QANom: Question-Answer driven SRL for Nominalizations

Ayal Klein[1] Jonathan Mamou[2] Valentina Pyatkin[1] Daniela Brook Weiss[1]

Hangfeng He[3] Dan Roth[3] Luke Zettlemoyer[4] Ido Dagan[1]

[1]Computer Science Department, Bar Ilan University
[2]Intel Labs, Israel
[3]University of Pennsylvania
[4]University of Washington
{ayal.s.klein,jonathan.mamou,valpyatkin,
daniela.stepanov}@gmail.com
{hangfeng,danroth}@seas.upenn.edu
lsz@cs.washington.edu dagan@cs.biu.ac.il

Abstract

We propose a new semantic scheme for capturing predicate-argument relations for nominalizations, termed QANom. This scheme extends the QA-SRL formalism (He et al., 2015), modeling the relations between nominalizations and their arguments via natural language question-answer pairs. We construct the first QANom dataset using controlled crowdsourcing, analyze its quality and compare it to expertly annotated nominal-SRL annotations, as well as to other QA-driven annotations. In addition, we train a baseline QANom parser for identifying nominalizations and labeling their arguments with question-answer pairs. Finally, we demonstrate the extrinsic utility of our annotations for downstream tasks using both indirect supervision and zero-shot settings.

1 Introduction

Semantic Role Labeling (SRL) is the prominent representation for annotating predicate-argument structures. SRL annotations were shown useful for various downstream tasks, such as machine comprehension (Wang et al., 2015), cross-document coreference (Barhom et al., 2019), dialog (Chen et al., 2013) and summarization (Trandabăţ, 2011). Traditionally, SRL research is biased toward focusing on verbal predicates, as evident by their dominance in large scale semantic resources, such as PropBank (Kingsbury and Palmer, 2002), FrameNet (Baker et al., 1998) and OntoNotes (Pradhan et al., 2013), and consequentially among SRL models (He et al., 2018; Tan et al., 2018; Strubell et al., 2018).

Nevertheless, other types of predicates, such as nominalizations, are frequent in natural language, which also draw some research attention (Hajic et al., 2009; Jiang and Ng, 2006; Zhao and Titov, 2020). As a significant milestone, the NomBank initiative (Meyers et al., 2004a) provided extensive predicate-argument annotations for various types of nouns. In particular, since *deverbal nominalizations* share an underlying argument structure with their verbal counterparts, NomBank annotates these by applying the same annotation scheme as PropBank. A verb-derived noun will be mapped to a frame file shared by PropBank's verbal predicates, and accordingly a shared role-set. This design principle is meant for converging the semantic role representation of deverbal nominalizations with their corresponding verbal predicates, thus abstracting semantic content over surface realization specifics (Meyers et al., 2004b).

Annotating SRL resources involves substantial effort and cost. This hinders research progress, as it is hard to extend large-scale resources to additional text genres and languages. In order to address this data collection barrier, question-answer driven semantic role labeling (QA-SRL) (He et al., 2015) was proposed as a natural, easily attainable formulation of SRL. QA-SRL labels each predicate-argument relationship with a question-answer pair, where natural language questions represent semantic roles, and answers correspond to arguments (see Table 1 for a comparative illustration). This format yields rich and

Proceedings of the 28th International Conference on Computational Linguistics, pages 3069–3083
Barcelona, Spain (Online), December 8-13, 2020

Thomas has **proved** in different ways that God exists, including an argument dubbed "the Ontological argument".				
PropBank	ARG0	QA-SRL	Who has proved something?	Thomas
	ARG1		What has someone proved?	that God exists
	ARGM-MNR		How did someone prove something?	in different ways | the Ontological argument
Thomas has provided different **proofs** for the existence of God, including an assertion dubbed "the Ontological argument".				
NomBank	ARG0	QANom	Who has proved something?	Thomas
	ARG1		What has someone proved?	the existence of God
	-		How did someone prove something?	the Ontological argument

Table 1: An illustration of PropBank, NomBank, QA-SRL and QANom annotations for corresponding semantic information. Implicit arguments are highlighted, captured by the QA formalisms but not the others. The bar (|) separates multiple answers.

easily interpretable semantic annotations, facilitating scalable crowdsourced annotation methodologies, both for large train sets (Fitzgerald et al., 2018) and for high-quality evaluation sets (Roit et al., 2020). QA-SRL was shown to cover most predicate-argument structures captured by PropBank (He et al., 2015; Roit et al., 2020), to subsume popular intermediate representations (Stanovsky and Dagan, 2016), and recently, to enhance strong transformer-based sentence encoders (He et al., 2020) (§2.2).

In this work, we further pursue the overarching goal of developing a broad-coverage structured representation of sentence semantics through a natural, easily interpretable, and scalably attainable annotation scheme, following the QA-SRL paradigm. We introduce *QA-SRL for Nominalizations*, denoted QANom, as the most natural first extension of verbal QA-SRL (See Table 1). Analogical to the original NomBank motivation, we wish to construct a unified question-answer based scheme for verbal and nominal predicates. We identify verbal nouns that are eventive in nature along with their corresponding verbal predicates, and label their arguments using verb-centric questions adhering to the QA-SRL format.

Our contributions are outlined as follows: (1) We propose a novel SRL representation for deverbal nominalizations, QANom, which extends and complements the QA-SRL paradigm; (2) We present an annotation methodology for crowdsourcing QANom data with low costs yet good quality; (3) We collect the first QANom dataset, consisting of over 10K sentences and 26K QA-pairs, and assess its internal quality by measuring annotation consistency and comparing it to the expert-annotated NomBank dataset and other related resources; (4) We present an end-to-end QANom baseline system and evaluate it to serve as baseline for future parsers; (5) Finally, we demonstrate QANom's external utility for improving downstream applications, both through indirect supervision for enhancing pretrained sentence encoders, and as a proxy or signal for event extraction in a zero shot setting.[1]

2 Background

2.1 Nominal SRL

The most commonly used resource of English predicate-argument structure is the Propositional Bank (PropBank) (Kingsbury and Palmer, 2002), which assigns semantic role labels (e.g. **ARG1**, **ARG2**, **ARGM-TMP**) to arguments of verbal predicates in the Penn Treebank (PTB) corpus. PropBank's roles loosely correspond to thematic roles (**Agent**, **Patient**, **Time**, etc.) but are defined per frame (i.e. verb-sense).

With the goal of complementing and unifying PropBank with information about nominal predicates, NomBank was introduced (Meyers et al., 2004a), annotating every argument-taking noun in the PTB with a noun sense and corresponding semantic roles. For verb-derived nouns (e.g. *proof — prove*), a noun-sense is combined with its matching verb-sense to form a shared frame and a corresponding role-set, allowing the representation to generalize over predicate lexical category (see illustration in Table 1). Indeed, these overlapping argument structures of verbs and nouns were utilized by different machine learning techniques, e.g. natural language generation (Mille et al., 2018) and transfer learning (Padó et al., 2008; Zhao and Titov, 2020).

[1] Our dataset, guidelines and all relevant software can be found in `https://github.com/kleinay/QANom`.

About 75% of NomBank instances are deverbal nominalizations.[2] In addition, NomBank consists of many non-verbal types of argument-taking nouns (e.g. PARTITIVE, as in *a **set** of meetings*, or RELATIONAL, as in *John's older **brother***). These are out of scope for this work and are left for future research.

Annotating the large-scale NomBank resource was a tremendous endeavor,[3] which included the devising of an extensive annotation manual (Meyers, 2007) (126 pages), handcrafting of frame files, expert annotator training and of course the annotation process itself. In this work, we propose a much lighter-weight QA-based formalism which can be annotated at scale by employing lightly-trained crowdworkers, making it easily transferable to new domains or even languages. We also show that our annotations cover most of the semantic information in NomBank nominalizations (§4.2.2).

2.2 QA-driven Semantic Approach

With the goal of collecting natural, laymen-intuitive semantic annotations, QA-SRL (He et al., 2015) have been proposed, suggesting an alternative to traditional verbal SRL schemes. In QA-SRL, a Question and Answer pair (QA) about a sentence captures a single predicate-argument relation. Each sentence is preprocessed with a Part-of-Speech tagger to identify all non-copular verbs. A verbal predicate is then annotated with a list of QA pairs, where each is comprised of a role question and (one or more) contiguous answer spans from the sentence (for specifications see §8.1 in the Appendix). Fully utilizing the intuitive nature of QA-SRL, Fitzgerald et al. (2018) crowdsourced a large scale QA-SRL corpus via Amazon Mechanical Turk, and released the first QA-SRL parser.

QA-SRL is appealing not just for its scalability, but also for its content. By relying on natural comprehension of the sentence, the QA format elicits a richer argument set than traditional linguistically-rooted formalisms, including many valuable *implicit* arguments not manifested in syntactic structure (Roit et al., 2020). Although the importance of implicit relations has been established (Cheng and Erk, 2018; Do et al., 2017; Gerber and Chai, 2012), most SRL resources leave them out of scope.

QA-SRL was also proved beneficial for downstream processing. It was shown to subsume open information extraction (OIE) (Stanovsky and Dagan, 2016), which enabled constructing a large supervised OIE dataset (Stanovsky et al., 2018) to serve as an intermediate structure for end applications. Additionally, QA-SRL — as well as related QA-based semantic annotations (Michael et al., 2018) — were recently shown to improve downstream tasks by providing additional semantic signal through indirect supervision for modern pretrained-LM encoders (He et al., 2020). Overall, QA-SRL is shown to subsume traditional predicate-argument information (He et al., 2015; Roit et al., 2020) — which in turn has exhibited downstream utility for various tasks, such as machine comprehension (Wang et al., 2015), cross-document coreference (Barhom et al., 2019), dialog (Chen et al., 2013) and summarization (Trandabăţ, 2011).

A related QA-driven semantic annotation dataset is QA-driven Meaning Representation (QAMR) (Michael et al., 2018). While QAMR is also covering predicate-argument relations, it takes a different approach from QA-SRL and allows free-formed questions, requiring only to mention the target word. This results in a highly rich yet less systematic and loosely structured representation. In this work we follow the more constrained QA-SRL approach, and show that it produces better coverage of nominalizations' arguments (§4.3).

3 QANom — Representation and Dataset Construction

In this section, we describe our proposal for representing predicate-argument information regarding nominalizations. The fundamental rationale is that verb-derived nouns commonly share the argument structure of their verb counterpart.[4] Hence, it is natural to interrogate their arguments with verb-headed questions, much the same way QA-SRL captures verbal arguments. As an example, consider the two sentences in Table 1. The participants of the event denoted by the noun *proofs* are the same as those of

[2]Throughout this paper, we distinguish a *verbal* vs. *non-verbal* NomBank predicate by mapping it to its lexical entry in the NomLex dictionary (Macleod et al., 1998), leveraging its NOM-TYPE attribute. A predicate is considered verbal iff NOM-TYPE = VERB-NOM.

[3]They report initiating writing the specification at 01/2003. First version was released at 12/2007.

[4]Technically, we are not interested in the derivational process, but only care whether the noun has a verbal counterpart.

the equivalent *proven* event. Consequentially, they are naturally captured by the same role labels, both for PropBank and NomBank, and analogously for QA-SRL and QANom, resulting in a unified representation that can facilitate abstraction and generalization in downstream tasks (Meyers et al., 2004a).

3.1 Annotation Scope

Semantic role labeling involves three sub-tasks — detect predicates in text, extract their arguments, and label each with a semantic role. While verb predicates are relatively easy to detect, requiring no more than a POS tagger, this phase is not trivial for nominal SRL.

For QANom, the predicate detection task is coupled with finding a corresponding verb whose meaning is aligned to the nominal predicate. In this way, the QA-SRL questions centered by this verb could naturally capture arguments of the noun. Thus, our data collection setting involves leveraging lexical resources to find candidate nouns having a related verb. We use CatVar (Habash and Dorr, 2003), WordNet (Miller, 1995) and an in-house suffix-driven heuristic to identify those noun candidates, along with their corresponding verb. More details about our lexical candidate extraction procedure are in Appendix 8.2.

During annotation, workers first determine whether a candidate noun mention carries verbal meaning in the context of the sentence (ISVERBAL). We instructed the workers to consider the automatically extracted related verb, and to judge whether it will be natural to ask questions about the target noun instance using this verb. For example, given the noun phrase *"the **organization** of conferences and seminars"*, **organization** carries a verbal meaning with respect to the related verb **organize**, which may be used to ask *What does someone organize?* or similar questions. Conversely, the nominalization in *"health care **organization**"* should be judged *not verbal*, since this mention does not denote a verb-related event, about which it would be natural to ask verbal questions like *What is being organized?*.

3.2 Annotation Methodology

We follow the controlled crowdsourcing methodology presented in Roit et al. (2020) for annotating high-quality QA-SRL. After screening crowd-workers based on their performance in QA-SRL annotation tasks, they undergo a paid training process in which they learn a short annotation manual[5], annotate a few dozens of predicates and receive personal feedback. There are 7 workers in our annotator team.

Our annotation tasks leverage the QA-SRL machinery of Fitzgerald et al. (2018) and Roit et al. (2020). A *generator* worker is presented with a sentence and a highlighted target noun from the candidate nouns previously identified. As mentioned above, she first determines whether the noun carries a verbal meaning (ISVERBAL), and if so, generates questions and answers them using spans from the sentence, following the QA-SRL methodology (QASRL) (see Appendix 8.3 for an interface screenshot). While incorporating a preliminary "predicate detection" decision to the task interface complicates the task, it is also inherently related to the QASRL phase, since the feasibility of forming natural verbal questions about the target noun hints it is a deverbal nominal predicate. It is also noteworthy that in QANom, unlike NomBank specifications, a nominalization without any corresponding arguments in the sentence should also be denoted *verbal*. This design decision may promote downstream usages of our predicate classification information, e.g., capturing nominal event triggers or cross-sentence predicate-argument relations.

In previous QA-SRL work, a single generator protocol was shown to achieve limited coverage of QA-SRL arguments (Fitzgerald et al., 2018), undermining proper evaluation. Therefore, we follow the methodology proposed by Roit et al. (2020) for creating a high quality evaluation set. The QANom evaluation set (dev & test) is compiled by applying a subsequent *consolidation* task, in which a worker reviews the joint generated annotations of $k = 2$ generators and produces the final set (including an ISVERBAL decision and QA annotations). Nonetheless, favoring greater scalability, the train set is annotated by a single generator without consolidation.

[5]Our annotation manual is available at: `https://bit.ly/2Byn2dM`. It mainly targets the ISVERBAL decision, while referring to the QA-SRL guidelines compiled by Roit et al. (2020) for QA related guidelines. Together they consist of about 35 guidelines slides. For comparison, NomBank specifications are 126 pages long.

	Train	Dev	Test	Total
Sentences	7114	1557	1517	10188
Candidate Predicates	23060	4661	4461	32182
Verbal Predicates	9226	2616	2401	14243
QAs	15895	5577	4886	26358
Answers	18900	6925	6064	31889

Table 2: QANom Dataset Statistics, where each sentence contains at least one candidate predicate.

*The **value** of the British **pound** sterling has fallen resulting in an **increase** in **exports** abroad.*			
value	Verbal	What was being valued?	*the British pound sterling*
pound	Not verbal		
increase	Verbal	Where did something increase? What increased? Why did something increase somewhere?	*abroad* *exports* *The value of the British pound sterling has fallen*
exports	Verbal	Where was something being exported? Who exports?	*abroad* *British*

Table 3: QANom crowd-sourced annotations for an example sentence (Wikinews dev).

3.3 Data Collection

For the QANom evaluation sets (dev & test), we annotate the sentences from Wikinews and Wikipedia previously annotated by Roit et al. (2020). For the train set, we sample 8K sentences of these two domains from the train split of Large-Scale QA-SRL (Fitzgerald et al., 2018). Statistics for the final dataset are presented in Table 2; gold-standard annotations for an example sentence can be seen in Table 3.

Annotating a sentence for the dev/test sets yielded 1.6 positive predicates and 3.4 QA pairs, with a cost of 76¢ on average. For the train set, the average cost was 23.5¢ for 2.2 QAs targeting 1.3 positive predicates per sentence. Despite incorporating an additional predicate-detection decision, our cost per consolidated QA (22.3¢) is only %20 higher than that of QA-SRL (18.7¢) (Roit et al., 2020). The whole annotation process, including annotator training and feedback, lasted approximately 7 weeks.

4 Dataset Analysis and Evaluation

4.1 Evaluation Metrics

Evaluating the QANom task consists of evaluating its two subtasks, namely nominal predicate detection (IsVerbal) and QA generation (QASRL).[6] As both the annotation pipeline and the baseline parser leverage our *candidate extraction* procedure to detect candidate nominalizations, we formulate IsVerbal simply as a binary classification task over candidates, measured with accuracy along with recall-precision rates. Only matched positive predicates are inspected for QASRL evaluation.

As for QASRL, we follow and refine the previous QA-SRL metrics in Roit et al. (2020). For each verb, we first align its predicted arguments (i.e. answer spans) to the gold arguments, and then evaluate question equivalence, i.e., whether the predicted and gold questions of aligned answers correspond to the same semantic role.

To measure **Unlabeled Argument detection**, we apply the (**UA**) measure proposed in Roit et al. (2020). Specifically, answer spans are matched at the token level using an intersection over union (IOU) ≥ 0.3 criterion. Since this may induce a many-to-many mapping, we employ maximal bipartite matching between the two sets of answer spans, where each pair of spans passing the above mentioned IOU criterion is considered connected. The resulting maximal matching constitutes the true positive set, while remaining non-aligned arguments become false positives or false negatives.

[6]The described metrics are applied for evaluating both the crowdsourced dataset (§4) and the baseline QANom parser (§5).

Matched answer spans from the previous step are then inspected for role label equivalence to assess **Labeled Argument detection** (**LA**). However, since QA-SRL roles are expressed by natural language questions, evaluating label equivalence is not trivial, as there can be many correct questions for a role (illustration is provided in Appendix 8.1). While previous approaches selected strict question-matching criteria to avoid overestimating agreement, we propose a tighter estimate by mapping question templates into a small set of ROLEs, considering a pair of questions as equivalent if they are mapped into the same ROLE. We embrace the syntactic heuristics proposed by He et al. (2015) to map a QA-SRL question into a corresponding ROLE, taken from a fixed set. ROLEs include subject, object, indirect objects and different kinds of modifier, e.g. *Where*, *When* or *How* (see formal definition in Appendix 8.4).

4.2 Dataset Analysis

4.2.1 Inter-Annotator Agreement

To estimate dataset consistency across different annotations, we measure inter-annotator agreement (IAA) on a sample of 87 sentences (318 candidate nouns), as shown in Table 4. While the worker-vs-worker agreement for QAs is somewhat partial, the overall consistency of the dataset — assessed by comparing different consolidated annotations obtained by disjoint triplets of workers — achieves a reasonable 77% F1 UA. Notably, the consolidation task boosts consistency significantly. We conjecture that this consistency improvement is proportional to the portion of disagreement which is not a matter of controversy, but rather is attributed to the competence of a single worker to identify all related (explicit and implicit) arguments of a target event.

Our agreement measures are lower than reported IAA figures for comparable expert-annotated tasks, such as NomBank (between %82–%90, (Meyers et al., 2004a)). This is within reason, considering annotator training and guidelines scale differences. Further, since QANom annotations do not rely on syntactic analysis or word sense dictionaries (such as NomBank's frame files), more space for semantic disagreement is found.

The consistency figures are lower than those exhibited by using the same crowdsourcing methodology on verbal QA-SRL (Roit et al., 2020), which yielded 79.9 and 84.1 on the Generation and Consolidation tasks respectively. This indicates that nominal arguments are harder to detect, and are identified by a more subjective judgement. We conjecture this relates to the different argument structure of nominalizations, as we further analyze hereinafter (§4.3).

4.2.2 Comparison to NomBank

Inspired by previous verbal QA-SRL works (He et al., 2015; Roit et al., 2020) who have compared themselves to PropBank (Kingsbury and Palmer, 2002), we analyze our agreement with NomBank (Meyers et al., 2004a). Agreement is analyzed with respect to both nominal predicate detection and unlabeled argument detection (as there is no clear mapping between labels of the two). To that end, we ran the QANom annotation pipeline on a sample of 126 sentences from PTB for comparison.

Nominal Predicate Detection As mentioned before, NomBank's definition of nominal predicates is substantially different from that of QANom. While QANom targets only mentions of deverbal nominalizations, NomBank includes non-verbal noun classes. On the other hand, since NomBank annotates only noun instances taking explicit arguments (Meyers, 2007) — where an argument must pertain to certain patterns of syntactic relation to the predicate — many nominalizations annotated in QANom can fall out of NomBank's scope. For this reason, we compare QANom against a subset of verbal NomBank predicates (see Footnote 2). In addition, we review the automatic comparison manually, accounting for cases where an error is attributed to scope discrepancies rather than to QANom annotation errors.

Overall, QANom identified 196 nominalizations while NomBank annotated 224. Comparing the two annotations of our sample, QANom correctly covers 67 predicate mentions not covered by NomBank. Out of these, 48 fall outside NomBank's scope (mentions not having explicit arguments) and 19 are recall misses for NomBank. Conversely, Nombank covers 106 predicate mentions not covered by QANom. Out of these, 63 fall out of QANom's scope (failing the ISVERBAL criteria for deverbal nominalizations), and 43 are recall misses for QANom — mostly borderline cases of nominalizations used generically

	Generation	Consolidated
UA (F1)	67.2	77.1
Role (F1)	72.3	80.5
Is Verbal (Acc.)	81.8	85.6

Table 4: Inter-Annotator Agreement for single worker-vs-worker (Generation) and triplet-vs-triplet (Consolidated).

	Gold		Predicted	
	UA	LA	UA	LA
P	45.1	29.6	47.2	31.6
R	61.5	40.4	49.7	33.3
F1	52.0	34.2	48.4	32.4

Table 5: Performance of QA-SRL parser on QANom given either gold or predicted predicates as targets.

bearing non-eventive meaning. For example, the NomBank predicate *It's **time** for a new season* does not align with the meaning of the verb **to time**, thus should not be questioned about using QA-SRL verbal questions. This analysis reveals the high precision of QANom annotations in detecting verbal nominalizations, discovering only 6 human annotation errors out of 196 positive IsVerbal decisions, and 5 automatic POS tagging errors, which incorporated verb mentions into the dataset. Importantly, this analysis reveals that when considering QANom's intended scope and assuming NomBank is precise, the performance of the QANom annotators reaches 97% precision and 81% recall, relative to QANom's intended scope.

Argument Detection To compare QANom arguments with NomBank arguments, we report a semi-automated analysis of 47 sampled sentences, for which we automatically calculate true positives where both annotations agree, using our UA method (§4.1), and then manually inspect precision and recall mistakes. We catalog our 128 recall misses into the following categories. First, our evaluation methods are too strict, with 21 instances having correct matching arguments which were erroneously labeled as non-matching by the *IOU* metric. 29 instances were assigned to the *Formalism and Syntax* category, meaning that they were either NomBank arguments tackling syntactic patterns and not so much semantic arguments or that our question templates could not properly annotate this relation. Similarly, there are instances in NomBank which are *not eventive nominalizations* according to our definition, out of which there were 63 in this sample. The last category are the true QANom annotator misses, which turned out to be only 12% from a sample of 128 automatically calculated recall errors.

In terms of precision errors, 81 out of 113 were found to not actually be a mistake, as they were either implicit arguments (62), which are not covered by NomBank, or annotation misses in NomBank, or *IOU* metric errors. The other 32 instances were actual errors for QANom, out of which 37% were due to a too far-fetched interpretation of the sentence meaning.

Overall, the manual inspection results in a precision of 86.03 and a recall of 92.9 in terms of the NomBank coverage of QANom. The comparison to NomBank illustrates the quality of QANom annotations, as well as illuminates some of the differences between the two formalisms, particularly addressing implicit arguments in QANom.

4.3 Additional Comparisons

To gain a richer perspective on the QANom data and how it compares to other QA-based annotations, we quantified the percentage of *implicit* arguments in the development sets of QANom, QA-SRL (Roit et al., 2020) and QAMR (Michael et al., 2018). For this automatic analysis, an argument is considered *implicit* if, on a dependency tree, none of its words is connected to the predicate in a path length ≤ 2.[7] Inline with the syntactic properties of nominalizations — which unlike verbs, do not select mandatory arguments (Alexiadou, 2010) — we find that QANom arguments are expressed indirectly more frequently (**43%**) than QA-SRL arguments (**29%**), though significantly less than in QAMR (**64%**).

In order to understand how well arguments of nominalizations are captured by QAMR annotations, we also manually analyzed its coverage compared to QANom on a shared sample of 40 predicates. We find that while QANom yields an average of 2.1 QAs tackling semantic arguments, many QAMR free-formed questions are essentially asking about different predicates in the sentence, resulting in a lower average

[7] We use predicted dependency trees by SpaCy, and consider paths of both directions.

of 1.4 argument-tackling QAs.[8] We conclude that following the QA-SRL approach for restricting the question format is more promising for achieving a comprehensive structured semantic representation.

5 Baseline Parser for QANom

In this section, we present an initial QANom parser, to serve as baseline for future work on this task. We apply a simple pipeline corresponding QANom annotation sub-tasks, namely ISVERBAL and QASRL.

Given a sentence, a **predicate detector** classifies nominalization candidates (extracted using our lexical-based procedure) as *verbal* vs. *non-verbal*. We developed a vanilla BERT-based model implemented by fine-tuning `bert-base-cased` pre-trained model (Devlin et al., 2018) on QANom's ISVERBAL task. Contextualized representations of candidates are classified by a binary classification layer.

Positive (i.e. *verbal*) nominalization predicates are then passed on to a **QA-SRL parser**, for which we simply adopt and re-train the current state-of-the-art QA-SRL parser (Fitzgerald et al., 2018) for the QANom data. Their model is an argument-first pipeline. A span-based *argument detector* first makes independent binary decisions for all $O(n^2)$ spans in the sentence, selecting arguments passing a threshold τ. Then, a representation of each selected span is passed into the *question generation* model, which uses an LSTM decoder to sequentially predicts the 7 slots which comprise a QA-SRL question. Both models leverage contextualized representations to encode the sentence or the argument span.

Results Our predicate detector reaches an accuracy of **82.4** and F1 of **82.6**, with **88.4** precision and **77.4** recall. This performance is comparable to "human performance" as measured by IAA figures (§4.2.1).

As for QA-SRL annotations prediction, we evaluate our system both by-component (taking gold predicates as input to the QA-SRL parser, as well as gold argument spans for its question generator) and as a full pipeline, using the metrics specified in §4.1. Results (Table 5) indicate that QANom is a harder task than verbal QA-SRL, especially with respect to recall.[9] Manually inspecting some predictions, we conjecture that coverage is especially challenging for implicit arguments (which comprise a considerable percentage of QANom annotations), since these aren't solidly grounded in syntactic structure but rather depend on reasoning and common sense.

6 Extrinsic Evaluation of QANom

There are two main paths for utilizing explicit semantic representations downstream: (1) Utilizing the semantic signal they provide to enhance language understanding within learned models, through additional training; (2) Using them as an explicit intermediate structure over the text, which enables applying more controlled algorithms over this structure. In subsections §6.1 and §6.2, we evaluate the downstream utility of the QANom dataset, with respect to these two paths.

6.1 QUASE Experiments

Recently, QUASE (He et al., 2020) was proposed as a sentence encoder enhancement paradigm, utilizing QA data to incorporate distributed semantic features into the encoder. He et al. (2020) demonstrated that QUASE — taking a pre-trained BERT model (Devlin et al., 2018) and further pre-training it on QA-data — improves performance on various downstream tasks, including semantic dependency parsing (SDP), SRL, relation extraction and textual entailment.

In order to assess the semantic signal of QANom and its extrinsic utility, we compare QUASE which further pre-trained on QANom (QUASE_{QANom}) both to a vanilla BERT and to QUASE further pre-trained on other comparable-size QA-format annotations. We evaluate the models both on a subset of the original downstream tasks (SDP (Oepen et al., 2015) and PropBank (Kingsbury and Palmer, 2002)), as well as new tasks that can benefit from nominal predicate-argument structure information, namely NomBank (Meyers et al., 2004a), SRL annotations in Ontonotes (Pradhan et al., 2013), and event extraction (ACE) (Walker et al., 2006). For each target task, we experiment both with finetuning on the full train set of the task, as well as on a small portion (10%), simulating a low resource setting. For Ontonotes

[8] We used the development sets of both QANom and QAMR. QAMR dev was produced by 3 question-generation workers per target, compared to 2 for QANom dev.

[9] QA-SRL parser's performance on QASRL is reported to be **85.0** F1 for span detection (similar to the **UA** measure).

and ACE, we decompose train/test sets by the lexical category of the predicate or trigger in focus, to investigate the interaction between the semantic space of the pre-training task and of the target task. See Appendix 8.5 for further experimental details.

Our findings are shown in Table 6. Generally, results re-establish the findings of He et al. (2020), that further pre-training on QA annotations improve downstream performance over the BERT encoder, especially in low resource settings. This effect is more profound for semantically-oriented QA annotation (QAMR, QA-SRL and QANom), tackling semantic relations within a sentence, than for simple question-answering data, i.e., SQuAD (Rajpurkar et al., 2018).

As a trend, results also indicate that the performance gain is larger the more relevant the semantic space of the pre-training task is for the target task. Specifically, QAMR provides the best "general-purpose" semantic signal, outperforming all the other models on most heterogeneous tasks (i.e. tasks with mixed types of "targets"). This is inline with its "whole-sentence" semantic nature, capturing relations for all word types. On the other hand, QANom's semantic signal is improving BERT's performance particularly for noun-targeting tasks, performing comparably to QAMR, and the same is true for QA-SRL regarding verb-targeting tasks. Yet, combining different QA data types for pre-training (as in QUASE$_{QANom+QASRL}$ and QUASE$_{QANom+QAMR}$) is generally worse than using only the better performing data type.

Altogether, at this point it seems that for "soft" utilization settings, such as QuASE finetuning, QAMR's rich and diverse semantic signal is advantageous, while how to best combine and exploit the more systematic structures of QA-SRL and QANom (see §4.3) in such settings remains an interesting topic for future research. On the other hand, the utility of QANom's systematic structure is apparent when it is leveraged in an explicit manner, as shown in the next subsection.

6.2 Zero-Shot Event Predicate/Argument Detection

The explicit structure produced by QANom predicate-argument relations may be leveraged in different ways in various downstream tasks. In this section, we investigate the utility of the QANom representation using a zero-shot setting, assessing how well models trained on QANom perform on the ACE event extraction task (Walker et al., 2006) without finetuning on it. ACE is an expertly-annotated event extraction dataset, covering 33 types of events (elicited by a *trigger*, i.e., predicate) along with their participants (i.e., arguments). In our evaluations, we consider only the nominal predicates in ACE (175 instances in the test set), which cover eventive nouns, notably not restricted to deverbal nominalizations (e.g., *storm*, *fire*). We evaluate QANom-trained models on both the Predicate Detection and Argument Detection subtasks, and compare to equivalent NomBank-trained models and to upper-bound ACE-trained models. As in §6.1, we consider only verb-derived NomBank instances for training, in order to keep the scope of the nominal predicates roughly comparable and focused on eventive deverbal nouns. For *Predicate Detection*, we apply our baseline QANom predicate detector (§5), which classifies lexically extracted candidates similarly to our annotation pipeline. When trained on NomBank and ACE, all common nouns are considered candidates, while verbal NomBank predicates and annotated ACE event triggers constitute the positive class, respectively. In the *Argument Detection* experiment, we use the AllenNLP SRL model (Gardner et al., 2018; Shi and Lin, 2019) to train unlabeled argument detection given the gold predicates.

As shown in Table 7, the QANom models perform on par with NomBank models, reaching a slightly better F1 for both tasks, even though QANom is based on cheap crowd annotations. Regarding *Argument Detection*, absolute figures are relatively low for both zero-shot models, reaching roughly half of the ACE-trained model performance. This is due to the discrepancy between ACE slot fillers and SRL arguments, but nevertheless suggests that applying SRL can be an indicative signal for zero shot ACE slot filling, while QANom arguments provide a signal that is more informative than that of NomBank arguments.

With respect to *Predicate Detection*, we note that since ACE is inherently partial — covering only an arbitrary set of 33 event types — automatic precision figures are not very informative. Hence, we further manually analyze 60 predicates taken blindly from false-positives of both models (30 each). We find that

Tasks	SDP		PropBank		NomBank		Ontonotes				ACE	
Split							Nouns		Verbs		Nouns	Verbs
Train Size	10%	100%	10%	100%	10%	100%	10%	100%	10%	100%	100%	100%
BERT	78.16	91.53	35.58	67.06	20.31	56.84	0.62	33.68	47.78	70.73	0.47	2.4
QUASE$_{QANom}$	80.09	91.61	45.20	71.86	30.24	61.66	2.11	46.06	56.52	72.81	9.71	14.2
QUASE$_{QASRL}$	80.42	91.77	51.40	73.24	30.31	60.62	1.52	43.11	60.62	74.28	6.36	17.33
QUASE$_{QAMR}$	80.67	91.68	49.80	72.69	33.91	62.74	2.41	46.06	60.57	74.22	9.56	16.01
QUASE$_{SQuAD}$	79.46	91.53	43.41	70.28	27.80	60.35	0.48	41.1	55.28	72.34	9	14.67
QUASE$_{QANom+QASRL}$	79.63	91.89	47.84	72.92	31.42	61.88	2.3	46.24	59.7	74.28	9.82	17.96
QUASE$_{QANom+QAMR}$	80.24	91.89	48.02	72.22	32.54	62.14	2.58	45.26	59.16	73.93	8.13	14.87

Table 6: Comparison between BERT and QUASE models further pre-train on various types of same-sized QA data. QUASE$_{QANom+QASRL}$ and QUASE$_{QANom+QAMR}$ are trained on half-by-half combined training sets. All results are reported according to the span based micro F1 measure.

	Predicate Detection			Argument Detection		
Train Set	P	R	F1	P	R	F1
NomBank	21.3	77.1	33.4	20.0	25.3	22.3
QANom	24.9	74.9	37.4	24.8	29.7	27.0
ACE	73.2	82.9	77.7	51.3	65.6	57.5

Table 7: Zero-shot Predicate and Argument Detection evaluated on ACE nominal triggers. We leverage the zero-shot setting to compare the QANom vs. NomBank datasets with respect to event extraction.

19 of the predicates predicted by the QANom model (vs. 11 for the NomBank model) are valid eventive nominalization of event types not covered by ACE, estimating the effective F1 performance at the lower 80s. Observing the misses of our predicate detector, most of them are eventive nouns not corresponding to an equivalent verb (e.g. *war* or *conflict*), which fall out of QANom's scope. All in all, QANom seems to capture important information for event extraction, especially for nominal event trigger identification, and can thus be utilized as an additional signal or feature in models that target this task.

7 Conclusion

We present QANom, a light-weight QA-based scheme for annotating semantic roles for deverbal nominalizations. QANom shares its role space with the promising QA-SRL line of work, is attainable through a simple crowdsourcing task, and is shown to contain valuable information, both compared to expert-annotated nominal SRL resources and when extrinsically evaluated on nominalization-related tasks. While we present baseline figures for QANom parsing, future work should explore how to improve model performance, considering e.g. joint learning or transfer techniques (Zhao and Titov, 2020; Sánchez and Oliveira, 2017).

Acknowledgments

This work was supported in part by grants from Intel Labs, Facebook, the Israel Science Foundation grant 1951/17 and the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

References

Artemis Alexiadou. 2010. Nominalizations: A probe into the architecture of grammar part i: The nominalization puzzle. *Language and Linguistics Compass*, 4(7):496–511.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98/COLING '98, page 86–90, USA. Association for Computational Linguistics.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189.

Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky. 2013. Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 120–125. IEEE.

Pengxiang Cheng and Katrin Erk. 2018. Implicit argument prediction with event knowledge. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 831–840.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Quynh Ngoc Thi Do, Steven Bethard, and Marie-Francine Moens. 2017. Improving implicit semantic role labeling by predicting semantic frame arguments. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 90–99.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *ACL*, pages 484–490.

Nicholas Fitzgerald, Julian Michael, Luheng He, and Luke S. Zettlemoyer. 2018. Large-scale qa-srl parsing. In *ACL*.

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

Matthew Gerber and Joyce Y Chai. 2012. Semantic role labeling of implicit arguments for nominal predicates. *Computational Linguistics*, 38(4):755–798.

Nizar Habash and Bonnie Dorr. 2003. A categorial variation database for english. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 17–23. Association for Computational Linguistics.

Jan Hajic, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, M 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 (CoNLL 2009): Shared Task*, pages 1–18.

Luheng He, Mike Lewis, and Luke S. Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *EMNLP*.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia, July. Association for Computational Linguistics.

Hangfeng He, Qiang Ning, and Dan Roth. 2020. Quase: Question-answer driven sentence encoding. In *Proeedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Zheng Ping Jiang and Hwee Tou Ng. 2006. Semantic role labeling of NomBank: A maximum entropy approach. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 138–145, Sydney, Australia, July. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*, pages 1989–1993. Citeseer.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. Nomlex: A lexicon of nominalizations. In *Proceedings of EURALEX*, volume 98, pages 187–193.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004a. Annotating noun argument structure for nombank. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004b. The nombank project: An interim report. In *Proceedings of the workshop frontiers in corpus annotation at hlt-naacl 2004*, pages 24–31.

Adam Meyers. 2007. Annotation guidelines for nombank-noun argument structure for propbank. *Online Publication: http://nlp. cs. nyu. edu/meyers/nombank/nombank-specs-2007. pdf*, 44.

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2018. Crowdsourcing question-answer meaning representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 560–568.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (sr'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajic, and Zdenka Uresova. 2015. Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926.

Sebastian Padó, Marco Pennacchiotti, and Caroline Sporleder. 2008. Semantic role assignment for event nominalisations by leveraging verbal data. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 665–672. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria, August. Association for Computational Linguistics.

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

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

Liliana Mamani Sánchez and Claudia Oliveira. 2017. Deverbal noun complementation rules applied to semantic role labeling. *Revista da ABRALIN*, 7(2).

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305.

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, Volume 1 (Long Papers)*, pages 885–895.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

	WH	AUX	SUBJ	VERB	OBJ1	PREP	OBJ2\MISC	?
1	Who	has		proven	something			?
2	How	did	someone	prove	something			?
3	What	has	someone	proven				?
4	What	has been		proven		by	someone	?
5	Who	did	someone	prove	something	about		?
6	What	did	someone	prove		about	someone	?

Table 8: Examples for QA-SRL questions decomposed into their slot-based template. Refer to (He et al., 2015) for the full description. Questions 3 and 4 capture the same role, while 5–6 jointly represent an alternative annotation which decomposes their answer span (*God existence*) into two answer (*God*, *Existence*).

Diana Trandabăţ. 2011. Using semantic roles to improve summaries. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 164–169. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 700–706.

Yanpeng Zhao and Ivan Titov. 2020. Unsupervised transfer of semantic role models from verbal to nominal domain. *arXiv preprint arXiv:2005.00278*.

8 Appendices

8.1 QA-SRL Question Template and Answers

Each non-copular verb in QASRL is annotated with a list of QAs, where each one corresponds to a role, and contains one or more answer spans that are considered the role's arguments. Answer spans for the same predicate cannot overlap. The template defining the structure of QA-SRL questions is shown in Table 8. It consists of 7 slots, where some slots contribute the core meaning of the role (**WH**, **PREP**, and argument placeholders) while others merely make the question grammatical and sound in context (**AUX** and **VERB**-inflection). Since we adopt the auto-complete mechanism of Fitzgerald et al. (2018) for QANom annotations, only grammatical combinations of slot-fillings are included in the label space.

The table also illustrates possible differences between valid QASRL annotations. For example, questions 3 and 4 both tackle the theme (*God existence* in the running example from Table 1), though differing in voice and aspect.

8.2 Lexical Heuristics for Candidate Nominalization Extraction

In preliminary experiments, available lexical resources for English have performed well in filtering common nouns that have at least one morphologically related verb. This section provides further details about the lexical resources and heuristics we used in pre-processing in order to extract nominalization candidates. We rely on three sources for retrieving derivationally related verbs. In cases when the union of related verbs attained from our lexical resources contains more than one verb, we select the verb which minimizes the Levenshtein distance to the noun.

CatVar The Categorial Variation Database for English (CatVar)[10] is a lexical bank that clusters categorical variants (i.e. part-of-speech) of the same lexeme. For example, the *develop* cluster contains the words: develop (V), developer (N), developed (AJ), developing (N), developing (AJ), development (N)). We map every noun occurring in CatVar to the verbs in its cluster (if any). This mapping covers $13,788$ lexical items resolved as candidate nominalization.

[10] https://clipdemos.umiacs.umd.edu/catvar/

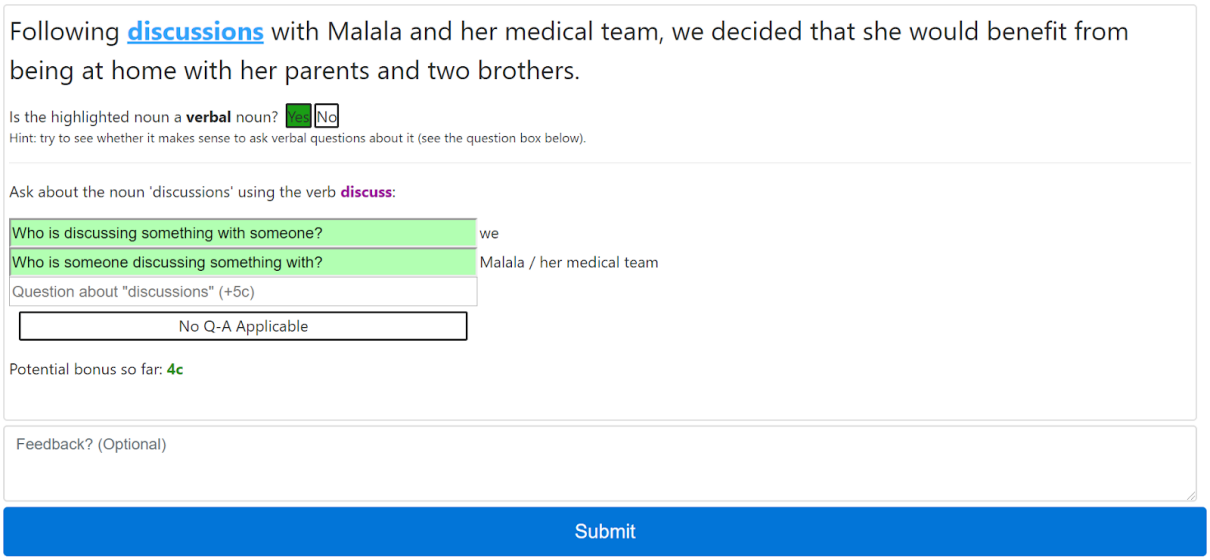

Figure 1: User Interface of the QANom Annotation Tool. This is embedded into the worker's task page in Amazon Mechanical Turk.

WordNet To enhance coverage, and to experiment with a multi-lingual resource, we also utilize WordNet for identifying potential nominalizations. Every noun is mapped to all the *derivationally related forms* available through any of its *lemmas*. A noun having at least one verbal related form is taken as a candidate.

Suffix-based heuristic Finally, since most common nominalization patterns in English can be described by simply fusing certain suffixes to the verb, we extend our coverage with a simple lookup method. Given a seed of verbs, we apply manually crafted suffix-substitution rules in order to derive a large list of possible nominalizations. For example, for the seed verb **develop**, our list includes the synthetically generated words (or pseudo-words) *developing*, *development*, **develoption*, **developance*, **developal*, etc. A common noun is looked up in this large list, and assigned the corresponding seed verb as the verbal form.

8.3 Annotation Task Interface

A screenshot of our annotation tool for QANom is shown in Figure 1. It inherits the properties of the Large Scale QA-SRL annotation interface, including an auto-complete and an auto-suggest mechanisms. The automaton-based auto-complete mechanism facilitates question generation while enforcing the questions to be grammatical and to adhere the QA-SRL question-format. The auto-suggest feature streamlines annotation by proposing question pertaining ROLEs which are not yet asked about, realized in the same tense, aspect and voice and modality as previous questions.

On top of the QA-SRL UI properties, QANom has several new features. First, we add a preliminary ISVERBAL question, where a *No* answer automatically hides the question generation component. The questions are be generated using the pre-computed related verb, which is highlighted for the user (*discuss* in this example). In addition, we allow for a positive predicate (i.e. a noun for which ISVERBAL is *Yes*) to have *No Q-A Applicable* (unlike verbal QA-SRL), because for some nominalizations no argument is mentioned in the sentence.

8.4 Question ROLE Set

As mentioned in Section 4.1, we embraced the syntactic heuristics proposed by He et al. (2015) to map a question into a corresponding ROLE from the set $\mathcal{R}$ based on its **WH**, **SBJ**, **OBJ1**, **OBJ2** and **PP** slots.

$\mathcal{R}$ is defined as follows:

$$\mathcal{R} = \{\text{A0}, \text{A1}, \text{A2}, \text{A2}[p], adjunct, adjunct[p]\}$$
$$adj \in \{\textbf{Where}, \textbf{When}, \textbf{Why}, \textbf{How}, \textbf{How long}, \textbf{How much}\}$$
$$p \in Prepositions$$

In short, roles A0–A2 correspond to subject, direct object and indirect object whereas $adjunct$ roles correspond to adjuncts, while taking the preposition into account. See Table 7 in He et al. (2015) for the full algorithm of mapping a question into a ROLE.

8.5 Details on QUASE Experiments

This section provides further details about the QUASE framework (He et al., 2020) we leverage in Section 6.1 in order to evaluate the external utility of the QANom dataset, and specifies about our experimental setting.

QUASE is proposed as a method to retrieve distributed meaning representations from QA pairs. Specifically, two types of sentence encodings, s-QUASE and p-QUASE, are proposed to extract semantic features for single-sentence and paired-sentence tasks. Both use the BERT sentence encoder (Devlin et al., 2018), but s-QUASE wraps it in a special architecture allowing it to leverage further pre-training on semantically rich QA tasks for improving on single-sentence downstream tasks.

In essence, we replicate the experiments specified by He et al. (2020) (Table 3), with new QA-datasets for further-pre-training along with new downstream tasks. We use only s-QUASE as all our downstream tasks are single-sentence tasks. Treating s-QUASE as a specialized semantic feature extractor, the original word embeddings are concatenated/replaced with the s-QUASE representation in the input layer of specific models in downstream tasks.

In our experiments, we compare s-QUASE further pre-trained on various QA-dataset (including QANom) with BERT on three downstream tasks: SRL, event extraction, and SDP. Specifically, for further pretraining we use SQuAD (Rajpurkar et al., 2018), QA-SRL Large Scale (Fitzgerald et al., 2018), QAMR (Michael et al., 2018), QANom, and combinations between the latter three. All training sets consist roughly 16K QA pairs as QANom's train set. As for downstream tasks, we use Nombank (Meyers et al., 2004b) for nominal SRL, Propbank (Kingsbury and Palmer, 2002) for verbal SRL, Ontonotes 5.0 (Pradhan et al., 2013) for both nominal and verbal SRL, ACE 2005 (Walker et al., 2006)[11] for event extraction, and SemEval'15 shared task with DELPH-IN MRS-Derived Semantic Dependencies target representation (Oepen et al., 2015) for SDP. Similar to He et al. (2020), we use simple BiLSTM model for nominal SRL, verbal SRL and event extraction, and the biaffine network in Dozat and Manning (2018) for SDP (part-of-speech tags are removed from its input). In addition, we replace the original word embeddings in these models (e.g., GloVe (Pennington et al., 2014)) by BERT, and the results are reported on the development sets. For SDP, we concatenate word embeddings with s-QUASE features, while we replace word embeddings with s-QUASE features for nominal SRL, verbal SRL and event extraction.

[11] We use the same train/dev/test sets as Li et al. (2013) for ACE 2005.