1	285	2481	4	25	20	23	95
A2	85	364	2	49	17	51	74
A3	11	62	7	8	4	16	31
A4	2	30	5	11	2	4	30
A5	0	0	0	1	0	2	0
ADV	5	44	9	2	25	27	6
CAU	0	3	1	0	23	1	0
DIR	0	6	1	13	0	4	0
EXT	0	4	0	0	0	5	5
LOC	1	35	10	89	0	13	11
MNR	5	47	2	8	4	108	14
PNC	2	21	0	1	39	7	2
PRD	1	1	0	0	0	1	0
TMP	2	51	341	2	11	20	10

Table 6: Co-occurrence of wh-words in QA-SRL annoations and role labels in PropBank.

Bank agents (A0), and *where* and *when* questions correspond to PropBank temporal and locative roles, respectively. Some types of questions are divided much more evenly among PropBank roles, such as *How much*. These cases show how our questions can produce a more easily interpretable annotation than PropBank labels, which are predicate-specific and can be difficult to understand without reference to the frame files.

Together, these results suggest that non-experts can annotate much of the information contained in PropBank, and produce a more easily interpretable annotation.

3.5 Inter-Annotator Agreement

To judge the reliability of the data, we measured agreement on a portion of the data (100 sentences in the newswire domain and 108 sentences in the Wikipedia domain) annotated by five annotators.

Measuring agreement is complicated by the fact that the same question can be asked in multiple ways—for example *"Who resigned?"* and *"Who resigned from something?"*—and annotators may choose different, although usually highly overlapping, answer spans. We consider two QA pairs to be equivalent if (1) they have the same wh-word and (2) they have overlapping answer spans. In this analysis, *Who* and *What* are considered to be the same wh-word.

Figure 2 shows how the number of different QA pairs (both overall and agreed) increases with number of annotations. A QA pair is considered to be agreed upon if it is proposed by at least two of the five annotators. After five annotators, the number of agreed QA pairs starts to asymptote. A single annotator finds roughly 80% of the agreed QA pairs that are found by five annotators, suggesting that high recall can be achieved with a single stage of annotation. To further improve precision, future work should explore a second stage of annotation where annotators check each other's work, for example by answering each other's questions.

4 Question Generation

Given a sentence s and a target verb v, we want to automatically generate a set of questions containing v that are answerable with phrases from s. This task is important because generating answerable questions requires understanding the predicate-argument structure of the sentence. In essence, questions play the part of semantic roles in our approach.[6]

We present a baseline that breaks down question generation into two steps: (1) we first use a classifier to predict a set of roles for verb v that are likely present in the sentence, from a small, heuristically defined set of possibilities and then (2) generate one question for each predicted role, using templates extracted from the training set.

Mapping Question Fields to Semantic Roles To generate questions, we first have to decide the primary role we want to target; each question's answer is associated with a specific semantic role. For example, given the sentence *UCD finished the 2006 championship* and target verb *finished*, we could ask either: (Q1) *Who finished something?* or (Q2) *What did someone finish?*. Q1 targets the role associated with the person doing the finishing, while Q2 focuses on the thing being finished. To generate high quality questions, it is also often necessary to refer to roles other than the primary role, with pronouns. For example, Q2 uses "someone" to refer to the finisher.

Although it is difficult to know a priori the ideal set of possible roles, our baseline uses a simple discrete set, and introduces heuristics for identifying the roles a question refers to. The roles $\mathcal{R}$ include:

$$\begin{aligned} \mathcal{R} =& \{\mathrm{R0}, \mathrm{R1}, \mathrm{R2}, \mathrm{R2}[p], w, w[p]\} \\ w \in& \{\textbf{Where}, \textbf{When}, \textbf{Why}, \textbf{How}, \textbf{HowMuch}\} \\ p \in& \text{Prepositions} \end{aligned}$$

We then normalize the annotated questions by mapping its fields **WH**, **SBJ**, **OBJ1** and **OBJ2** to the roles $r \in \mathcal{R}$, using a small set of rules listed in Table 7. In our example, the **WH** field of the Q1 (*Who*) and the **SBJ** of Q2 (*someone*) are both mapped to role R0. The **WH** of Q2 (*What*) and the **OBJ1** of Q1 (*something*) are mapped to role R1. rtSome roles can be subclassed with prepositions. For example, the **WH** field of the question *What did something rise from?* is mapped to R2[from].

In most cases, R0 is related to the A0/AGENT roles in PropBank/FrameNet, and R1/R2 are related to A1/PATIENT roles. Since our questions are defined in a templated space, we are able to do this mapping heuristically with reasonable accuracy. In the future, we might try to induce the set of possible roles given each target verb, following the semantic role induction work of Titov and Klementiev (2012) and Lang and Lapata (2011), or use crowdsourcing to label proto-roles, following Reisinger et al. (2015).

$wh \in$ {Who, What} $\wedge$ voice $=$ active		
WH $\rightarrow$ R0	WH $\rightarrow$ R1	WH $\rightarrow$ R2[p]
SBJ $= \phi$	SBJ $\rightarrow$ R0	SBJ $\rightarrow$ R0
OBJ1 $\rightarrow$ R1	OBJ1 $= \phi$	OBJ1 $\rightarrow$ R1
OBJ2 $\rightarrow$ R2[p]	OBJ2 $\rightarrow$ R2[p]	OBJ2 $= \phi$
$wh \in$ {Who, What} $\wedge$ voice $=$ passive		
WH $\rightarrow$ R1	WH $\rightarrow$ R2	
SBJ $= \phi$	SBJ $\rightarrow$ R1	
OBJ1 $\rightarrow$ R2	OBJ1 $= \phi$	
OBJ2 $\rightarrow$ R2[p]	OBJ2 $\rightarrow$ R2[p]	
$wh \in$ {When, Where, Why, How, HowMuch} $\wedge$ voice $=$ active		
WH $\rightarrow wh[p]$	WH $\rightarrow wh$	
SBJ $\rightarrow$ R0	SBJ $\rightarrow$ R0	
OBJ1 $\rightarrow$ R1	OBJ1 $\rightarrow$ R1	
OBJ2 $= \phi$	OBJ2 $\rightarrow$ R2[p]	
$wh \in$ {When, Where, Why, How, HowMuch} $\wedge$ voice $=$ passive		
WH $\rightarrow wh[p]$	WH $\rightarrow wh$	
SBJ $\rightarrow$ R1	SBJ $\rightarrow$ R1	
OBJ1 $\rightarrow$ R2	OBJ1 $\rightarrow$ R2	
OBJ2 $= \phi$	OBJ2 $\rightarrow$ R2[p]	

Table 7: Mapping question fields to roles in $\mathcal{R}$. The mapping is based on whether certain question fields are empty and the voice of the verb in the question (active or passive). ϕ indicates that a field is either an empty string or equals "do/doing". If a question is in passive voice and contains the preposition "by", then **OBJ2** is tagged with R0 instead, as in *What is built by someone?*

Predicting Question Roles Given this space of possible roles, our first step in generation is to determine which roles are present in a sentence, and select the pronouns that could be used to refer to them in the resulting questions. We formulate this task as a supervised multi-label learning problem. We define the set of possible labels $\mathcal{L}$ by combining the roles in $\mathcal{R}$ with different pronoun values:

$$\begin{aligned} \mathcal{L} =& \{role{:}val \mid role \in \mathcal{R}\} \\ val \in& \{\phi, \text{someone, something, do something,} \\ & \quad \text{doing something}\} \end{aligned}$$

For example, to support the generation of the questions *Who finished something?* and *What did someone finish?*, we need to first predict the labels R0:someone and R1:something. Adjunct roles, such as *When* and *How*, always take an empty pronoun value.

[6] The task also has applications to semi-automatic annotation of sentences with our scheme, if we could generate questions with high enough recall and only require annotators to provide all the answers. We leave this important direction to future work.

Question	Abstract Question				
	WH	**SBJ**	**Voice**	**OBJ1**	**OBJ2**
Who finished something?	R0	/	active	R1	/
What did someone finish?	R1	R0	active	/	/

Table 8: Example surface realization templates from abstract questions.

For each sentence s and verb v, the set of positive training samples corresponds to the set of labels in the annotated questions, and the negative samples are all the other labels in $\mathcal{L}_{train}$, the subset of labels appeared in training data.[7] We train a binary classifier for every label in $\mathcal{L}_{train}$ using L2-regularized logistic regression by Liblinear (Fan et al., 2008), with hyper-parameter $C = 0.1$. Features of the binary classifiers are listed in Table 10. For each sentence s and verb v in the test data, we take the k highest-scoring labels, and generate questions from these.

Question Generation After predicting the set of labels for a verb, we generate a question to query each role. First, we define the concept of an *abstract question*, which provides a template that specifies the role to be queried, other roles to include in the question, and the voice of the verb. Abstract questions can be read directly from our training data.

We can map an abstract question to a surface realization by substituting the slots with the pronoun values of the predicted labels. Table 8 shows the abstract questions we could use to query roles R0 and R1; and the generated questions, based on the set of predicted labels {R0:someone, R1:something}.

Therefore, to generate a question to query a role $r \in \mathcal{R}$, we simply return the most frequent abstract question that occurred in training data that matches the role being queried, and the set of other predicted labels.

Experiments Native English speakers manually evaluated 500 automatically generated questions (5 questions per verb). Annotators judged whether the questions were grammatical [8] and answerable from the sentence.

We evaluated the top k questions produced by our baseline technique. The results in Table 9 show that our system is able to produce questions which are both grammatical and answerable. The average number of QA pairs per verb collected by human annotator is roughly 2.5, demonstrating significant room for improving these results.

	Newswire		**Wikipedia**	
	Ans.	**Gram.**	**Ans.**	**Gram.**
prec@1	66.0	84.0	72.0	90.0
prec@3	51.3	78.7	53.3	86.0
prec@5	38.4	77.2	40.0	82.0

Table 9: Manual evaluation results for question generation in two domains, including the averaged number of distinct questions that are answerable given the sentence (Ans.) and the averaged number of questions that are grammatical (Gram.).

5 Answer Identification

The goal of the answer identification task is to predict an answer a given sentence s, target verb v and a question q. Our annotated answers can be a series of spans, so the space of all possible answers is $2^{|s|}$. To simplify the problem, we transform our span-based answer annotation to answer head words, thus reducing the answer space to $|s|$. We model whether a word is the head of an answer as a binary classification problem.

Each training sample is a tuple $\langle s, v, q, a, \pm 1 \rangle$. The answer head a is extracted from the k-best dependency parses and the annotated answer span. Given a dependency tree, if any word in the annotated answer span has a parent coming from outside the span, then it is considered an answer head. Therefore, a gold question-answer pair can be transformed into multiple positive training samples. The negative samples come from all the words in the sentence that are not an answer head. For learning, we train a binary classifier for every word in the sentence (except for the verb v).

Experiments We use L2-regularized logistic regression by Liblinear (Fan et al., 2008) for binary classification. Features are listed in Table 10.

The performance of our answer identification approach is measured by accuracy. For evaluation, given each test sentence s, verb v and question q, we output the word with highest predicted score using the binary classifier. If the predicted word is contained inside the annotated answer span, it is considered a correct prediction. We also use the

[7] We pruned the negative samples that contain prepositions that are not in the sentence or in the set of frequently-used prepositions (by, to, for, with, about).

[8] Some automatically generated questions are ungrammatical because of label prediction errors, such as *Who sneezed someone?*, where the label R1:someone shouldn't be predicted.

Feature Class	Question Generation	Answer Identification
Predicate	Token, Predicted POS-tag, Lemma extracted from Wiktionary	
	Dependency parent and edge label, dependency children and edge label	
Question	Question role label, Wh-word, Preposition	
Answer Word	/	Syntactic parent and edge label, Left/Right-most syntactic children,
Predicate-Answer	/	Relative position (left or right), Syntactic relation, Syntactic path

Table 10: Indicator features that are included in our role classifiers for question generation (Section 4) and the answer identification classifier (Section 5). Many come from previous work in SRL (Johansson and Nugues, 2008; Xue and Palmer, 2004). To mitigate syntactic errors, we used 10-best dependency parses from the Stanford parser (Klein and Manning, 2003).

	Newswire	Wikipedia
Classifier	78.7	82.3
Random	26.3	26.9

Table 11: Answer identification accuracy on newswire and Wikipedia text.

baseline method that predicts a random syntactic child from the 1-best parse for each question.

In each of the two domains, we train the binary classifiers on the training set of that domain (See Table 4 for dataset size). Table 11 shows experiment results for answer identification. Our classifier-based method outputs a correct answer head for 80% of the test questions, establishing a useful baseline for future work on this task.

6 Discussion and Future Work

We introduced the task of QA-SRL, where question-answer pairs are used to specify predicate-argument structure. We also presented a scalable annotation approach with high coverage, as compared to existing SRL resources, and introduced baselines for two core QA-SRL subtasks: question generation and answering.

Our annotation scheme has a number of advantages. It is low cost, easily interpretable, and can be performed with very little training and no linguistic expertise. These advantages come, in large part, from the relatively open nature of the QA-SRL task, which does not depend on any linguistic theory of meaning or make use of any frame or role ontologies. We are simply using natural language to annotate natural language.

Although we studied verbal predicate-argument structure, there are significant opportunities for future work to investigate annotating nominal and adjectival predicates. We have also made few language-specific assumptions, and believe the annotation can be generalized to other languages—a major advantage over alternative annotation schemes that require new lexicons to be created for each language.

The biggest challenge in annotating sentences with our scheme is choosing the questions. We introduced a method for generating candidate questions automatically, which has the potential to enable very large-scale annotation by only asking the annotators to provide answers. This will only be possible if performance can be improved to the point where we achieve high recall question with acceptable levels of precision.

Finally, future work will also explore applications of our annotation. Most obviously, the annotation can be used for training question-answering systems, as it directly encodes question-answer pairs. More ambitiously, the annotation has the potential to be used for training parsers. A joint syntactic and semantic parser, such as that of Lewis et al. (2015), could be trained directly on the annotations to improve both the syntactic and semantic models, for example in domain transfer settings. Alternatively, the annotation could be used for active learning: we envisage a scheme where parsers, when faced with ambiguous attachment decisions, can generate a human-readable question whose answer will resolve the attachment.

Acknowledgments

This research was supported in part by the NSF (IIS-1252835), DARPA under the DEFT program through the AFRL (FA8750-13-2-0019), an Allen Distinguished Investigator Award, and a gift from Google. We are grateful to Kenton Lee and Mark Yatskar for evaluating the question generation task, and Eunsol Choi, Yejin Choi, Chloé Kiddon, Victoria Lin, and Swabha Swayamdipta for their helpful comments on the paper. We would also like to thank our freelance workers on oDesk/Upwork for their annotation and the anonymous reviewers for their valuable feedback.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 228–238.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th International Conference on Computational Linguistics*, volume 1, pages 86–90. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the Linguistic Annotation Workshop*.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the 2012 International Conference on Language Resources and Evaluation*, volume 12, pages 3196–3200.

Claire Bonial, Olga Babko-Malaya, Jinho D Choi, Jena Hwang, and Martha Palmer. 2010. Propbank annotation guidelines. *Center for Computational Language and Education Research, CU-Boulder*.

Desai Chen, Nathan Schneider, Dipanjan Das, and Noah A Smith. 2010. Semafor: Frame argument resolution with log-linear models. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 264–267. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 57–60. Association for Computational Linguistics.

Richard Johansson and Pierre Nugues. 2008. Dependency-based semantic role labeling of propbank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 69–78. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *Proceedings of the 2002 International Conference on Language Resources and Evaluation*. Citeseer.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1, pages 423–430. Association for Computational Linguistics.

Joel Lang and Mirella Lapata. 2011. Unsupervised semantic role induction via split-merge clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1117–1126. Association for Computational Linguistics.

Mike Lewis, Luheng He, and Luke Zettlemoyer. 2015. Joint a* ccg parsing and semantic role labeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Ding Liu and Daniel Gildea. 2010. Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 716–724. Association for Computational Linguistics.

Bill MacCartney and Christopher D Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200. Association for Computational Linguistics.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The nombank project: An interim report. In *HLT-NAACL 2004 workshop: Frontiers in corpus annotation*, pages 24–31.

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

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 192–199. Association for Computational Linguistics.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van

Durme. 2015. Semantic proto-roles. *Transactions of the Association for Computational Linguistics*, 3:475–488.

Josef Ruppenhofer, Michael Ellsworth, Miriam RL Petruck, Christopher R Johnson, and Jan Scheffczyk. 2006. Framenet ii: Extended theory and practice.

Asher Stern and Ido Dagan. 2014. Recognizing implied predicate-argument relationships in textual inference. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Ivan Titov and Alexandre Klementiev. 2012. A bayesian approach to unsupervised semantic role induction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 12–22. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 173–180. Association for Computational Linguistics.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 88–94.

Deniz Yuret, Aydin Han, and Zehra Turgut. 2010. Semeval-2010 task 12: Parser evaluation using textual entailments. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 51–56. Association for Computational Linguistics.