

Crowdsourcing Question-Answer Meaning Representations

Julian Michael¹, Gabriel Stanovsky,^{*1,3} Luheng He¹, Ido Dagan², and Luke Zettlemoyer¹

¹Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA

²Bar-Ilan University Computer Science Department, Ramat Gan, Israel

³Allen Institute for Artificial Intelligence, Seattle, WA

{julianjm, luheng, lsz}@cs.washington.edu
gabriel.stanovsky@gmail.com, dagan@cs.biu.ac.il

Abstract

We introduce Question-Answer Meaning Representations (QAMRs), which represent the predicate-argument structure of a sentence as a set of question-answer pairs. We develop a crowdsourcing scheme to show that QAMRs can be labeled with very little training, and gather a dataset with over 5,000 sentences and 100,000 questions. A qualitative analysis demonstrates that the crowd-generated question-answer pairs cover the vast majority of predicate-argument relationships in existing datasets (including PropBank, NomBank, and QA-SRL) along with many previously under-resourced ones, including implicit arguments and relations. We also report baseline models for question generation and answering, and summarize a recent approach for using QAMR labels to improve an Open IE system. These results suggest the freely available¹ QAMR data and annotation scheme should support significant future work.

1 Introduction

Predicate-argument relationships form a key part of sentential meaning representations, and support answering basic questions such as *who did what to whom*. Resources for predicate-argument structure are well-developed for verbs (e.g. PropBank (Palmer et al., 2005)) and there have been efforts to study other parts of speech (e.g. NomBank (Meyers et al., 2004) and FrameNet (Baker et al., 1998)) and introduce whole-sentence structures (e.g. AMR (Banarescu et al., 2013)). However, highly skilled and trained annotators are re-

Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

Who will **join** as **nonexecutive director**? - Pierre Vinken
What is **Pierre**'s last name? - Vinken
Who is **61 years old**? - Pierre Vinken
How **old** is **Pierre Vinken**? - 61 years old
What will he **join**? - the board
What will he **join the board** as? - nonexecutive director
What type of **director** will **Vinken** be? - nonexecutive
What day will **Vinken join the board**? - Nov. 29

Figure 1: Example QAMR.

quired to label data within these formulations for each new domain, and it takes significant effort to model each new type of relationship (e.g., noun arguments in NomBank). We propose a new method to annotate relatively complete representations of the predicate-argument structure of a sentence, which can be done easily by non-experts.

We introduce Question-Answer Meaning Representations (QAMRs), which represent the predicate-argument structure of a sentence as a set of question-answer pairs (see Figure 1). Following the QA-SRL formalism (He et al., 2015), each question-answer pair corresponds to a predicate-argument relationship. There is no need for a carefully curated ontology and the labels are highly interpretable. However, we differ from QA-SRL in focusing on *all words* in the sentence rather than just verbs, and allowing *free form* questions instead of using templates.

The QAMR formulation provides a new way of thinking about predicate-argument structure. Any form of sentence meaning—from a vector of real numbers to a logical form—should support the challenge of determining which questions are answerable by the sentence, and what the answers are. A QAMR sidesteps intermediate formal representations by surfacing those questions and an-

*Work performed while at Bar-Ilan University.

¹github.com/uwnlp/qamr

swers *as* the representation. As with any other representation, this can then be reprocessed for downstream tasks. Indeed, the question-answer format facilitates reprocessing for tasks that are similar in form, for example Open IE (see Section 4).

A key advantage of QAMRs is that they can be annotated with crowdsourcing. The main challenge is coverage, as it can be difficult for a single annotator to write all possible QA pairs for a sentence. Instead, we distribute the work between multiple annotators in a novel crowdsourcing scheme, which we use to gather a dataset of over 100,000 QA pairs for 5,000 sentences in Newswire and Wikipedia domains.

Although QAMR questions’ free-form nature is crucial for our approach, it means that predicates are not explicitly marked. However, with a simple predicate-finding heuristic, we can align QAMR to PropBank, NomBank, and QA-SRL and show high coverage of predicate-argument structure, including more than 90% of non-discourse relationships. Further analysis reveals that QAMRs also capture many phenomena that are not modeled in traditional representations of predicate-argument structure, including coreference, implicit and inferred arguments, and implicit relations (for example, with noun adjuncts).

Finally, we report simple neural baselines for QAMR question generation and answering. We also highlight a recent result (Stanovsky et al., 2018) showing that QAMR data can be used to improve performance on a challenging task: Open Information Extraction. Together, these results show that there is significant potential for follow up work on developing innovative uses of QAMR and modeling their relatively comprehensive and complex predicate-argument relationships.

2 Crowdsourcing

We gather QAMRs with a two-stage crowdsourcing pipeline² using monetary incentives and crowd-driven quality control to ensure high coverage of predicate-argument structure. *Generation* workers write QA pairs and *validation* workers answer or reject the generated questions. Full details of our setup are given in Appendix A.

Generation Workers receive an English sentence with up to four *target words*. They are asked to write as many QA pairs as possible containing

²Built using Amazon Mechanical Turk: www.mturk.com

	PTB	Train	Dev	Test
Sentences	253	3,938	499	480
Annotators	5	1	3	3
QA Pairs	27,082	73,561	27,535	26,994
Filtered	18,789	51,063	19,069	18,959
Cost	\$2,862	\$7,879	\$2,919	\$2,919
Cost/token	\$0.44	\$0.08	\$0.25	\$0.25

Table 1: Summary of the data gathered.

each target word in the question or answer, subject to light constraints (for example, the question must contain a word from the sentence and be answered in the sentence, and they must highlight the answer in the sentence). Workers must write at least one QA pair for each target word to receive the base pay of 20c. An increasing bonus of $3(k + 1)$ cents is paid for each k -th additional QA pair they write that passes the validation stage.

Validation Workers receive a sentence and a batch of questions written by an annotator in the first stage (with no marked target words or answers). The worker must mark each question as *invalid* or *redundant* with another question, or highlight its answer in the sentence. Two workers validate and answer each set of questions. They are paid a base rate of 10c for each batch, with an extra 2c for each question past four.

Quality control Question writers are disqualified if the percentage of valid judgments on their questions falls below 75%. Validators need to pass a qualification test and maintain above 70% agreement with others, where overlapping answer spans are considered to agree.

2.1 Data Preparation and Annotation

We drew our data from 1,000 Wikinews articles from 2012–2015 and 1,000 articles from Wikipedia’s 1,000 core topics,³ partitioned by document into train, dev, and test, and preprocessed using the Stanford CoreNLP tools (Manning et al., 2014). We also annotated 253 sentences from the Penn Treebank (Marcus et al., 1993) chosen to overlap with existing resources for comparison (see Section 3).

For each sentence, we group its non-stopwords sequentially into groups of 3 or 4 target words, removing sentences with no content words. By presenting workers with nearly-contiguous lists of

³https://en.wikipedia.org/wiki/Wikipedia:1,000_core_topics

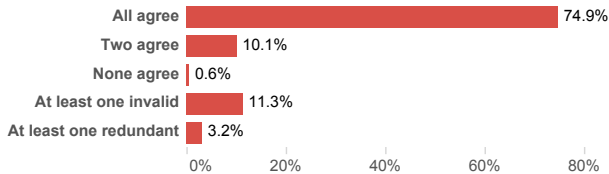


Figure 2: Agreement and validation statistics on all data gathered. Answers were considered to agree if their spans overlapped. High agreement on answers indicates that questions were generally interpretable and answers were consistent.

target words, enforcing non-redundancy, and providing bonuses, we encourage exhaustiveness over all possible QA pairs. By allowing the target word to appear in the question *or* the answer, we make the requirements flexible enough that there is almost always some QA pair that can be written.

Figure 2 shows agreement statistics for question validation. We removed questions either validator counted invalid or redundant, as well as questions not beginning with a *wh*-word,⁴ which we found to be of low quality. We also annotated the partitions at different levels of redundancy to allow for more exhaustive dev, test, and comparison sets. See Table 1 for statistics.

3 Data Analysis

In this section, we show that QAMR has high coverage of predicate-argument structure and uses a rich vocabulary to label fine-grained and implicit semantic relations.

Coverage To show that QAMR captures the same kinds of predicate-argument relations as existing formalisms, we compare our data to PropBank, NomBank, and QA-SRL. Since predicates in the questions are not explicitly marked, we use a simple predicate-finding heuristic to help align to other formalisms: for each minimal span that appears in the QAMR questions and answers (i.e., none of its subspans appear independently of it elsewhere in the QAMR), we compute its *predicate score* as the proportion of its appearances that are in a question rather than in an answer.⁵ We then choose the span with the highest predicate score in each question as its predicate.

We measure recall on the shared Penn Treebank sentences for each resource by randomly sampling n annotators out of 5 for each group of target

⁴*who, what, when, where, why, how, which, and whose*

⁵This follows the intuition that predicates are more likely to appear in the question; for example, see **join** in Figure 1.

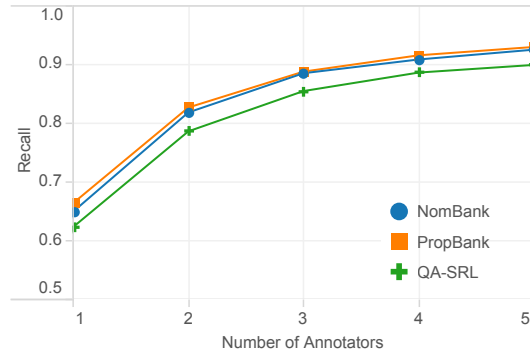


Figure 3: Recall of predicate-argument relations for sentences shared with each of our reference datasets, with increasing number of annotators.

words, which simulates the situation for the training set (1 annotator) and the dev/test sets (3 annotators). For each n we took the mean of 10 runs. Full details of our comparison are in Appendix B.

Results are shown in Figure 3. Single annotators cover over 60% of relationships, and coverage quickly increases with the number of annotators, reaching over 90% with all five. This shows that QAMR’s representational capacity covers the vast majority of relevant predicate-argument relations in existing resources. However, coverage in our training set is low due to low annotation density.

For a qualitative analysis, we sample 150 QA pairs (see Table 2 for examples).⁶ Of our sample, over 90% of question-answer pairs correspond to a predicate-argument relation expressed in the sentence,⁷ including arguments and modifiers of nouns and verbs as well as relationships like those within proper names (Table 2, ex. 2c, 3a) and coreference (ex. 3c, 4c). Questions that do not align to predicate-argument structure often target shallow inferences (ex. 3b, 7c).

Rich vocabulary Annotators use the open question format to introduce a large vocabulary of *external phrases* which do not appear in the sentence. Overall, 5,687 different external phrases are introduced (excluding stopwords), appearing 25,952 times in 38.7% of the questions (see Figure 4). These include typing words like *state* and *country* (Table 2, ex. 5), most often directly after the *wh*-word, and relation-denoting phrases like *work for* (ex. 2b), *last name* (ex. 3a), and *victim*

⁶This sample, and statistics for the remainder of this section, were taken from the filtered train and development sets, with a total of about 70k QA pairs.

⁷We assume a QA pair targets the relation corresponding to the semantic role of the *wh*-word in the question.

Sentence	Ann.	Question	Answers
(1) Climate change affects distribution of weeds, pests, and diseases.	VAR	(a) What affects distribution of diseases? (b) What is affected?	Climate change distribution of... / distribution
(2) Baruch ben Neriah, Jeremiah’s scribe, used this alphabet to create the later scripts of the Old Testament.	SYN ROLE	(a) Who wrote the scripts? (b) Who did Baruch work for? (c) What is old?	Baruch ben Neriah Jeremiah Testament / the Old Testament
(3) Mahlunga has said he did nothing wrong and Judge Horn said he “failed to express genuine remorse”.	ROLE INF CO	(a) What is the Judge’s last name? (b) Who doubted his remorse was genuine? (c) Who didn’t express genuine remorse?	Horn Judge Horn Mahlunga
(4) In Byron’s later memoirs, “Mary Chaworth is portrayed as the first object of his adult sexual feelings.”	IMP VAR	(a) Who is portrayed in the work? (b) Who was the object of his sexual feelings? (c) Who was Mary the object of sexual feelings for?	Mary Chaworth Mary Chaworth Byron
(5) Volunteers are presently renovating the former post office in the town of Edwards, Mississippi, United States for the doctor to have an office.	IMP VAR	(a) What town is the post office in? (b) What state is the post office in? (c) What country are the volunteers renovating in? (d) What country is the city of Edwards in?	Edwards Mississippi United States United States
(6) The ossicles are the malleus (hammer), incus (anvil), and the stapes (stirrup).	VAR	(a) What is the malleus one of?	The ossicles / ossicles
(7) Liam “had his whole life in front of him”, said Detective Inspector Andy Logan, who was the senior investigator of his murder.	ROLE ROLE INF	(a) Who is the murder victim Logan is investigating? (b) What rank of investigator is Andy Logan? (c) Who was Detective Logan speaking about?	Liam Detective Inspector / senior Liam
(8) This cemetery dates from the time of Menkaure (Junker) or earlier (Reisner), and contains several stone-built mastabas dating from as late as the 6th dynasty.	INF IMP	(a) How old are the stone-built mastabas? (b) What period was earlier than Menkaure? (c) What dates from the 6th dynasty?	dating from as late as the 6th dynasty / from as late as the 6th dynasty Reisner mastabas / several stone-built mastabas

Table 2: Examples of question-answer pairs capturing various semantic relations, annotated with interesting phenomena they exhibit: syntactic variation (VAR), synonym use (SYN), explicit role names for implicit relations (ROLE), coreference (CO), implicit arguments (IMP), and inferred relations (INF).



Figure 4: Novel phrases appearing more than 50 times. Darker phrases appear more commonly after *who*, *which*, or *how*. The vast majority of external phrases are used to denote entity/event types or semantic relations.

(ex. 7a). Despite the open format, synonyms are not a major issue, obscuring the semantic relation in only 2% of our sample (ex. 2a).

We also find verbal paraphrases of noun compounds, as proposed by Nakov (2008). For example, where *Gallup poll* appears in the text, one annotator has written *Who conducted the poll?*, which explicates the relationship between *Gallup* and *poll*. Similarly, *Who received the bailouts?* is written for the phrase *bank bailouts*.

Semantics, not just syntax Only 63% of QA pairs characterize their predicate-argument relation using the same syntactic relationship as in the sentence. 5% have answers coreferent with the

syntactic argument (Table 2, ex. 3c, 4c); 17% exhibit syntactic variation, using different prepositions (ex. 4c, 6a), alternating between active and passive (ex. 1b), or changing between the noun and verb form of the predicate (ex. 8a); 6% ask about implicit arguments (ex. 4b, 5c, 8b); and 6% ask about inferred relations (ex. 3b).

4 Models

To establish initial baselines, we apply existing neural models for QAMR question generation and answering. We also briefly summarize a recent end task result, where QAMR annotations were used to improve an Open IE system.

Question generation In question generation (QG), we learn a mapping from a sentence w to a set of questions q_1, \dots, q_m . We enumerate pairs of words (w_q, w_a) from the sentence to seed the generator. During training, outputs are questions q and inputs are tuples (w, w_q, w_a) , where $w_q \in q$ and w_a is in q ’s answer. We also add negative samples where the output is a special token and the input has w_q, w_a that never appear together.

We use an encoder-decoder model with a copying mechanism (Zhou et al., 2017) to generate a question from an input sentence with tagging features for part of speech, w_q , and w_a . At test time,

we run all pairs of content words (w_i, w_j) where $|i - j| \leq 5$ through the model to yield a set of questions. On the QAMR test set, this achieves 28% precision and 24% recall with fuzzy matching (multi-BLEU⁸ > 0.8).

Question answering The format of QAMRs allows us to apply an existing question-answering model (Seo et al., 2016) designed for the SQuAD (Rajpurkar et al., 2016) reading comprehension task to answer QAMR questions. Training and testing with the SQuAD metrics on QAMR, the model achieves 70.8% exact match and 79.7% F1 score. We further improve performance to 75.7% exact match and 83.9% F1 by pooling our training set with the SQuAD training data. The relative ease of QA in comparison to QG suggests that in QAMR, most of the information is contained in the questions.

Open IE Finally, we also expect that the predicate-argument relationships represented in QAMRs will be useful for many end tasks. Such a result was recently shown for Open IE (Stanovsky et al., 2018), using our QAMR corpus. Open IE involves extracting tuples of natural language phrases that express the propositions asserted by a sentence. They show that, using a syntactic dependency parser, a QAMR can be converted to a list of Open IE extractions. Augmenting their training data with a conversion of our QAMR dataset yields state-of-the-art performance on several Open IE benchmarks (Stanovsky and Dagan, 2016b; Xu et al., 2013; de Sá Mesquita et al., 2013; Schneider et al., 2017). The gains come largely from the extra extractions (e.g., with nominal predicates) that QAMRs support over traditional resources focusing on verbal predications.

5 Related Work

In addition to the semantic formalisms (Palmer et al., 2005; Meyers et al., 2004; Banarescu et al., 2013; He et al., 2015) we have already discussed, FrameNet (Baker et al., 1998) also focuses predicate-argument structure, but has more fine-grained argument types. Gerber and Chai (2010) target implicit nominal arguments. Stanovsky and Dagan (2016a) annotate non-restrictive noun phrase modifiers on top of QA-SRL. Other linguistically motivated annotation schemes include

UCCA (Abend and Rappoport, 2013), HSPG treebanks (Flickinger et al., 2017), and the Groningen meaning bank (Basile et al., 2012).

Crowdsourcing has also been applied to gather annotations of structure in the setup of multiple choice questions, for example, for Dowty’s semantic proto-roles (Reisinger et al., 2015; White et al., 2016) and human-in-the-loop parsing and classification (He et al., 2016; Duan et al., 2016; Werling et al., 2015), while Wang et al. (2017) use crowdsourcing with question-answer pairs to annotate some PropBank roles directly. Our approach recovers paraphrases of noun compounds similar to those crowdsourced by Nakov (2008).

More broadly, non-expert annotation has been used extensively to gather question-answer pairs over natural language texts, for example in reading comprehension (Rajpurkar et al., 2016; Richardson et al., 2013; Nguyen et al., 2016) and visual question answering (Antol et al., 2015). However, while these treat question answering as an end task, we regard it as a representation of predicate-argument structure, and focus annotators on a smaller selection of text (a few target words in a single sentence, rather than a paragraph) aiming to achieve high coverage.

6 Conclusion

QAMR provides a new way of thinking about meaning representation: using open-ended natural language annotation to represent rich semantic structure. This paradigm allows for representing a broad range of semantic phenomena with data easily gathered from native speakers. Our dataset has already been used to improve the performance of an Open IE system, and how best to leverage the data and model its complex phenomena is an open challenge which our annotation scheme could support studying at a relatively large scale.

Acknowledgments

This work was supported in part by grants from the MAGNET program of the Israeli Office of the Chief Scientist (OCS); the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1); the Israel Science Foundation (grant No. 1157/16); the US NSF (IIS1252835, IIS-1562364); and an Allen Distinguished Investigator Award.

⁸An average of the BLEU1–BLEU4 scores.

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*.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.
- 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*.
- 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*.
- Manjuan Duan, Ethan Hill, and Michael White. 2016. Generating disambiguating paraphrases for structurally ambiguous sentences. In *Proceedings of the 10th Linguistic Annotation Workshop*.
- Dan Flickinger, Stephan Oepen, and Emily M. Bender. 2017. *Sustainable Development and Refinement of Complex Linguistic Annotations at Scale*.
- Matthew Gerber and Joyce Y Chai. 2010. Beyond nombank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592. Association for Computational Linguistics.
- Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.
- Luheng He, Julian Michael, Mike Lewis, and Luke Zettlemoyer. 2016. Human-in-the-loop parsing.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. *The Stanford CoreNLP natural language processing toolkit*. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*.
- Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekeley, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The nombank project: An interim report. In *HLT-NAACL 2004 workshop: Frontiers in corpus annotation*.
- Preslav Nakov. 2008. *Noun Compound Interpretation Using Paraphrasing Verbs: Feasibility Study*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- 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*. In *Advances in Neural Information Processing Systems*.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *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*.
- 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*.
- Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, pages 193–203.
- Filipe de Sá Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 447–457.
- Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Loser. 2017. Analysing errors of open information extraction systems. *CoRR*, abs/1707.07499.
- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.
- Gabriel Stanovsky and Ido Dagan. 2016a. Annotating and predicting non-restrictive noun phrase modifications. In *Proceedings of the 54rd Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.
- Gabriel Stanovsky and Ido Dagan. 2016b. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, Texas. Association for Computational Linguistics.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Chenguang Wang, Alan Akbik, Laura Chiticariu, Yunyao Li, Fei Xia, and Anbang Xu. 2017. [Crowd-in-the-loop: A hybrid approach for annotating semantic roles](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1923. Association for Computational Linguistics.

Keenon Werling, Arun Tejasvi Chaganty, Percy S Liang, and Christopher D Manning. 2015. On-the-job learning with bayesian decision theory. In *Advances in Neural Information Processing Systems*.

Aaron Steven White, Drew Reisinger, Sakaguchi Tim Vieira, Sheng Zhang Rachel Rudinger Kyle Rawlins, and Benjamin Van Durme. 2016. Universal compositional semantics on universal dependencies.

Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. 2013. [Open information extraction with tree kernels](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 868–877, Atlanta, Georgia. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. *arXiv preprint arXiv:1704.01792*.

A Crowdsourcing Details

In this section we provide full details of the data collection methodology described in Section 2. For the exact text of the instructions shown to workers, and code to reproduce the annotation or demo the interface, see github.com/uwnlp/qamr.

Stages Data collection proceeded in two stages: *generation* and *validation*. These were run as two types of HITs (Human Intelligence Tasks) on the Amazon Mechanical Turk platform. Workers wrote questions and answers for the generation task, and those questions would be immediately uploaded as new HITs for the validation task, which ran concurrently. Two workers would validate each question. The worker writing the question would be assessed based on the validators’ judgments, and the validators would be assessed

based on their agreement. In this way, the quality of workers in either stage could be quickly assessed so spammers or low-quality workers could be disqualified before causing much damage.

Question constraints In both stages, we define a *valid* question to

- (1) contain at least one word from the sentence,
- (2) be about the sentence’s meaning,
- (3) be answered obviously and explicitly in the sentence,
- (4) not be a yes/no question, and
- (5) not be *redundant*,

where we define two questions as being redundant by the informal criterion of “having the same meaning” and the same answer. These requirements are illustrated with examples.

Workers in the generation phase are instructed only to write valid questions, while workers in the validation phase are instructed only to answer valid questions (marking the rest invalid or redundant).

When we ask that the question contains a word from the sentence, we allow for changing forms of the word through inflectional or derivational morphology (with examples of both). The only constraint on questions that is strictly enforced by the interface is a length limit of 50 characters.

Target words In the generation task, each sentence is presented to the worker with several underlined *target words*. They are required to write at least one QA pair for each target word, where the target word must appear either in the question or the answer. We choose sets of target words by chunking consecutive words (ignoring stopwords) into groups of 3 or 4 (or fewer for very short sentences). Because the target words shown to a single worker are close to each other—and often share a constituent—it restricts the set of QA pairs they write to relate to a certain part of the sentence. However, asking that the target word appears either in the question *or* the answer makes it flexible enough so that the worker is almost never stuck with no reasonable question to write. We identified this approach after some experimentation, finding that together with the monetary incentives described below, it struck the appropriate

Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

Pierre

<input type="text" value="What is Pierre's last name?"/>	Vinken
<input type="text" value="When will Pierre join the board?"/>	Nov. 29
<input type="text" value="Question (+9c)"/>	

Vinken

<input type="text" value="Who will join the board?"/>	Pierre Vinken
<input type="text" value="Question (+9c)"/>	

61

<input type="text" value="How old is Pierre?"/>	--> 61 years old
<input type="text" value="Question (+9c)"/>	

years

<input type="text" value="Question (required)"/>
--

Figure 5: Annotation interface for generation.

balance of scope that was small enough to get exhaustive annotation, but not so small that it cornered workers into writing awkward questions or getting frustrated.

Interface In the generation stage, below the sentence, each target word is listed with a text field below it where they write a question for that target word. While the text field is focused, they highlight the answer tokens in the sentence using custom implemented highlighting functionality. The highlighted tokens then appear next to the focused question. The answer tokens need not be a contiguous span in our interface (though they almost always are in practice). Once a question is written and answered, a new text field appears directly below it for another question, allowing the annotator to write as many questions as they can. See Figure 5 for a screenshot of the generation task interface.

In the validation stage, no target words are indicated to the user; they only see a list of questions written in a single HIT by a worker in the generation stage. They use the arrow keys to switch between the questions, and use the mouse to assess them: either highlighting the answer in the sentence, clicking another question to mark the selected one redundant, or clicking the *invalid* button to mark a question invalid. See Figure 6.

Incentives and payment Base pay for the generation stage was 20c, with a bonus of $3(k + 1)$ cents for each question beyond the number required (so, the first extra question would reward them 6c, the next 9c, and so on). However, their

Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

<input type="button" value="Invalid"/>	What is Pierre's last name? -->
Highlight answer above, move with arrow keys, or click on a redundant question	
<input type="button" value="Invalid"/>	When will Pierre join the board?
<input type="button" value="Invalid"/>	Who will join the board?
<input type="button" value="Invalid"/>	How old is Pierre?
<input type="button" value="Invalid"/>	Who is 61 years old?
Bonus: 2c	

Figure 6: Annotation interface for validation.

bonuses were only calculated based on the number of questions considered valid by annotators. So if a worker in the generation task wrote 2 extra questions, but any 2 (or 3, or more) of their questions were judged invalid, then they would receive no bonus.

In the validation stage, workers were paid 10c plus a bonus of 2c per question beyond four.

Quality control We used Mechanical Turk's quality control mechanisms in several ways. First, we used the built-in Locale qualification to limit the tasks to workers based in the United States as a proxy for English proficiency. Second, we wrote a multiple-choice qualification test for the validation task, which tested workers' understanding of the definitions of question validity and redundancy. Workers were required to get a score of 75% on this test before working on the validation task.

Finally, we used Mechanical Turk's built-in qualification mechanism to keep track of worker accuracy and agreement ratings. Before working on either task, a worker would have to request a qualification which stored their accuracy or agreement value. Then as they worked, it would be updated over time and they could check its value in their Mechanical Turk account to see how they were doing. In the generation task, accuracy was calculated as the proportion of all judgments (aggregating those from both validators) that were not *invalid* or *redundant*, and accuracy had to remain above 75% to avoid disqualification. In the validation task, agreement was calculated by treating answer spans as agreeing if they had any overlap, and *redundant* judgments agreeing if their targets had agreeing answer spans. A worker's agreement had to stay above 70% for them to remain qualified.

If a worker's accuracy or agreement rating

dropped within 5% of the threshold, the worker was automatically sent an email with a warning and a list of common mistakes and tips they might use to improve.

Implementation All of our code was written in Scala, using the Java AWS SDK on the backend to interface with Mechanical Turk, Akka Actors and Akka HTTP to implement the web server and quality control logic, and Scala.js with React to implement the user interface.

Dataset Our dataset was gathered over the course of 1 month from 330 unique workers. See Section 2.1 for details.

B SRL Comparison

In this section we provide the full details of the comparison of QAMR to PropBank, NomBank, and QA-SRL given in Section 3.

Preprocessing For each of these resources, there were certain predicate-argument relationships that we filtered out of the comparison for being out of scope.

For PropBank, we filter out predicates and arguments that are auxiliary verbs, as well as reference (R-) roles since aligning these properly is difficult and their function is primarily syntactic. We also remove discourse (-DIS) arguments such as *but* and *instead*: these may be regarded as involved in *discourse* structure separately from the predicate-argument structure we are investigating. 78% of the original dependencies remain.

For NomBank, we also remove auxiliaries, and we remove arguments that include the predicate—which are present for words like *salesman* and *teacher*—leaving 83% of the original dependencies.

For QA-SRL, we use all dependencies, and where multiple answers were provided to a question, we take the union of the answer spans to be the argument span.

Alignment Because QAMR does not mark predicates explicitly, we use a simple predicate-finding heuristic to align the QA pairs in a QAMR to the predicate-argument relations in each resource independently.

For each QAMR, we identify every *minimal span* appearing in its questions and answers, i.e., a span from the sentence where none of its subspans appear independently of it in the QAMR.

We then calculate a *predicate score* for each span, as the proportion of times it appeared in a question versus an answer. Then for each QA pair, we identify the span in the question with highest predicate score as its *predicate span*, and the answer as its argument span. This is then aligned to the predicate-argument arc in the chosen resource with the greatest non-zero argument overlap such that the predicate is contained within the question’s predicate span. If there is no such alignment, we check for an opposite-direction alignment where the predicate is in the answer of a QA pair and the argument completely contains the question’s predicate span.

Results See Section 3 for a description of the results. With 1 annotator, we get around 60% recall, but it begins to level off over 85% with 3 annotators.

We manually examined 25 sentences to study sources of coverage loss in the 5-annotator case. In comparison to PropBank and NomBank, the missing dependencies are due to missing QA pairs (44%), mistakes in our alignment heuristic (28%), and subtle modifiers/idiomatic uses (28%). For example, annotators sometimes overlook phrases such as *so far* (marked as a temporal modifier in PropBank) or *let’s* (where *’s* is marked as a core verbal argument). Comparing to QA-SRL, 60% of the missed relations are inferred/ambiguous relations that are common in that dataset. Missed QA pairs in QA-SRL account for another 20%.

In aggregate, these analyses show that the QAMR labels capture the same kinds of predicate-argument structures as existing resources. However, while our development and test sets can be expected to have reasonable coverage, where we have labels from only one annotator for each target word (as in our training set), the recall low compared to expert-annotated structures, which may pose challenges to learning.