QASem Parsing: Text-to-text Modeling of QA-based Semantics

Ayal Klein **Eran Hirsch** **Ron Eliav**
Valentina Pyatkin **Avi Caciularu** **Ido Dagan**
Computer Science Department, Bar-Ilan University
{ayal.s.klein,hirsch.eran,roneliav1,valpyatkin,avi.c33}@gmail.com
dagan@cs.biu.ac.il

Abstract

Various works suggest the appeal of incorporating explicit semantic representations when addressing challenging realistic NLP scenarios. Common approaches offer either comprehensive linguistically-based formalisms, like AMR, or alternatively Open-IE, which provides a shallow and partial representation. More recently, an appealing trend introduces semi-structured natural-language structures as an intermediate meaning-capturing representation, often in the form of questions and answers.

In this work, we further promote this line of research by considering three prior QA-based semantic representations. These cover verbal, nominalized and discourse-based predications, regarded here as jointly providing a comprehensive representation of textual information — termed *QASem*. To facilitate this perspective, we investigate how to best utilize pre-trained sequence-to-sequence language models, which seem particularly promising for generating representations that consist of natural language expressions (questions and answers). In particular, we examine and analyze input and output linearization strategies, as well as data augmentation and multitask learning for a scarce training data setup. Consequently, we release the first unified QASem parsing tool, easily applicable for downstream tasks that can benefit from an explicit semi-structured account of information units in text.

1 Introduction

A traditional line of research in NLP has been devoted to designing various kinds of semantic representations, that aim to explicate textual meaning with a formal, consistent annotation schema. Representations such as Semantic Role Labeling (SRL; e.g. Baker et al., 1998), Discourse Representation Theory (Kamp et al., 2011) and others (Copestake et al., 2005; Banarescu et al., 2013; Abend and Rappoport, 2013; Oepen et al., 2015; White et al., 2016; Bos et al., 2017) provide applications with an explicit account of semantic relations in a text. Numerous recent works illustrate how leveraging explicit representations facilitate downstream processing of challenging tasks (Lee and Goldwasser, 2019; Huang and Kurohashi, 2021; Mohamed and Oussalah, 2019; Zhu et al., 2021; Chen and Durrett, 2021; Fan et al., 2019). While traditional representations rely on pre-defined schemata or lexica of linguistic classes (e.g. semantic roles), the popular approach of Open Information Extraction (OpenIE; Etzioni et al., 2008) aims for more loosely-structured, easily attainable representations, comprised of tuples of natural language fragments. These light-weight structures, however, come with a cost of lacking consistency and comprehensive coverage, and do not capture deeper semantic information like semantic roles.

In a recent trend, which can be seen as an emerging mid-point between full-fledged semantic formalisms and bare-bone textual fragments, researches leverage question-answer pairs (QAs) as a representation of textual information (Michael et al., 2018). For example, several works proposed using QAs as an intermediate structure for assessing information alignment between texts, e.g. for evaluating summarization quality (Eyal et al., 2019; Gavenavicius, 2020; Deutsch et al., 2021) and faithfulness (Honovich et al., 2021; Durmus et al., 2020), using a question-generation plus question-answering (QG-QA) approach. Nevertheless, such question generation and answering models were not trained to provide a coherent representation of text meaning.

In this work, we follow an evolving paradigm, consisting of tasks that aim to comprehensively capture certain types of predications using question-answer pairs. The pioneering work in this framework is Question Answer driven Semantic Role Labeling (QA-SRL; He et al., 2015). Targeting verbal predicates, QA-SRL labels each predicate-argument relation with a question-answer pair,

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7742 - 7756

Both were **shot** in the **confrontation** with police and have been **recovering** in hospital since the **attack** .			
QA-SRL	1	When was someone **shot?**	in the confrontation ; the attack
	2	Who was **shot?**	Both
	3	Who **shot** someone?	police
	4	Where has someone been **recovering?**	in hospital
	5	How long was someone **recovering** from something?	since the attack
	6	Who was **recovering** from something?	Both
	7	What was someone **recovering** from?	shot
QANom	8	Who **confronted** with something?	Both
	9	What did someone **confront** with?	police
QADiscourse	10	*Since when* have both been **recovering** in hospital?	since the **attack**
	11	*While what* were both **shot**?	During the **confrontation** with police

Table 1: An example sentence annotated with QASem (V1) — QA-SRL, QANom and QADiscourse. Target predicates (verbs and nominalizations) are shown in **bold**, while QADiscourse prefixes are shown in *italics*. Multiple answers are delimited by a semicolon (;).

where a natural language question represents a semantic role, while answers correspond to arguments (See Table 1). Notably, QA-SRL was shown to subsume OpenIE, which can be derived from QA-SRL annotations by reducing them to unlabeled predicate-argument tuples (Stanovsky and Dagan, 2016). This appealing QA-based framework, well suited for scalable crowdsourcing (Fitzgerald et al., 2018), has been extended to account for deverbal nominalizations (QANom; Klein et al., 2020) and for information-bearing discourse relations (QADiscourse; Pyatkin et al., 2020). We deem these individually-presented tasks as milestones toward a broad-coverage QA-based semantic representation, which we denote as *QASem*. To make this goal accessible, we develop a comprehensive modeling framework and release the first unified tool for parsing a sentence into a systematic set of QAs, as in Table 1. This set covers the core information units in a sentence, based on the above three predication types (verbs, nominalizations and discourse relations).[1]

Current best models for QA-SRL/QANom and QADiscourse (Fitzgerald et al., 2018; Pyatkin et al., 2020) are classifier-based pipelines, each targeting a specific QA format. Predictors of relation labels (questions) use a specialized architecture that suits the task-specific question structure, and are modeled independently from relation participants (answers). Our work leverages recent progress in text-to-text pre-trained neural models, and specifically T5 (Raffel et al., 2020), for predicting QA-based annotations in a generic manner. Our semi-structured QASem use-case is an interesting mid-ground between natural language generation and structured prediction tasks. A QASem output sequence includes *a set* of *restricted natural language* fragments (the QAs), possibly harnessing the seq2seq language generation pre-training objective rather than merely model's language understanding.

We find that fine-tuning T5 on the QA-based semantic tasks is favorable over prior approaches, producing state-of-the-art models for all the aforementioned tasks. Our experiments suggest that T5 is good at learning the grammar characterizing our semi-structured outputs, and that input and output linearization strategies have a significant effect on performance. We further explore the benefits of joint multi-task training of nominal and verbal QA-SRL. Our tool, including models and code, is publicly available.[2]

2 Background

2.1 QA-based Semantic Representation

The traditional goal of semantic representations is to reflect the meaning of texts in a formal, explicit manner (Abend and Rappoport, 2017). SRL schemes (Baker et al., 1998; Kingsbury and Palmer, 2002; Schuler, 2005), for example, decompose a textual clause into labeled predicate-argument relations specifying "who did what to whom", while discourse-level representations (Mann and Thompson, 1987; Kamp et al., 2011; Prasad et al., 2008) capture inter-clause relations. Such semantic representations can be leveraged by NLP applications that require an explicit handle of textual content units for their algorithms — for example, content selection for text generation tasks (Mohamed and

[1]This paper presents QASEM V1. Future versions will include QA-based tasks that capture complementary information specified by adjectival predicates and other noun modifier, which are currently at a stage of ongoing work.

[2]We publish a unified package for jointly producing all QASem layers of annotation with an easy-to-use API — https://github.com/kleinay/QASem. The repository also includes model training and experiments code.

Oussalah, 2019; Liu et al., 2015; Hardy and Vlachos, 2018) or information consolidation in multi-document settings (Liao et al., 2018; Pasunuru et al., 2021; Chen and Durrett, 2021).

A main drawback of these carefully-designed formalisms is their annotation cost — since they rely on schemata of linguistically-oriented categories (e.g. semantic roles), dataset construction requires extensive annotator training, restricting their applicability to new text domains and new languages.

In recent years, several works proposed to remedy this annotation bottleneck by taking a more "open-ended" approach, capturing semantics using natural language self-explanatory terms (Butnariu et al., 2009; Shi and Demberg, 2019; Yung et al., 2019; Elazar et al., 2021). In a related trend, many recent works utilize question-answer pairs from generic QA models for soliciting a manageable, discrete account of information in a text. These can be used as content units for planning text generation (Narayan et al., 2022), or for guiding textual information alignment (Eyal et al., 2019; Gavenavicius, 2020; Deutsch et al., 2021; Honovich et al., 2021; Durmus et al., 2020). In Section 7 we discuss the limitations of such "ad-hoc" representations in comparison to the QA-based semantic framework which we set forth here.

This paper pursues *QASem*, a systematic framework for QA-based semantic representation, based on an evolving line of research that introduced so far three concrete complementary representations — namely, QA-SRL, QANom and QADiscourse. QASem can be seen as an overarching endeavor of developing a comprehensive layered representation scheme, covering all important types of information conveyed by a text. We now turn to present the three current building blocks of QASem.

2.2 QASem Tasks

QA-SRL With the goal of collecting laymen-intuitive semantic annotations, QA-SRL (He et al., 2015) annotates verbs with a set of natural language QAs, where each QA corresponds to a single predicate-argument relation. QA-SRL questions adhere to a 7-slots template, with slots corresponding to a WH-word, the verb, auxiliaries, argument placeholders (SUBJ, OBJ1, OBJ2), and a preposition. The QA-SRL templates were designed to comprehensively and systematically capture all kinds of arguments and modifiers, as illustrated in Table 1 A question is aligned with one or more answers (when a role has multiple 'fillers'), each is a continuous span from the sentence.

Beyond data collection scalability (Fitzgerald et al., 2018), QA-SRL yields a richer argument set than linguistically-rooted formalisms like PropBank (Kingsbury and Palmer, 2002), including valuable implicit arguments (Roit et al., 2020). It was also shown to subsume the popular OpenIE representation (Stanovsky and Dagan, 2016) and to enhance pre-trained encoders (He et al., 2020).

QANom In a follow-up work, Klein et al. (2020) extended the QA-SRL framework to also cover deverbal nominal predicates, which are prevalent in texts. First, candidate nominalizations — nouns that have a derivationally related verb — are extracted using lexical resources (Miller, 1995; Habash and Dorr, 2003). QANom annotators then classify whether the candidate carries a verbal, eventive meaning in context ("The **construction** of the offices...") or not ("...near the huge **construction**"). Then, predicative nominalizations undergo QA-SRL annotation, generating QAs in exactly the same format as verbal QA-SRL. The result is a unified framework for verbs and nominalizations (See Table 1), analogous to the relationship between the PropBank (Kingsbury and Palmer, 2002) and NomBank (Meyers et al., 2004) projects.

QADiscourse The relationship between propositions in a text can by itself deliver factual information. Several formalisms, such as Rhetorical Structure Theory (RST; Mann and Thompson, 1987) or the Penn Discourse TreeBank (PDTB; Miltsakaki et al., 2004), have labeled inter and intra-sentential discourse relations using a taxonomy of pre-defined relation senses, e.g. CONTINGENCY.CONDITION or TEMPORAL.ASYNCHRONOUS.SUCCESSION. Following the QA-SRL paradigm, Pyatkin et al. (2020) proposed to annotate discourse relations using natural language question-answer pairs (See Table 1). They devised a list of question prefixes (e.g. *In what case X?* or *After what X?*) corresponding to a subset of PDTB relation types capturing all 'informational' relations, excluding senses specifying structural or pragmatic properties of the realized passage. Annotators were presented with a sentence and certain heuristically extracted event targets marked in that sentence. They were then asked to relate such event targets with a question starting with one of the prefixes, if applicable. The

question body (after the prefix) was a copied sentence span containing one of the targets whereas the answer span contained the other. Different from QA-SRL and QANom, both copied spans could be slightly edited to sound grammatical and fluent.

2.3 Relationship to Other Representations

Schema-based Semantic Formalisms It is noteworthy that while QASem achieves a systematic coverage of semantic relations through carefully designed question templates, these QA-based annotations do not map directly into a formal semantic ontologies like traditional semantic representations. Rather, the QASem philosophy is to capture how non-professional proficient speakers perceive the semantic relations in the text and express them in a natural question-answer form. While QASem is generally proposed as an appealing alternative to traditional (schema-based) representations, the two approaches may also be seen as complementary. QASem can be used in many downstream tasks that require an explicit account of semantic relation structure, which may well be represented in an "open" natural language based form (similar to OpenIE), while including an informative signal about relation types (which OpenIE lacks). In other scenarios, where well-defined or fine-grained semantic distinctions are crucial, schema-based semantic formalisms like traditional SRL might be more suitable.

QAMR Following a similar philosophy, Michael et al. (2018) introduced Question-Answer driven Meaning Representation (QAMR), a crowdsourcing scheme for annotating sentence semantics using QAs. Unlike the templated questions in QASem, QAMR consists of free-formed questions incorporating at least one content word from the sentence, along with corresponding answer spans. This results in a highly rich yet less controlled representation. Consequently, as shown by Klein et al. (2020, §4.3), the QAMR annotation approach yields much less comprehensive coverage of semantic relations compared to the template-based approach of QASem.

2.4 Prior QASem Models

As mentioned above, previous models for QA-SRL/QANom and QADiscourse were designed to match the specific question format of each of the tasks. We hereby provide further details about these models.

Leveraging its intuitive nature, Fitzgerald et al. (2018) crowdsourced a large-scale QA-SRL dataset. The dataset was then used for training an argument-first pipeline model for parsing the concrete QA-SRL format, comprised of a span-level binary classifier for argument detection, followed by a question generator. The latter is an LSTM decoder which, given a contextualized representation of the selected span, sequentially predicts fillers for the 7 slots which comprise a QA-SRL question.

Since corresponding verbs and nominalizations share the same semantic frame, but differ in their syntactic argument structure, modeling both types of predicates jointly is a non-trivial yet promising approach (Zhao and Titov, 2020). Nevertheless, Klein et al. (2020) have only released a baseline parser, retraining the model of Fitzgerald et al. (2018) on QANom data alone. Their model achieves mediocre performance, presumably due to the limited amount of QANom training data, which is by an order of magnitude smaller than the training data available for verbal QA-SRL.

Pyatkin et al. (2020) modeled the QADiscourse task with a three-step pipeline. Utilizing the discrete set of question prefixes, they employ a prefix classifier, followed by a pointer generator model (Jia and Liang, 2016) to complete question generation. Finally, they fine-tune a machine reading comprehension model for selecting an answer span from the sentence.

Differing from previous pipeline approaches, we model each of the QASem tasks using a one-pass encoder-decoder architecture. In addition, we regard the three tasks as sub-tasks of a single unified framework, proposing a single architecture for parsing QA-based semantic annotations, also applicable for future extensions of the QASem framework.

3 Modeling

We release a *QASem tool* for parsing sentences with any subset of the QA-based semantic tasks. Our tool first executes sentence-level pre-processing for QA-SRL/QANom. It runs a part-of-speech tagger to identify verbs and nouns,[3] then applies candidate nominalization extraction heuristics (See §2) followed by a binary classifier for detecting predicative nominalizations (Klein et al., 2020). Identified predicates are then passed into the QA-SRL or QANom text-to-text parsing models, while

[3] we use SpaCy 3.0 — https://spacy.io/

Task	QA-SRL			QANom			QADiscourse		
Dataset	2018	2020		(Klein et al., 2020)			(Pyatkin et al., 2020)		
Split	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
Sentences	44476	1000	999	7114	1557	1517	7994	1834	1779
Predicates	95253	1000	999	9226	2616	2401	-	-	-
Questions	215427	2895	2852	15895	5577	4886	10985	2632	2996
Answers	348349	3546	3549	18900	6925	6064	10985	2632	2996

Table 2: QASem Datasets Statistics. QA-SRL Training set comes from Fitzgerald et al. (2018), while evaluation sets are from Roit et al. (2020).

the QADiscourse model takes a raw sentence as input with no pre-processing required. The models are described in detail in the following subsections.

3.1 Baseline Models

We first finetune pre-trained text-to-text language models on each of the QASem tasks separately (BASELINE). Unless otherwise mentioned, most modeling details specified hereafter apply also for the joint models (§3.2). We experiment both with BART (Lewis et al., 2020) and with T5 (Raffel et al., 2020), but report results only for the T5 model for clarity, as we consistently observed its performance to be significantly better. We use `T5-small` due to computational cost constraints.

Our text-to-text modeling for QA-SRL and QANom is at the *predicate-level* — given a single predicate in context, the task is to produce the full set of question-answer pairs targeting this predicate. Our input sequence consists of four components — task prefix, sentence, special markers for the target predicate, and verb-form — as in this nominalization example:

> *parse: Both were shot in the [PREDICATE] confrontation [PREDICATE] with police ... [SEP] confront*

The prefix (*"parse:"*) is added in order to match the T5 setup for multitask learning. Then, the sentence is encoded together with bilateral marker tokens signaling the location of the target predicate (we report alternative methods to signal predicates in Appendix A.2). At last, the verbal form of the predicate (*"confront"*) is appended to the input sequence. This is significant for QANom, since the output verb-centered QA-SRL questions involve the verbal form of the nominal predicate. Verbal forms are identified during the candidate nominalization extraction phase in pre-processing, and are thus available both at train and at test time.[4]

Since the intended output is a *set* of QAs, one can impose any arbitrary order over them. We examine different output linearization strategies, and present our findings in Section 5.1, while the main results section (§5.2) report the best model per dataset. Finally, the ordered QA list is joined into a structured sequence using three types of special tokens as delimiters — `QA|QA` separator, `Question|Answers` separator, and `Answer|Answer` separator for questions with multiple answers.

For the QADiscourse task we train a *sentence-level* model. The input is the raw sentence, while the output is the set of QA pairs pertaining to all targets occurring in the sentence. Inline with our approach in QA-SRL parsing, we prepend inputs with a new task prefix, and use special tokens as delimiters (`QA|QA` and `Question|Answer`).

3.2 Joint QASem Learning

Leveraging the shared output format of QA-SRL and QANom, we further train a unified model on both datasets combined (JOINT). Taking into account the imbalance in training set size for the two tasks, we duplicate QANom data samples by a factor of 14, approximating a 1:1 ratio between QAs of verbal and nominal predicates (See Table 2).

It is worth mentioning that we have tested several methods for incorporating explicit signal regarding the source task (i.e. predicate type — verbal or nominal) of each training instance, aiming to facilitate transfer learning. Our experiments include: prefix variation (e.g. *"parse verbal/nominal:"*); typed predicate marker, i.e., having a different marker token for verbal vs. nominal predicates; and appending the predicate type to the **output** sequence, simulating a predicate-type classification objective in an auxiliary multitask learning framework (e.g. Bjerva, 2017; Schröder and Biemann, 2020). Nonetheless, throughout all our experiments, unin-

[4]For verbal QA-SRL, appending the verb-form (which is the predicate itself) did not improve performance. However, in the joint verbal and nominal model, all instances are appended with a verb-form for consistency.

formed joint learning of verbal and nominal predicates works significantly better.

4 Experimental Setup

Datasets We use the QADiscourse and QANom original datasets (Pyatkin et al., 2020; Klein et al., 2020). For QA-SRL, we make use of the large scale training set collected by Fitzgerald et al. (2018). However, prior work (Roit et al., 2020) pointed out that their annotation protocol suffered from limited recall along with multiple, partially overlapping reference answers, hindering parser evaluation. For these reasons, Roit et al. (2020) applied a controlled crowdsourcing procedure and produced a high-quality evaluation set, dedicated for fair comparison of future QA-SRL parsers. We adopt their annotations for validation and test.[5] Datasets statistics are presented in Table 2.

Evaluation Metrics For QA-SRL and QANom evaluation, we adopt the measures put forward by Klein et al. (2020). The unlabeled argument detection metric (**UA**) measures how many of the predicted answers are aligned with ground truth answers, based on token overlap. Aligned QAs are then inspected for question equivalence to assess semantic label assignment, comprising the labeled argument detection metric (**LA**). Consequently, **LA** figures are bounded by **UA**, as they require to match both the answer and the question to a gold QA to count as a true positive QA. Analogously, we embrace the **UQA** and **LQA** metrics proposed by Pyatkin et al. (2020) for QADiscourse evaluation. See Appendix A.1 for a more detailed description of the evaluation measures.

Output Set Linearization Experiment As stated, the output of the model is parsed into a set of question-answer pairs at post-processing. Thus, the ordering one applies over the linearization of QAs into an output sequence can be arbitrary. It is therefore appealing to examine which ordering schemes facilitate model learning more than others.[6] We compare a randomized order (**Random-Order**) with two consistent ordering methods. The **Answer-Order** method orders the QAs according to answer position in the source sentence, teaching the model to "scan" the sentence sequentially in the search for arguments of the predicate. Alternatively, QAs can be ordered more conceptually, with respect to the semantic role they target. The **Role-Order** method sorts QAs by their WH-word which is a proxy of semantic role.[7]

In contrast to methods that confine the model to a fixed order, one could aim to teach the model to ignore QA ordering altogether. One way to achieve order invariance is to train over various permutations of the QA set rather than a fixed order per instance (Ribeiro et al., 2021). In addition to order-invariance, training on multiple permutations may enhance performance from a data-augmentation perspective, especially in a realistic medium-size dataset setting.

Thus, we experiment with three permutation-based augmentation methods. The most straightforward approach is to include all QA permutations of each predicate (**All-Permutations**).[8] Nevertheless, in order to cope with the exponential data imbalance toward predicates with more QA pairs, an alternative method samples a fixed number of k permutations for all predicates (**Fixed-Permutations**; we set $k = 3$). On the other hand, there are reasons to assume that predicates with more QAs would be generally harder for the model to learn (see Appendix A.5). The third method therefore samples $n = |QAs|$ permutations for each predicate, producing linearly imbalanced training data in which instance frequency is proportional to the number of QAs in its output (**Linear-Permutations**).

We train QA-SRL and QANom baseline models using each of the above mentioned linearization methods. These models differ both in the semantic task they tackle (i.e. verbs vs. nominalizations) and in the training data scale; thus, in order to distinguish these two effects, we also experiment with training on a random subset of the verbal QA-SRL training set with the same size as the QANom training set (**QA-SRL small**). Results of comparing the different linearization methods are in Section 5.1.

Training Details We tuned the models' hyperparameters on the validation sets with a grid search, detailed in Appendix A.3. The joint QA-SRL and QANom models were tuned to optimize QANom validation measures.

[5] All datasets related to the QASem paradigm have been uploaded to Huggingface's dataset hub, while unifying their data format to the extent possible — see the datasets at https://huggingface.co/biu-nlp.

[6] To gain a more complete perspective, we refer readers to other similar output-linearization explorations (Chen et al., 2021; Lopez et al., 2021).

[7] We use this order: *What*, *Who*, *When*, *Where*, *How*, *Why*.

[8] To avoid memory overflow, we restrict the number of incorporated permutations by $M = 10$.

		QA-SRL Full			QA-SRL Small			QANom		
		P	R	F1	P	R	F1	P	R	F1
Random-Order	**UA**	74.1	58.6	65.5	65.4	60.0	62.6	65.0	52.1	57.9
	LA	61.6	48.7	54.4	50.3	46.1	48.1	45.1	36.1	40.1
Role-Order	**UA**	76.3	64.4	**69.9**	68.4	59.1	63.4	61.3	56.8	58.9
	LA	63.7	53.8	**58.4**	52.0	45.0	48.2	43.1	39.9	41.4
Answer-Order	**UA**	74.7	63.8	68.8	69.6	58.4	**63.5**	65.6	53.6	59.0
	LA	62.5	53.3	57.6	53.4	44.9	48.8	45.7	37.3	41.1
All-Permutations	**UA**	63.1	64.8	64.0	66.1	59.1	62.4	62.7	53.6	57.8
	LA	51.0	52.3	51.6	52.9	47.3	**50.0**	44.3	37.8	40.8
Fixed-Permutations	**UA**	75.2	60.0	66.7	65.8	58.3	61.8	62.0	52.8	57.1
	LA	62.2	49.6	55.2	50.6	44.8	47.6	44.4	37.9	40.9
Linear-Permutations	**UA**	72.5	62.8	67.3	64.3	60.0	62.1	61.5	57.0	**59.2**
	LA	60.9	52.7	56.5	50.5	47.1	48.8	43.1	40.0	**41.5**

Table 3: Output linearization experiment results for the baseline models, comparing different methods for linearizing the set of QAs into output sequence(s). *QA-SRL Full* refers to training on the full QA-SRL training set, while *QA-SRL Small* refers to training on a sample whose size is equivalent to QANom training set.

		QA-SRL Test			QANom Test		
		P	R	F1	P	R	F1
Role-Order	**UA**	73.1	61.3	66.7	65.7	53.5	59.0
	LA	60.5	50.7	55.2	49.2	40.1	44.2
Answer-Order	**UA**	76.2	62.4	**68.6**	64.9	54.4	**59.2**
	LA	63.9	52.4	**57.6**	48.1	40.2	43.8
Linear-Permutations	**UA**	72.7	60.9	66.3	64.3	54.8	**59.2**
	LA	60.7	50.9	55.4	48.6	41.4	**44.7**

Table 4: Output linearization experiment results for the joint QA-SRL–QANom models.

5 Results

In this section, we present the experiments we conducted on the QASem tasks and the corresponding results. We start with results of the experiment testing different linearization methods, and then discuss final performance of best models. We conclude by assessing out-of-domain generalization.

5.1 Linearization Experiment

As can be seen in Table 3, selecting a coherent ordering scheme (**Role-Order** or **Answer-Order**) consistently improves performance over the random-order baseline. In addition, augmenting the training data with permutations, especially using a linear bias toward longer sequences, enhances performance for QANom, but is harmful for QA-SRL.[9] This may be attributed to some extent to the difference in train set scale — when abundant training samples are available, data augmentation is less effective and has lower priority compared to output's structural consistency. However, the "medial" effect on **QA-SRL small**, where augmentation methods exhibit moderate deterioration, suggest that the contrast might also be attributed to the verbal vs. nominal distinction; for example, to nominalizations' more flexible argument structure (Alexiadou, 2010), positing output order consistency less effective than for verbal predicates.

The latter conjecture is supported by an additional linearization experiment we applied on the joint learning setting, whose results are shown in Table 4. While testing on nominal predicates favors the order-invariant, permutation-based method, the same model benefits the most from the **Answer-Order** method when testing on verbal predicates.

Overall, our experiment indicates that linearization techniques have a substantial effect on predicting semi-structured outputs (e.g. sets) with seq2seq models. In the next subsection, we compare our best models to prior QASem models.

5.2 Models Performance

QA-SRL and QANom Table 5 presents evaluation measures of the best performing model per setting from the previous subsection.[10] We can see that the T5-based models are improving over the previous approach with a noticeable margin, especially with respect to question quality (**LA**). Notably, the argument-detection (**UA**) improvement

[9] We have also applied the permutation-based methods on QADiscourse; however, none of these improved performance over the baseline model.

[10] That is — **Linear-Permutations** for the QANom models, **Role-Order** for QA-SRL Baseline, and **Answer-Order** for the joint model tested on QA-SRL.

model		QA-SRL			QANom		
		P	R	F1	P	R	F1
Fitzgerald et al. (2018)	**UA**	79.1	60.1	68.3	45.1	61.5	52.0
	LA	53.8	40.9	46.4	29.6	40.4	34.2
T5 baseline	**UA**	76.3	64.4	**69.9**	61.3	57.5	**59.4**
	LA	63.7	53.8	**58.4**	44.6	41.8	43.1
T5 joint	**UA**	76.2	62.4	68.6	64.3	54.8	59.2
	LA	63.9	52.4	57.6	48.6	41.4	**44.7**

Table 5: Final results of parsing verbal QA-SRL and nominal QA-SRL (QANom). Test sets are from (Roit et al., 2020) and (Klein et al., 2020) respectively.

	UQA			LQA	Prefix
	P	R	F1	Accuracy	Accuracy
Pyatkin et al. (2020)	80.8	86.8	83.7	66.6	49.9
Ours (T5)	87.0	84.3	**85.6**	**73.3**	**57.8**

Table 6: Evaluation results on the QADiscourse test set.

for QANom is much more profound than for QA-SRL. We ascribe this to its smaller training size, putting more weight on the pre-training phase.[11]

As for the joint learning of verbal and nominal predicates, it seems to have a positive effect only for question quality in the nominal domain. This can also be attributed to training size — whereas verbal QA-SRL is slightly impaired from adding nominal instances to the training data, the benefit of nominal predicates from significantly enlarging the training set overcomes this adverse effect.

Overall, turning to T5 improved both QA-SRL and QANom **LA** F1 performance by over 25% compared to previous state-of-the-art parsers, while joint learning gains another 9% recall and 4% F1 for QANom.[12]

QADiscourse Performance evaluation of our QADiscourse model over the QADiscourse task, compared to the previous pipeline model (Pyatkin et al., 2020), is reported in Table 6. While unlabeled detection of discourse relations is improving by a relatively small margin, the question quality — assessed by the LQA and prefix accuracy metrics — is substantially increased. Results suggest that the model is leveraging the generative language modeling pre-training, possibly making its generated question-answer statements more semantically sound, as may also be entailed from the large increase in precision (8%).

5.3 Out-of-Domain Generalization

Finally, to estimate the expected performance of our parser in a realistic downstream scenario, we conducted an experiment tackling out-of-domain generalization. The QA-SRL training set, taken from the large-scale QA-SRL corpus released by Fitzgerald et al. (2018), includes 3 considerably diverse domains — encyclopedic (WIKIPEDIA), news (WIKINEWS) and scientific text books (TQA). The evaluation set is comprised only of the first two. While the models reported so far were trained on all available domains, in order to compare in-domain and out-of-domain generalization more carefully, we trained models on each domain separately and evaluated against single-domain test sets. We bound the training set sizes to that of **QA-SRL Small** (for comparability with Table 3), use **Answer-Order** linearization, and perform the same grid hyper-parameter search procedure (Appendix A.3).

Results (Table 7) indicate that while best performance is obtained using in-domain training data, out-of-domain performance decreases by merely 3.0–0.7 F1 points. This implies that models generalize quite robustly to out-of-domain corpora, which is encouraging for downstream usage.

6 Analyses

Output Validity As mentioned in Section 2.2, QA-SRL questions adhere to a specialized constrained format. It is therefore not trivial for a model pre-trained on free natural language to acquire these format specifications. Nevertheless, we observe that the models have robustly internalized

[11] The model version we used for the prior QA-SRL model (Fitzgerald et al., 2018) is using ELMo contextualized embeddings (Peters et al., 2018), which although belonging to the pre-trained language-model regime, are significantly weaker compared to more recent PLMs (Devlin et al., 2019).

[12] Taking memory efficiency into account, our QASem tool uses the **Answer-Order** joint model for both QA-SRL and QANom by default, fetching a single model for both types of predicates.

Train Domain	Test Domain		P	R	F1
Wikipedia	*Wikipedia*	UA	*71.0*	*57.8*	*63.7*
		LA	*58.0*	*47.2*	*52.1*
TQA	Wikipedia	UA	72.2	55.7	62.9
		LA	58.4	45.1	50.9
Wikinews	Wikipedia	UA	72.0	56.0	63.0
		LA	58.3	45.3	51.0
Wikinews	*Wikinews*	UA	*74.4*	*66.3*	*70.1*
		LA	*61.1*	*54.4*	*57.5*
TQA	Wikinews	UA	73.4	63.8	68.3
		LA	58.6	51.0	54.5
Wikipedia	Wikinews	UA	68.4	65.9	67.1
		LA	55.4	53.3	54.3

Table 7: QA-SRL evaluation results of in-domain (*italics*) vs. out-of-domain test settings.

the special grammar of the QA sequences. Only a small fraction (1.2%) of output QA-SRL/QANom QAs were automatically detected as not conforming with QA-SRL specifications, of which vast majority ($> 95\%$) are due to answer–sentence misalignment mostly owing to tokenization issues (e.g. answer token is out-of-vocabulary).

Manual Error Analysis Prior works on QA-SRL have acknowledged that the automatic evaluation metrics are under-estimating true performance figures (Roit et al., 2020; Klein et al., 2020). We inspected the joint model predictions on the verbal and nominal QA-SRL test sets, taking samples of 50 QAs automatically classified as precision mistakes, and of 50 gold-standard QAs classified as recall misses (200 QAs total).

Our findings are detailed in Appendix A.4. To summarize the verbal QA-SRL findings, we conclude that 42% of the precision mistakes are actually acceptable answers, whereas 40% of counted recall mistakes have correct counterparts in model predictions, both of which erroneously rejected by the strict alignment-based evaluation metric. Acceptable mistakes are often caused by taking different span-selection decisions, or by an argument structure having multiple correct interpretations. Unacceptable mistakes commonly concern answer-repetition, verb-particle constructions, and missing harder implied arguments. Overall, considering this manual analysis, the joint model interpolated **UA** precision on QA-SRL is **87.0** while recall is **78.6**. Interpolated **UA** for QANom is much lower — **77.2** precision, **62.0** recall — leaving room for future improvements.

QA Position Effect A further analysis, reported in Appendix A.5, examines the effect of position in generated sequence on QA quality.

7 Conclusion

We propose to bundle three QA-based semantic tasks into a congruent conceptual paradigm. We hence develop and release new state-of-the-art models for these tasks, based on a unified framework for fine-tuning a seq2seq pre-trained language model. Specifically, we show the importance of output linearization choices, including permutation-based data augmentation techniques, and propose using joint learning of verbal and nominal QA-SRL for further enhancing performance in medium-size dataset settings. We further demonstrate these models' out-of-domain robustness.

Utilizing these models, the QASem tool we release can be used in various downstream scenarios where an explicit account of textual information units is desired. For example, the recent trend of leveraging QAs as an intermediate representation for various summarization-related tasks indicates the perceived attractiveness of this "open" representation style. In particular, questions and answers provide a natural linguistic mechanism for explicitly focusing on concrete information units. However, the common QA datasets (e.g. SQuAD; Rajpurkar et al., 2016), over which prior QA representations have been trained, were developed for modeling QA or reading comprehension as end-tasks, but were not designed to provide a systematic semantic representation. Hence, QA models trained on such datasets yield a non-systematic set of QAs, which might introduce overlapping and non-exhaustive information units, hindering their downstream utility. On the other hand, QASem is designed to produce a systematic — i.e. consistent and comprehensive — set of QAs, each targeting an atomic "statement" concerning different predications, thus providing a more precise representation of semantic structure.

Future work would incorporate upcoming QASem tasks regarding adjectives and noun modifiers into the current seq2seq framework. Further, we plan to explore sentence-level modeling for predicting all QASem QAs jointly.

8 Limitations

Our QASem model is built upon vanilla T5. Nevertheless, many question-answering datasets exist, which could quite probably enhance QASem parsing through some multitask or transfer learning setting. Although we have had a preliminary experiment with a pre-trained question-generator

model, yielding negative results, a more careful exploration of this path seems promising. In particular, one could leverage QA datatsets to pre-train a text-to-text model on QA-generation in the same output format as our QASem tasks.

An essential limitation of the current approach is that model outputs do not assign any confidence score for generated QAs. This seem like a crucial feature to have for deployment in downstream systems, e.g. for controlling over the precision/recall trade-off. As posterior probabilities of generated tokens are conditioned on all previous tokens in the sequence, it is not trivial to deduce a confidence score for a sub-sequence. Hence, the challenge of confidence estimation for semi-structured predictions using seq2seq warrants further research.

Acknowledgements

This work was supported in part by grants from Intel Labs, the Israel Science Foundation grant 2827/21, and the PBC fellowship for outstanding PhD candidates in data science.

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 (Volume 1: Long Papers)*, pages 228–238.

Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 77–89.

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.

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 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Johannes Bjerva. 2017. Will my auxiliary tagging task help? estimating auxiliary tasks effectivity in multi-task learning. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 216–220, Gothenburg, Sweden. Association for Computational Linguistics.

Johan Bos, Valerio Basile, Kilian Evang, Noortje J Venhuizen, and Johannes Bjerva. 2017. The groningen meaning bank. In *Handbook of linguistic annotation*, pages 463–496. Springer.

Cristina Butnariu, Su Nam Kim, Preslav Nakov, Diarmuid O Séaghdha, Stan Szpakowicz, and Tony Veale. 2009. Semeval-2010 task 9: The interpretation of noun compounds using paraphrasing verbs and prepositions. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 100–105.

Jifan Chen and Greg Durrett. 2021. Robust question answering through sub-part alignment. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1251–1263.

Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. 2021. Pix2seq: A language modeling framework for object detection. In *International Conference on Learning Representations*.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2):281–332.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Yanai Elazar, Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2021. Text-based np enrichment. *arXiv preprint arXiv:2109.12085*.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4186–4196, Hong Kong, China. Association for Computational Linguistics.

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

Mantas Gavenavicius. 2020. Evaluating and comparing textual summaries using question answering models and reading comprehension datasets. B.S. thesis, University of Twente.

Nizar Habash and Bonnie Dorr. 2003. CatVar: a database of categorial variations for English. In *Proceedings of Machine Translation Summit IX: System Presentations*, New Orleans, USA.

Hardy Hardy and Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using abstract meaning representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 768–773.

Hangfeng He, Qiang Ning, and Dan Roth. 2020. QuASE: Question-answer driven sentence encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8743–8758, Online. Association for Computational Linguistics.

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

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q2:: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870.

Yin Jou Huang and Sadao Kurohashi. 2021. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3046–3052.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Hans Kamp, Josef van Genabith, and Uwe Reyle. 2011. Discourse representation theory. In *Handbook of philosophical logic*, pages 125–394. Springer.

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

Ayal Klein, Jonathan Mamou, Valentina Pyatkin, Daniela Stepanov, Hangfeng He, Dan Roth, Luke Zettlemoyer, and Ido Dagan. 2020. QANom: Question-answer driven SRL for nominalizations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3069–3083, Barcelona, Spain (Online). International Committee on Computational Linguistics.

I-Ta Lee and Dan Goldwasser. 2019. Multi-relational script learning for discourse relations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4214–4226.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract meaning representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086.

Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and Charibeth Cheng. 2021. Simplifying paragraph-level question generation via transformer language models. In *PRICAI 2021: Trends in Artificial Intelligence: 18th Pacific Rim International*

Conference on Artificial Intelligence, PRICAI 2021, Hanoi, Vietnam, November 8–12, 2021, Proceedings, Part II, pages 323–334.

William C Mann and Sandra A Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.

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

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.

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

Eleni Miltsakaki, Rashmi Prasad, Aravind K Joshi, and Bonnie L Webber. 2004. The penn discourse treebank. In *LREC*. Citeseer.

Muhidin Mohamed and Mourad Oussalah. 2019. Srl-esa-textsum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing & Management*, 56(4):1356–1372.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Dipanjan Das, and Mirella Lapata. 2022. Conditional generation with a question-answering blueprint. *arXiv preprint arXiv:2207.00397*.

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.

Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021. Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4768–4779.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. 2020. QADiscourse - Discourse Relations as QA Pairs: Representation, Crowdsourcing and Baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2804–2819, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Fynn Schröder and Chris Biemann. 2020. Estimating the influence of auxiliary tasks for multi-task learning of sequence tagging tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2971–2985, Online. Association for Computational Linguistics.

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

Wei Shi and Vera Demberg. 2019. Learning to explicitate connectives with seq2seq network for implicit discourse relation classification. In *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*, pages 188–199.

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, Austin, Texas. Association for Computational Linguistics.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723.

Frances Yung, Vera Demberg, and Merel Scholman. 2019. Crowdsourcing discourse relation annotations by a two-step connective insertion task. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 16–25.

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

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733.

A Appendices

A.1 Detailed Evaluation Metrics

Evaluating QA-based semantic tasks involves two core aspects. First, we would like to estimate how many of the *semantic relations* are captured correctly. For SRL, this is analogous to measuring argument detection, while for discourse, it assesses whether pairs of events are related to each other or not. Second, given that the model identified the same predicate-argument or predicate-predicate relation as present in the gold set, we want to assess its predicted label for the relation type (semantic role or discourse relation sense). A manifestation of these objectives for the QA-SRL and QADiscourse formats considers an *unlabeled* and a *labeled* evaluation measure per task (Roit et al., 2020; Pyatkin et al., 2020).

For computing QA-SRL's unlabeled argument detection (**UA**) metric, QAs in the predicted set are aligned to QAs in the reference set using maximum bipartite matching based on lexical intersection-over-union (IOU) of the answers. A pair of QAs must surpass a minimum IOU threshold Γ to count as aligned. Then, aligned QA pairs are re-inspected for question equivalence to form the labeled argument detection measure (**LA**).

QA-SRL question templates have no plain mapping to semantic roles, and determining whether two questions refer to the same role is non-trivial. Thus, previous QA-SRL works have proposed different heuristics for evaluating approximated question equivalence. Here we apply the evaluation measures put forward by Klein et al. (2020), using a technique for mapping questions into a discrete space of "syntactic roles", and setting $\Gamma = 0.3$. We apply it on both QA-SRL and QANom to have comparable figures.

As for QADiscourse, we simply embrace the **UQA** and **LQA** metrics proposed by Pyatkin et al. (2020). These are analogous to **UA** and **LA**, with minor adaptations. The unlabeled alignment between QA pairs is computed as IOU between question-and-answer tokens jointly ($\Gamma = 0.5$), excluding question prefix, because the question words denote which proposition is participating in the discourse relation with the answer. In addition, labeled alignment is simply a match over question prefixes, since unlike QA-SRL question, these question prefixes do map into relation senses.

A.2 Alternative QA-SRL Input Linearization Methods

Here we specify in greater detail about experiments we ran assessing alternative linearization methods for QA-SRL and QANom models.

Concerning the input encoding, we experimented with four methods of highlighting the target predicate token within the sentence:

1. Repeating the target word at the end of the sequence
2. Special token before the target
3. Special token after the target
4. Special tokens before and after the target

Method 4. outperformed methods 2. and 3. by a small margin, while method 1. was worse.

A.3 Training Details

In our preliminary experiments, model training was shown to be quite sensitive to hyper-parameter tuning. Nevertheless, it is impractical to execute a wide hyper-parameter search to test each linearization method. Instead, for the small training-set

experiments (QANom and **QA-SRL Small**) we constrained the tuning phase to a small grid search:

$$\text{learning rate} \in \{0.001, 0.005, 0.01\}$$

$$\text{dropout rate} \in \{0.1, 0.15\}$$

$$\text{effective batch size} \in \{96, 168\}$$

As the training set of **QA-SRL Full** is 14-times larger, even this grid-based method has been unaffordably expensive. This also applies for the joint model's training process. Thus, for these settings we fix the hyper-parameters throughout all linearization methods, using:

$$\text{learning rate} = 0.005$$

$$\text{dropout rate} = 0.1$$

$$\text{effective batch size} = 96$$

All models were fine-tuned for 20 epochs, with `fp16` mode, and used a beam size of 5 for decoding.

A.4 Manual Error Analysis

As mentioned in Section 6, we have manually inspected the joint model predictions on the both (verbal) QA-SRL and nominal QA-SRL (QANom) test sets. For each task, we took a sample of 50 QAs automatically classified by the UA measure as precision mistakes, and a sample of 50 gold-standard QAs classified as recall misses.

QA-SRL We judged 21 / 50 of precision mistakes (42%) as acceptable answers, and 20 / 50 (40%) of recall mistakes as having correct counterparts in model predictions.

These are mostly characterized by the fact that the model concatenates answers while the gold-standard has a better separation of answers. For example, the gold-standard contains the pair *Q: Who pleaded something? A: ['Co-defendant', 'Daniel Spitler']*, while the model's prediction has the same question with the concatenated version of the answer *'Co-defendant Daniel Spitler'*. Another common type of the acceptable mistakes is where two QAs (i.e. roles) can be alternatively captured by a single QA. For example, for the sentence *The company also **announced** Daniel Ammann as its new president*, the gold-standard contains: *Q: Who did someone announce as something? A: Daniel Ammann ; Q: What did someone announce someone as? A: its new president*. In contrast, the model predicts *Q: What did someone announce? A: Daniel Ammann as its new president*.

With respect to genuine mistakes, some precision errors occur in sentences with phrasal verbs, such as *'come across'* or *'carry out'*, where the model fails to ask the correct question using the verb particle construction. On the other hand, we observed that several recall errors are regarding adjuncts occurring in a non-standard position; for instance, for the sentence: *The top deck of the bus was crushed on one side after hitting the truck and **spinning***, the models misses the following gold QA — *Q: When did something spin? A: after hitting the truck*. Quantitatively, gold-standard questions starting with *Why* or *How* have a better chance of being missed by the model, in line with their stronger reliance on common-sense reasoning skills.

QANom The automatic evaluation for QANom have been more accurate. We judged 18 / 50 of precision errors (36%) as acceptable QAs, and only 8 / 50 of recall errors (16%) as having correct counterparts in model predictions.

For QANom, acceptable precision mistakes are often due to incomplete coverage of the gold annotations. For example, annotations for the sentence *Alex Neil, the Scottish cabinet **minister** responsible for the legislation, said: " This is a historic moment for equality in Scotland"* are missing the following model-generated QA — *Q: Where did someone minister something? A: Scotland*. Another common cause of acceptable mistakes are slight variations in phrasing in the question-answer pair. An example is the following gold-standard QA — *Q: Where did someone legislate? A: In Scotland* — compared to the following prediction: *Q: Where did someone legislate something? A: Scotland*.

The genuine precision mistakes are characterized by the model generating questions that have no answer in the sentence, thus aligning it to an unfaithful answer. For example, for the predicate *Prosecutors claim political **assassinations** and suicide attacks were planned*, one of the model-generated QAs is *Q: Who assassinated someone? A: Prosecutors*. Once such a question is generated, a generation of an answer will inevitably lead to a mistake. Similarly to QA-SRL, recall mistakes commonly concern implicit arguments, which are more frequent at the QANom dataset compared to QA-SRL (Klein et al., 2020). For example, for

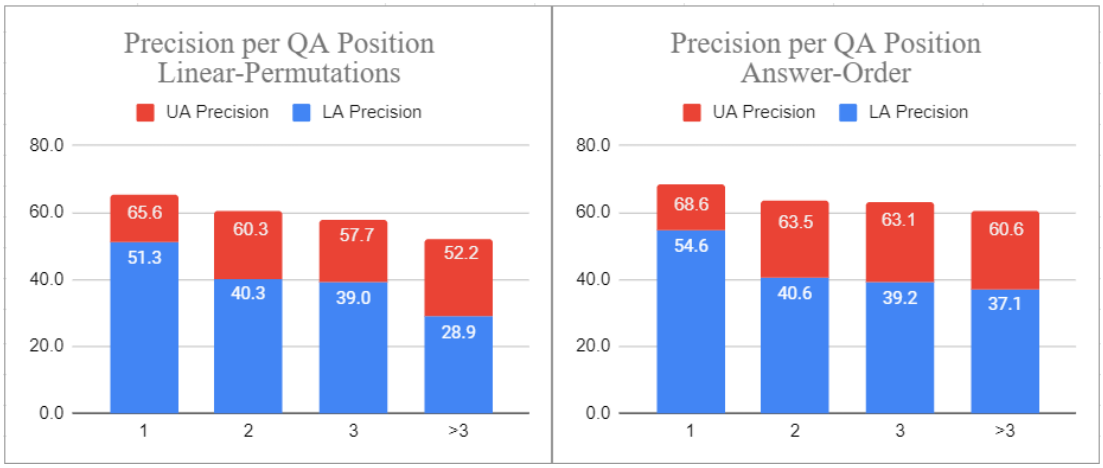

Figure 1: Predicted QA Precision (y axis) per QA position in output sequence (x axis).

the sentence *As a **protest** against the punishment, Issawi began a publicized hunger strike*, the model misses the following gold-standard QA — *Q: How did someone protest? A: began a publicized hunger strike*.

A.5 QA-Position Impacting Model Precision

In this section we investigate how QA position in output sequence affects generation quality, and whether output linearization methods interact with these effects.

Taking QANom-Baseline as our model, we analyze the precision of predicted QAs with respect to their position in the output sequence. Results for the **Answer-Order** and **Linear-Permutations** output linearization methods are plotted in Figure 1. There is a clear effect of the QA's position on its accuracy — QAs generated first by the auto-regressive decoder have higher quality than those generated last. A consequence, also quantitatively observed in model predictions, is that predictions for predicates with many true arguments would have lower precision than those with few arguments.

Interestingly, the above mentioned effect is mitigated when training on a fixed linearization order (**Answer-Order**) rather than on permutations. This may be caused by the fact that, following the fixed order of QAs with respect to answer position in sentence seen during training, the model is learning to "constrain" itself to predicting spans from the "remaining" part of the sentence, narrowing its false-positive choices.