Explicating the Implicit: Argument Detection Beyond Sentence Boundaries

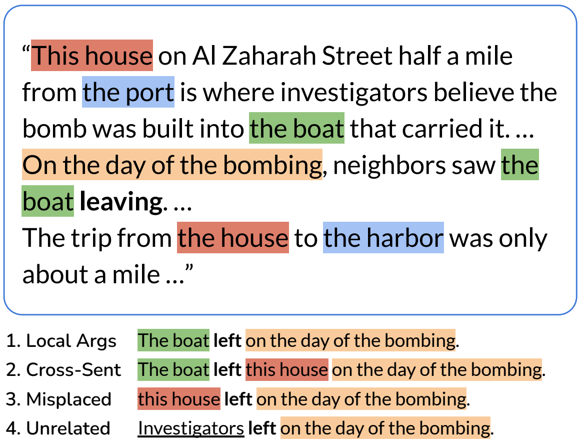
Anonymous ACL submission

Abstract

Detecting semantic arguments of a predicate word has been conventionally modeled as a sentence-level task. The typical reader, however, perfectly interprets predicate-argument relations in a much wider context than just the sentence where the predicate was evoked. In this work, we reformulate the problem of argument detection through textual entailment to capture semantic relations across sentence boundaries. We propose a method that tests whether some semantic relation can be inferred from a full passage by first encoding it into a simple and standalone proposition and then testing for entailment against the passage. Our method does not require direct supervision, which is generally absent due to dataset scarcity, but instead builds on existing NLI and sentence-level SRL resources. Such a method can potentially explicate pragmatically understood relations into a set of explicit sentences. We demonstrate it on a recent document-level benchmark, outperforming some supervised methods and contemporary language models.

1 Introduction

Identifying which entities in text play certain semantic role with respect to a given predicate (i.e. a verb) is a core ability of language comprehension (Fillmore, 1976). Such basic semantic information is often surfaced via simple lexical and syntactical patterns in the sentence. Readers however can perfectly interpret such semantic relations pragmatically in a wider context. Consider the example in Figure 1. The entity being paid can be resolved as the manufacturer by deduction, and ‘the deposit’ is understood as the currency that changes ownership in exchange for recycling. These examples showcase where semantics departs from syntax, and allow us to systematically investigate how humans and machines reason over events in text in cases where they cannot rely on easy-to-follow grammatical patterns.



“This house on Al Zaharah Street half a mile from the port is where investigators believe the bomb was built into the boat that carried it. …
On the day of the bombing, neighbors saw the boat **leaving**. …
The trip from the house to the harbor was only about a mile …”

1. Local Args The boat **left** on the day of the bombing.
2. Cross-Sent The boat **left** this house on the day of the bombing.
3. Misplaced this house **left** on the day of the bombing.
4. Unrelated Investigators **left** on the day of the bombing.

Figure 1: Example of semantic arguments in the sentence and document scope. The predicate is in boldface while arguments are highlighted in color. The bottom part shows four different propositions: (1) A proposition constructed from in-sentence arguments of the predicate. (2) The same proposition with additional arguments from anywhere in the document (3) A proposition with some arguments (the house) placed in incorrect syntactic position, that do not align with their semantic role. (4) A proposition with a non-argument phrase. Both (3) and (4) are not supported by to the document.

In this work[1], we address detecting cross-sentence semantic arguments for verbal and deverbal noun predicates. We propose a method based on textual-entailment (Dagan et al., 2005) and supervised only with NLI and sentence-level Semantics Role Labeling (SRL) (Gildea and Jurafsky, 2000) data. It takes a document and a marked predicate and outputs a set of simple, easy-to-grasp sentences that incorporate semantic arguments from anywhere in the document (e.g. ‘the house’ argument from a different sentence incorporated into the leave event in Figure 1, proposition 2). We assume that an argument is omitted by the speaker from the predicate’s sentence due to its redundancy

[1] The codebase, dataset, and models will be made publicly available in the non-anonymous version of this manuscript.

in discourse while re-inserting it back into its designated position next to the predicate should not alter the meaning of the event in the passage. Our basic idea is that a simple proposition constructed from a set of true arguments should be entailed from the passage (see props 1-2 in Figure 1), while any proposition that targets the same predicate and contains a non-argument phrase or a misplaced phrase should not be entailed (see props 3-4 in Figure 1). Therefore, we design a method that starts at the local parse of the predicate, builds a proposition from the extracted in-sentence arguments, and then examines candidate phrases one by one from across the document by inserting them into different positions and testing for entailment. Our method does not require a frame repository such as PropBank (Palmer et al., 2005) or FrameNet (Baker et al., 1998) to operate. Instead, it uses the explicit syntactic argument structure in the proposition as a syntactic surrogate for the underlying semantics of the predicate in the passage (see how the meaning changes in the misplaced argument example, prop 3 in Figure 1).

Some recent works from the event extraction literature apply similar slot-filling (Li et al., 2021) or entailment-based methods (Sainz et al., 2022; Lyu et al., 2021). However, they rely on a limited event ontology for predefined templates for argument extraction. In contrast, our work uses English syntax for creating propositions, akin to the clause structure in Del Corro and Gemulla (2013).

This generally illuminates another benefit of our approach, being schema-free, the propositions can be easily processed downstream by parsers trained on abundant single-sentence data, for example for relation extraction (Hendrickx et al., 2010) or event participant detection (Doddington et al., 2004). Thus, explicating to downstream tasks the set of document-level semantic relations that were previously unreachable, now encoded in a simple sentence form.

To summarize, our contributions include a novel distantly supervised argument detection method based on combining Textual Entailment with SRL analysis. Our implementation achieves higher performance than supervised models on a document-level dataset (Elazar et al., 2022) for noun-phrase relations, and outperforms other approaches on a re-annotated benchmark for verbal predicates (Moor et al., 2013).

2 Background and Related Works

Implicit Arguments Mainstream research efforts in semantic role labeling (SRL) (Gildea and Jurafsky, 2000; Kingsbury and Palmer, 2002) have focused on the problem of assigning semantic roles only to syntactically related phrases, e.g. the subject or object phrases of verbs, while neglecting constituents from the wider passage that are pragmatically interpreted as participants. The latter ones, referred to as *implicit* arguments (Gerber and Chai, 2010; Ruppenhofer et al., 2010) despite being overtly understood by readers, constitute a sizeable portion of the potentially identified argument set (Klein et al., 2020; Roit et al., 2020; Gerber and Chai, 2010; Fillmore, 1986). While some recent works (FitzGerald et al., 2018) have annotated large datasets with semantic arguments captured anywhere within the sentence scope, to this day, only a handful of limited resources for SRL in the document scope exist (Gerber and Chai, 2010; Moor et al., 2013; Ruppenhofer et al., 2010; Feizabadi and Padó, 2015). Some resources contain only a few hundred instances, others lack diversity, capturing only a tiny set of predicates (5-10 unique verbs), and all focused only on semantic core roles (i.e. the numbered arguments in PropBank), neglecting other meaningful information for the reader such as temporal or locative modifiers. O'Gorman et al. (2018) annotated a dataset of cross-sentence arguments on top of AMR graphs (Banarescu et al., 2013) specifying arguments as AMR concepts, without their exact location in the sentence.

Earlier supervised models for implicit SRL relied on extensive feature engineering and also using gold features (Gerber and Chai, 2012). Many works additionally attempted to overcome data scarcity by creating artificial training data using coreference (Silberer and Frank, 2012) or aligning predicates in comparable documents (Roth and Frank, 2015), Cheng and Erk (2018) proposed to transform the problem into a narrative cloze task, creating synthetic datasets. More recently, Zhang et al. (2020) improved upon the baseline model proposed for the RAMS dataset (Ebner et al., 2020), and trained a supervised model that detects argument heads before expanding to the full constituent.

QA-SRL (He et al., 2015) represents the label of each semantic argument as a simple Wh-question that the argument answers, for example, *'Who acquired something?'* encodes the agent, and *'Who* *did someone give something to?'* encodes the recipient. These question-labels point at the syntactic position of the argument in a declarative form of the QA pair, e.g.: 'The agent acquired something' or 'Someone gave something to the recipient' (see the example in Figure 2, top-left, where the position of the answer is apparent from the question). Each question also encodes the tense of the event, the modality, and negation properties (might the event occur or has the event occurred?) which are used to instantiate our propositions. Klein et al. (2020) extended QA-SRL to deverbal nominal predicates, recently leveraged for training a joint verbal and nominal QA-SRL model (Klein et al., 2022).

TNE is a dataset for modeling semantic relations between noun phrases (NPs) across a document and is annotated on top of Wikipedia. A relation consists of an anchor and complement phrases that are labeled with a preposition, i.e. [the investigation]$_{\textsc{Anchor}}$ *by* [the police]$_{\textsc{Complement}}$. Each document is first segmented into a list of non-overlapping NPs and every NP pair is annotated with either a preposition or a 'no-relation' tag. Each NP is also assigned to a cluster of co-referring within-document mentions.

ON5V (Moor et al., 2013) is a dataset containing 390 instances of five different verbal predicates, selected from 260 documents from the development and train partitions in OntoNotes (Pradhan and Xue, 2009). Original annotation only filled vacant core roles (i.e. numbered, ARG0, ARG1) with the closest argument phrase that fits the role description.

3 Method

Our approach for identifying semantic arguments of a predicate is based on verifying the correctness of an assignment of phrases to semantic roles. In a nutshell, we represent the assignment as a simple proposition, named the *semantic hypothesis*, that consists of phrases placed in subject or object positions according to their roles. If the semantic hypothesis is entailed from the passage, we conclude that the relations encoded within the proposition are also present in the document. For example, in proposition no. 2 in Figure 1 'The boat' is placed into the subject position and represents the leaver, while 'on the day of the bombing' is placed as an adjunct and represents the time of the event. On the other hand, incorrectly placing an argument phrase, 'this house' (proposition no. 3), in the subject position would assign an incorrect semantic role to the phrase, and should not be entailed.

We apply this verification procedure in a multi-step process as follows. First, we retrieve a set of semantic arguments in the vicinity of the predicate, leveraging highly performant parsers trained on in-sentence data (FitzGerald et al., 2018; Klein et al., 2020). Next, we verify their correctness to prevent parsing errors from propagating to later stages (Figure 2, top-center), and construct a base hypothesis from the verified arguments (see the top row in Figure 2). Finally, we insert different candidate phrases from any sentence into different syntactic positions in the base hypothesis and verify the resulting hypotheses independently (see the bottom part of Figure 2). The first step ensures that the base hypothesis describes the target event by referring to the predicate's local arguments (the boat and the time of leaving), while the last step expands the argument set by integrating new phrases into the base hypothesis (the house and the port). In the next subsections, we will describe in detail the structure of the semantic hypothesis, how we initialize it, and how we expand it to include arguments beyond the scope of the predicate's sentence.

3.1 The Semantic Hypothesis

The semantic hypothesis is a simple English declarative sentence centered around a verb. It is constructed from the main verb and a set of phrases assigned to unique syntactic positions, and can be modified with tense, modality, and negation properties. The supported positions are directly related to the verb and include: subject (`SUBJ`), direct object (`DOBJ`), indirect object (`IOBJ`), and adjunct (`ADJ`). The last two may be preceded with an optional preposition. In our implementation, we construct a sentence in active voice if a subject phrase is present, or resort to passive voice if not. Generally, we apply the following templates: `SUBJ-VERB-DOBJ-IOBJ-ADJ`, or `DOBJ-VERB`$_{\texttt{passive}}$`-IOBJ-ADJ` and fill the phrases as required. Given these specifics, we determine the corresponding verb inflection, and when necessary use the plural form to agree with the subject and modify any auxiliary verbs accordingly. We note that a valid declarative sentence in English must contain a subject phrase. To satisfy this requirement, we allow unspecified 'placeholder' arguments to be inserted instead of concrete ones, placing 'someone' in an empty subject position or 'something' in empty object positions when neces-

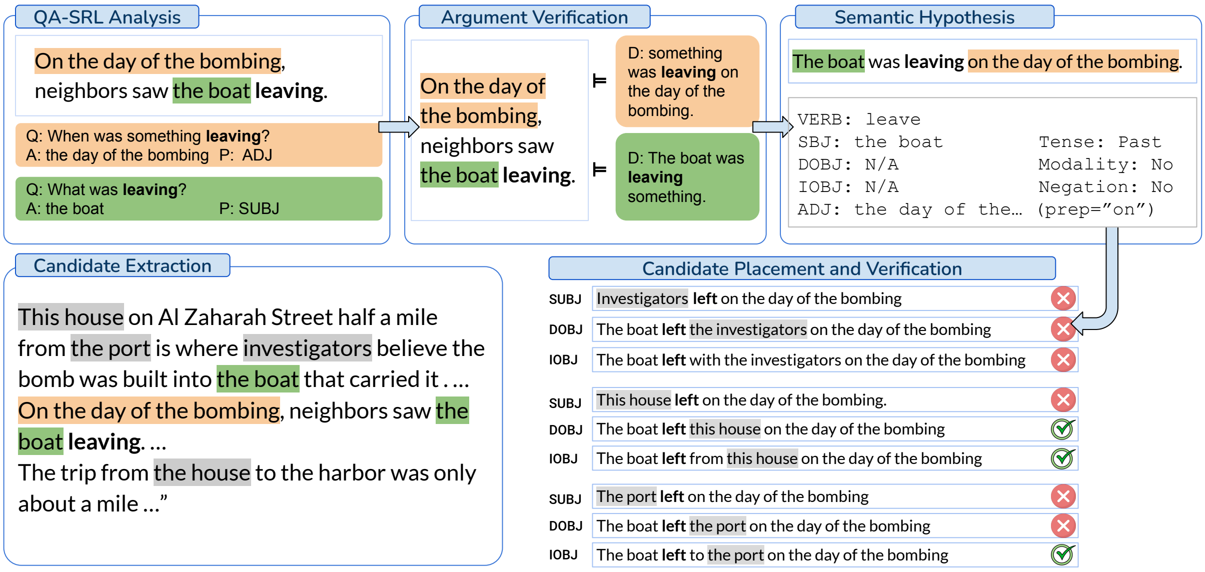

QA-SRL Analysis
On the day of the bombing, neighbors saw the boat leaving.
Q: When was something leaving?
A: the day of the bombing P: ADJ
Q: What was leaving?
A: the boat P: SUBJ
Argument Verification
On the day of the bombing, neighbors saw the boat leaving.
⊨
D: something was leaving on the day of the bombing.
⊨
D: The boat was leaving something.
Semantic Hypothesis
The boat was leaving on the day of the bombing.
VERB: leave
SBJ: the boat
DOBJ: N/A
IOBJ: N/A
ADJ: the day of the… (prep="on")
Tense: Past
Modality: No
Negation: No
Candidate Extraction
This house on Al Zaharah Street half a mile from the port is where investigators believe the bomb was built into the boat that carried it . ...
On the day of the bombing, neighbors saw the boat leaving. ...
The trip from the house to the harbor was only about a mile ..."
Candidate Placement and Verification
SUBJ Investigators left on the day of the bombing ✗
DOBJ The boat left the investigators on the day of the bombing ✗
IOBJ The boat left with the investigators on the day of the bombing ✗
SUBJ This house left on the day of the bombing. ✗
DOBJ The boat left this house on the day of the bombing ✓
IOBJ The boat left from this house on the day of the bombing ✓
SUBJ The port left on the day of the bombing ✗
DOBJ The boat left the port on the day of the bombing ✗
IOBJ The boat left to the port on the day of the bombing ✓

Figure 2: End-to-end example of our suggested argument extraction pipeline. *top-left:* Parsing with QA-SRL, the predicate is marked in bold while the local arguments are highlighted in color. The question (Q) is used to determine the syntactic position (P) of the argument (A). *top-center:* Verification of individual arguments, a proposition is constructed for each, the argument is placed in its designated position from the previous step and placeholders are inserted to other positions. The propositions are examined against the sentence for entailment. *top-right:* Validated arguments construct the base hypothesis; Event attributes such as tense, modality and negation are extracted from the QA-SRL parse and initialize the hypothesis sentence. *bottom-left:* Extracting candidate phrases from the document scope (highlighted in gray). *bottom-right:* Each candidate is inserted into three different positions in the base hypothesis and verified against the full document. The second candidate demonstrates two correct alternations.

sary. Such flexibility enables the system to force a specific valency pattern, for example, construct a transitive clause by inserting 'something' when a concrete direct object is unavailable.

3.2 Hypothesis from Local Arguments

In the first step, we initialize the semantic hypothesis with arguments from the sentence extracted with a highly performant QA-SRL parser (Klein et al., 2022). We retrieve the local arguments as QA pairs and apply a heuristic from Klein et al. (2020) over the questions to determine the syntactic position of each argument-answer in our proposition (Figure 2, top-left). To verify their correctness, we construct a hypothesis for each argument individually and validate them with an entailment model against the original sentence. This validates both the argument and its predicted syntactic position. Finally, the highest-scoring argument in each position is taken to construct the base hypothesis.

3.3 Expanding to Non-Local Candidates

To expand the argument set we inspect candidate phrases from the document. Each phrase is inserted into three positions independently in the base hypothesis: the subject, direct object, and indirect object, forming a new hypothesis on the grounds of the base one. Each is then scored using our NLI model against the document. We select the highest-scoring hypothesis for each candidate, and add it to the output if it passes a configurable threshold. In this work, we consider noun-phrase candidates from the entire document that don't overlap with generated arguments from the first stage.

3.4 Implementation Details

The prepositions for IOBJ or ADJ phrases are assigned in one of two ways: either by inspecting the dependency structure of the predicate's sentence for a connecting preposition or from the QA-SRL analysis, and lastly, if not captured by the preceding methods, using a masked language model that assigns the most semantically probable (Devlin et al., 2019) preposition given the full passage and the hypothesis. Attributes such as tense, modality, and negation are extracted from the local QA-SRL parse of the sentence. For more details see Appendix F

4 Predicate-Argument aware NLI Dataset

Throughout our experiments, we noticed that the readily available NLI models usually make poor decisions when considering different semantic hypotheses, assigning high probability to propositions with unrelated candidates — which resonates the findings of Min et al. (2020); Basmov et al. (2023). We believe that this is caused by the inherent lexical overlap between the hypothesis and the premise texts since our proposition is built entirely from phrases found in the original document. To circumvent this, we train a semantics-aware entailment model from QA-SRL data. We use the single-sentence training data and generate entailed and not-entailed propositions. Each training instance includes a sentence and a proposition centered on a predicate in the sentence. Positive instances include propositions built using the predicate's argument set. Each true argument is placed in the hypothesis according to their syntactic position as determined by their QA-SRL question. The positive propositions are then used to build the negative instances in the following two ways: The first inserts a noun phrase from the sentence that is not an argument into any position. The second switches between syntactic positions of true arguments in the positive proposition, replacing objects as subjects and vice-versa. This training setup encourages the model to be more sensitive to the semantics of the hypothesis, as encoded in its argument structure.

Our training set contains 465K sentence-hypothesis pairs extracted from the training partitions of QANom (Klein et al., 2020) and QAS-RLv2 (FitzGerald et al., 2018), with 30% positive (entailed) instances. Negative instances are split between subject-object swaps (14%), and insertions of non-argument phrases from the sentence (56%). We created multiple positive hypotheses for each predicate by omitting subsets of true arguments, anticipating low coverage conditions of the QA-SRL parser at inference time. For negative examples, we sampled one positive hypothesis for each predicate and applied our augmentations.

5 Experiment Setup

5.1 Evaluation Datasets

We apply our method to verbal and nominal predicates from several document-level benchmarks.

TNE (Elazar et al., 2022). We derive our main benchmark from the TNE dataset. We extract predicate-argument data by focusing on a subset of relations in TNE where the anchor's syntactic head is a deverbal noun, i.e. a nominal predicate, and hypothesize that their complements constitute semantic arguments of the predicate word. To filter the relevant anchors, we apply the nominal predicate classifier of QANom (Klein et al., 2020) with a threshold of 0.75 and identify 10946/1315/1206 predicate instances in the train, development, and test partitions respectively. On average, each deverbal anchor contains 4.5 complement entities, and notably, 2.5 of these have the closest mention to the predicate located in a different sentence. Examining a sample of 50 deverbal anchors we find that out of 275 cross-sentence complement entities, 93% exhibit a semantic relation that can be captured by a QA-SRL question, validating our initial hypothesis.

Our task in this setup is to select all NP complements given a deverbal anchor, the document, and the segmented list of noun-phrase candidates. When applying generative methods, we consider a specific NP candidate from the document as predicted if it matches one of the generated argument phrases, where two phrases match if either they share the same syntactic head or have a high token-wise overlap of above 0.5 Intersection-over-Union (IOU). Otherwise, any non-overlapping generated phrase is discarded.

ON5V (Moor et al., 2013) We also evaluate our method on ON5V. We use the unified set of predicates from both partitions as our evaluation data. To cover the coverage gap for modifier roles we asked an in-house annotator team to go over the existing data and add any argument phrase that can be captured by a QA-SRL question. The resulting dataset has 3271 arguments with 1800 novel cross-sentence mentions that did not belong to any previously annotated entity, emphasizing the need for exhaustive annotation. We refer to Appendix B for more details regarding the annotation protocol.

We use cross-fold validation over 4 folds split by document, we tune the NLI classification threshold over 3 folds and evaluate on the fourth. Results over the test folds are averaged and reported with std. dev. We limit the search for arguments to a context window of 7 sentences, with 5 preceding and 1 subsequent sentence around the predicate. This window follows the annotation scope set for our annotators and was found to be sufficient to locate more than 98% of all originally annotated arguments in the data.

5.2 Evaluation

We follow the methodology proposed by Ruppenhofer et al. (2010) in evaluating document-level argument detection. We assign credit for an argument only once at the entity level, regardless of the number of times it is mentioned in the passage or captured in the system output. Consider for example *the boat* argument from Figure 1, it appears twice in a short passage, and we give it a full score if at least one of these mentions were captured. This disentangles SRL evaluation at the document level from co-reference resolution. Practically, we map system and reference argument mentions to their entities using gold co-reference chains and calculate the standard precision and recall metrics over *entities*. For specific implementation details see Appendix C

5.3 Baselines

NP-SpanBERT (Elazar et al., 2022) is a classification model over NP pairs trained over TNE based on SpanBERT-Large (Joshi et al., 2020). We apply the label classifier on pairs of deverbal anchors and any other NP in the document and consider the phrase as an argument if the predicted label is any valid preposition.

QA-SRL Parser We re-train the generative parser from Klein et al. (2022) over a joint training set consisting of sentence level QA-SRL annotations for verbal and nominal predicates (FitzGerald et al., 2018; Klein et al., 2020) using a T5-Large encoder-decoder (Raffel et al., 2020). The parser is trained over examples of a sentence and marked predicate word as input and produces questions and answers in the QA-SRL format in its output, where each answer is a semantic argument. Our re-trained parser has significant performance boosts vs. previous published models on the QA-SRL data, for details refer to Appendix D. Training is performed for 5 epochs until convergence, using the Adam optimizer with a learning rate of $5e-05$ and a batch size of 16.

For the baseline, we simply apply the parser over complete passages during inference.

TNE-Parser Re-using the joint QA-SRL setup (Klein et al., 2022), we train a parser directly over passage-level TNE data over the deverbal subset of anchors. The parser takes a passage with the marked anchor (the head word) as the predicate and outputs questions and answers. Questions are encoded using the "[anchor] [preposition]?" template to signify the semantic relation between the pair, e.g. *"investigation by?"* and the answer is the complement-argument phrase of that relation.

Mistral We evaluate a prompting approach using the open-source Mistral-7B (v0.1) instruction-tuned model (Jiang et al., 2023). We design two different prompts for the task, each includes an instruction, a few examples (2-5) in the required format, and the passage with the predicate surrounded by special tags. The first prompt variant asks the model to produce a list of semantically related arguments of the marked predicate, while the second version asks for a combined representation of an argument and its semantic role represented as a natural language question-answer pair. Please refer to Appendix E for concrete prompt examples. For ON5V we use examples from the QASRL-GS development set (Roit et al., 2020) containing a high ratio of implicit arguments. For TNE, we use examples from the TNE training set, with questions formatted in the TNE-Parser format. The examples are randomly selected and kept fixed for the entire evaluation, to reduce the dependence on specific examples we repeat the evaluation four times and report the average and standard deviation. Decoding is performed with beam-search (beam=4).

5.4 Our System

NLI We apply our entailment-based approach using an off-the-shelf[2] NLI model (Laurer et al., 2024), based on DeBERTA-V3-Large and trained over a mixture of challenging NLI datasets (Parrish et al., 2021; Williams et al., 2018; Nie et al., 2020; Liu et al., 2022). Reported performance is on par with current leading models on MNLI and ANLI. All NLI-based models are tuned on the development set, or using cross-fold validation to find the best-performing classification threshold for candidate phrases.

Instruct-NLI We also apply our method with the Mistral LLM serving as the underlying entailment engine. We assume that the entailment task is embedded in different training regimes and datasets for instruction tuning, and apply the model in a "zero-shot" setting without demonstrating examples in the prompt. The specific prompt for NLI is re-used from FLAN (Wei et al., 2022), assuming a similar prompt was also used to train Mistral LLM as well. We ask for a binary Yes/No answer, where Yes refers to entailment, and verify

[2] https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

System	Training Data	Full Document			Cross-Sentence		
		Precision	Recall	F1	Precision	Recall	F1
Baselines							
NP-SpanBERT (LG)	TNE	75.33	42.86	54.63	66.46	36.60	47.20
TNE-Parser (T5-LG)	TNE	62.60	51.73	56.65	51.57	40.02	45.07
QA-SRL Parser (T5-LG)	QA-SRL	**84.77**	25.14	38.77	**79.85**	7.64	13.95
Mistral (Arg, 7B)	Instructions	35.62$_{\pm 7.3}$	52.93 $_{\pm 14.9}$	40.72$_{\pm 2.1}$	26.76 $_{\pm 6.2}$	48.81 $_{\pm 15.2}$	32.70$_{\pm 1.2}$
Mistral (QA, 7B)	Instructions	46.29$_{\pm 3.5}$	18.03$_{\pm 3.9}$	25.85$_{\pm 4.4}$	34.95 $_{\pm 3.7}$	15.50$_{\pm 3.0}$	21.41$_{\pm 3.3}$
Entailment-based models							
Instruct-NLI (Mistral 7B)	Instructions	49.22	53.55	51.29	36.01	41.49	38.56
NLI (DeBERTa LG)	NLI mix.	47.42	58.76	52.49	36.09	49.37	41.70
SRL-NLI (DeBERTa LG)	NLI mix. + QA-SRL	56.52	**60.29**	**58.34**	46.41	**50.43**	**48.34**

Table 1: Results on the TNE test set for argument detection. Metrics are entity-level — multiple mentions of the same entity are considered as one. "Full Document" refers to results evaluated on all of the arguments, while "Cross-Sentence" considers only those reference and predicted arguments that have their closest mention to the anchor predicate appear in a different sentence. Direct prompting methods (Mistral) results include standard deviation (SD) over 4 runs with different examples.

System	Precision	Recall	F1
Baselines			
QA-SRL Parser (T5-LG)	**58.33**	1.29	2.52
Mistral (Arg, 7B)	9.93$_{\pm 3}$	20.24$_{\pm 6.6}$	12.71$_{\pm 2.1}$
Mistral (QA, 7B)	7.04$_{\pm 1.4}$	11.41$_{\pm 1}$	8.60$_{\pm 0.9}$
Entailment-based models			
Instruct-NLI (Mistral 7B)	16.34$_{\pm 1.1}$	39.47$_{\pm 5.4}$	22.99$_{\pm 1}$
NLI (DeBERTa-LG)	16.90$_{\pm 2.4}$	**52.13**$_{\pm 3.3}$	25.47$_{\pm 3}$
SRL-NLI (DeBERTa-LG)	25.41$_{\pm 5.5}$	36.10$_{\pm 5.2}$	**29.28**$_{\pm 3}$

Table 2: Results on the ON5V unified evaluation set on *cross-sentence* arguments (see Appendix A for Full Document results). We evaluated only those reference and predicted arguments that their closest mention to the predicate appears in a different sentence. All NLI methods use cross-fold validation of 4 folds to determine the classification threshold and report mean and SD over the test folds. Direct prompting methods (Mistral) report mean and SD of 4 runs with different sets of examples.

that one of them is the first emitted token in the response. To get a normalized probability of entailment given the premise-hypothesis pair, we apply the softmax function over the corresponding logit values of "Yes" and "No" from the first decoded vector of logits and select the probability of "Yes".

SRL-NLI Training We fine-tune our predicate-argument-aware NLI model with the weights initialized to the aforementioned NLI model. Our model is trained for 3 epochs, with batch size 32 and 5e-6 learning rate.

Inference We extract NP candidates that do not overlap with existing local arguments, our candidate extraction is based on Spacy's noun-chunker (Honnibal et al., 2020), and successfully covers 80-90% of cross-sentence arguments in several benchmarks (Gerber and Chai, 2010; Moor et al., 2013). We set a strict threshold for local argument verification of 0.5 for the base NLI models and 0.95 for the semantics-aware model. If a local argument fails to be verified, it is assumed to be misplaced and is appended to the candidate list for further processing. All NLI-based methods use the QA-SRL Parser internally by parsing the predicate's sentence to extract local arguments.

6 Results

Tables 1 and 2 present the results of the argument detection task on nominal predicates from TNE and verbal predicates from ON5V, respectively. For TNE, we report the results in two settings, (1) *Full Document* considering all semantic arguments in the entire document and (2) *Cross-Sentence*, focusing on arguments located in different sentences than the predicate. This separation allows us to analyze the parsers' performance beyond sentence boundaries. For ON5V, we show results for the Cross-Sentence setting in Table 2 and defer Full Document results to Appendix A due to our focus on cross-sentence performance.

Across both datasets, our predicate-argument-aware entailment model (SRL-NLI), trained on a diverse mix of NLI datasets and further fine-tuned on QA-SRL-derived entailment data (§4), exhibits superior overall performance (F1) compared to all evaluated approaches.

Our generic approach outperforms supervised models on TNE As shown in Table 1, our distantly supervised SRL-NLI approach achieves superior performance compared to supervised models like NP-SpanBERT and TNE-Parser, even though these models were directly trained on TNE. This indicates the effectiveness of our approach in tackling semantic argument detection without the need for task-specific supervision.

Predicate-Argument-aware entailment model boost performance SRL-NLI outperforms NLI (using the same DeBERTa underlying model) by 6.6 F1 points on TNE and 3.8 on ON5V, indicating the benefit of an enhanced classifier that is sensitive to predicate-argument semantics.

Cross-sentence is more difficult When evaluated on TNE, all examined models undergo a performance deterioration for the more challenging setting of cross-sentence argument detection. The drop in performance is especially detrimental for the QA-SRL Parser (-24.8 F1), which can be attributed to its single-sentence training scope. Notably, NLI-based models exhibit an on-par performance decrease with the TNE parser, which was supervised over task-specific document-level data. Hence, it seems that our SRL-NLI approach enjoys the best of both worlds — it learns document-level semantic understanding from NLI, while specializing in predicate-argument semantics due to the sentence-level QA-SRL supervision.

LLMs: Simple wins, complex stumbles Directly asking Mistral in the few-shot setting to identify all semantic arguments of a predicate within a paragraph leads to subpar performance (40.72 vs. 58.34 F1 on TNE and 12.71 vs. 29.28 F1 on ON5V for the best Mistral configuration). Interestingly, prompting Mistral with arguments-only prompt consistently achieves higher performance than with QA prompt, on both TNE and ON5V.

However, our approach of framing implicit argument identification as a series of entailment decisions, and leveraging Mistral as a zero-shot entailment model (Instruct-NLI) already yields remarkable performance gains. This method surpasses directly prompting Mistral for arguments, achieving a 5.9 F1-score improvement on TNE and an impressive 10+ F1-score increase on ON5V.

These results highlight the benefit of decomposing complex tasks into simpler binary decisions for LLMs, potentially due to reduced reasoning burden and better alignment with their instruction fine-tuning data.

7 Analysis

Our evaluation against the TNE datasets measures unlabelled argument detection, which leaves the role assignment accuracy of our system unexplored. Since our approach is schema-independent, the argument's semantic role is not provided explicitly but is expressed through its syntactic position in the proposition. We thus tap into the *labeling accuracy* of our system through a manual analysis. Specifically, we sample 50 deverbal nominal predicates from the TNE test set along with their 260 gold cross-sentence complements and inspect the complements' highest-ranked proposition during inference. Each proposition contains the complement in its most probable syntactic position as ranked by our SRL-NLI model. In order to align the setting of our analysis to a typical use case scenario of our method, we further run an OntoNotes parser (Shi and Lin, 2019) over the selected propositions to attain PropBank labels of the arguments. An author of this paper then verified that the predicted semantic role label matches in definition against the semantic relation captured by TNE annotators.

Omitting 14 TNE complements that don't correspond to verbal arguments, and 20 arguments that are missed by the OntoNotes parser, the extracted role is accurate at 161/226 (71%) of the cases. Mistakes include both OntoNotes parsing mistakes, as well as erroneous syntactic positions selected by the NLI-based ranking.

8 Conclusions

We have demonstrated how to reformulate the problem of argument detection into an entailment task, and successfully used it to detect arguments across sentence boundaries where training data is inherently scarce. Moreover, we have explicated the meaning of these distant arguments in the form of simple and easy-to-grasp propositions that keep the correct semantic role information without committing to a specific label schema. Our proposed method can thus augment any specialized SRL or event-extraction schema with cross-sentence arguments at test time, without additional annotation or training. Given a sentence-level parser, one can apply it on the extracted proposition to get a label for the captured implicit argument. The propositions by themselves can potentially serve applications that require information decomposition into smaller units, e.g. SCUs (Nenkova and Passonneau, 2004) for the summarization task and many more.

Limitations

We raise the following limitations of our method. First, our method relies on a strong entailment component that is sensitive to the syntactic argument structure of the hypothesis and has a good comprehension of the passage. As we have discovered, this is not a trivial task even for contemporary NLI models.

Secondly, our method might be prone to correct but undesired entailment judgments. For example, when a passage describes several different events with lexically similar predicates (e.g. two acquisition events), we might construct a hypothesis with a participant of one event while targeting the other event where that participant does not belong. This problem is inherent to the entailment task. It is not event-specific, it verifies the hypothesis against the entire premise, without a notion of a target predicate. We tried to address this by incorporating the candidate phrase into a hypothesis with other local arguments of the event, yet this is not a foolproof method.

In addition, ideally, a non-entailed hypothesis would specify which argument phrase is incorrect or misplaced, however, we don't have that level of granularity in NLI, therefore we build the hypothesis from the ground up, first verifying the local arguments, and then adding candidates to different positions one at a time. There may be a better combinatorial approach that adds multiple candidates at different positions and verifies them together to save computational steps.

Delving deeper into computational costs, computing entailment for each candidate phrase multiple times may seem at first costly, however, we have seen in practice that our method is quick to run even on modest accelerators. Each classification decision applies a single forward pass in an encoder network, and the number of forward steps is bounded by the number of candidates we examine. On the other hand, a generative approach makes a forward pass at inference time for each *token* of a predicted argument. Moreover, simple implementation optimziation can mitigate most of the computational cost. In a typical example, we compute entailment over different short hypotheses against the same long passage. So pre-computing the attention key-values for that passage, as customarily done in Decoder-Only models, can effectively mitigate most of the required computation for a single example.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. 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, Sofia, Bulgaria. Association for Computational Linguistics.

Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2023. Chatgpt and simple linguistic inferences: Blind spots and blinds.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 355–366, New York, NY, USA. Association for Computing Machinery.

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.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077.

Yanai Elazar, Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2022. Text-based NP enrichment. *Transactions of the Association for Computational Linguistics*, 10:764–784.

Parvin Sadat Feizabadi and Sebastian Padó. 2015. Combining seemingly incompatible corpora for implicit semantic role labeling. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 40–50, Denver, Colorado. Association for Computational Linguistics.

Charles J. Fillmore. 1976. Frame semantics and the nature of language *. *Annals of the New York Academy of Sciences*, 280.

Charles J Fillmore. 1986. Pragmatically controlled zero anaphora. In *Annual Meeting of the Berkeley Linguistics Society*, volume 12, pages 95–107.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-scale QA-SRL parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2060, Melbourne, Australia. Association for Computational Linguistics.

Matthew Gerber and Joyce Chai. 2010. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden. Association for Computational Linguistics.

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

Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 512–520, Hong Kong. Association for Computational Linguistics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Ayal Klein, Eran Hirsch, Ron Eliav, Valentina Pyatkin, Avi Caciularu, and Ido Dagan. 2022. QASem parsing: Text-to-text modeling of QA-based semantics. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7742–7756, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2024. Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli. *Political Analysis*, 32(1):84–100.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Tatjana Moor, Michael Roth, and Anette Frank. 2013. Predicate-specific annotations for implicit role binding: Corpus annotation, data analysis and evaluation experiments. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 369–375, Potsdam, Germany. Association for Computational Linguistics.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does putting a linguist in the loop improve NLU data collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sameer S. Pradhan and Nianwen Xue. 2009. OntoNotes: The 90% solution. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*, pages 11–12, Boulder, Colorado. 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. *J. Mach. Learn. Res.*, 21(1).

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.

Michael Roth and Anette Frank. 2015. Inducing implicit arguments from comparable texts: A framework and its applications. *Computational Linguistics*, 41(4):625–664.

Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2010. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 45–50, Uppsala, Sweden. Association for Computational Linguistics.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. 2022. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2439–2455, Seattle, United States. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling.

Carina Silberer and Anette Frank. 2012. Casting implicit role linking as an anaphora resolution task. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 1–10.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485.

A ON5V Results

System	Precision	Recall	F1
Baselines			
QA-SRL Parser (T5-LG)	89.38	37.48	52.81
Mistral (Arg, 7B)	17.01$_{\pm4.9}$	21.45$_{\pm5.4}$	18.16$_{\pm1.7}$
Mistral (QA, 7B)	15.37$_{\pm3.6}$	15.49$_{\pm1.1}$	15.27$_{\pm1.9}$
Entailment-based models			
Instruct-NLI (Mistral 7B)	33.97$_{\pm1.5}$	61.41$_{\pm4.5}$	16.34$_{\pm1.1}$
NLI (DeBERTa-LG)	31.28$_{\pm1.6}$	69.01$_{\pm2.9}$	43.03$_{\pm1.9}$
SRL-NLI (DeBERTa-LG)	46.59$_{\pm7.9}$	61.49$_{\pm2.4}$	52.64$_{\pm4.1}$

Table 3: Results on the ON5V unified evaluation set on *full-document* evaluation. All NLI methods use cross-fold validation of 4 folds to determine the classification threshold and report mean and std. dev. over the test folds. Direct prompting methods report an average and std. dev. of 4 runs with different sets of examples.

For completeness, we add the results for the full document evaluation on ON5V. We achieve comparable results to the QA-SRL parser on the full document. The parser does not extract almost any cross-sentence arguments, and its overall results stem from its high in-sentence performance.

B ON5V Annotation

We annotated additional arguments for the ON5V dataset for the existing predicates in the dataset. Annotators were instructed to add new argument phrases and write a question for each one using the QA-SRL question format. Our interface, depicted in Figure 3, presents the full document with the predicate and all of the already marked arguments from OntoNotes (Pradhan and Xue, 2009) and ON5V, and a selection of candidate phrases. Annotators were instructed to add new mentions and do not modify existing arguments. In our experience selecting arguments from a wide candidate list, as also performed in TNE (Elazar et al., 2022), streamlines annotation on a long passage and helps the annotator in covering lengthy contexts.

We scoped the annotation to be in a context window of sentences of 5 preceding sentences and 1 subsequent after the predicate. Past works have shown that more than 90% of all implicit arguments can be found within this window (Gerber and Chai, 2010). Our phrase candidates include noun-phrases extracted using the same procedure we describe in section 5, and the annotator is asked to remove them from a "TODO" list if they are not an argument, or write them a proper QA-SRL question. If a candidate is co-referring to a current argument, we ask the annotator to add it to its set of answers. Otherwise, we ask them to add it as a new QA pair, even if the question repeats itself.

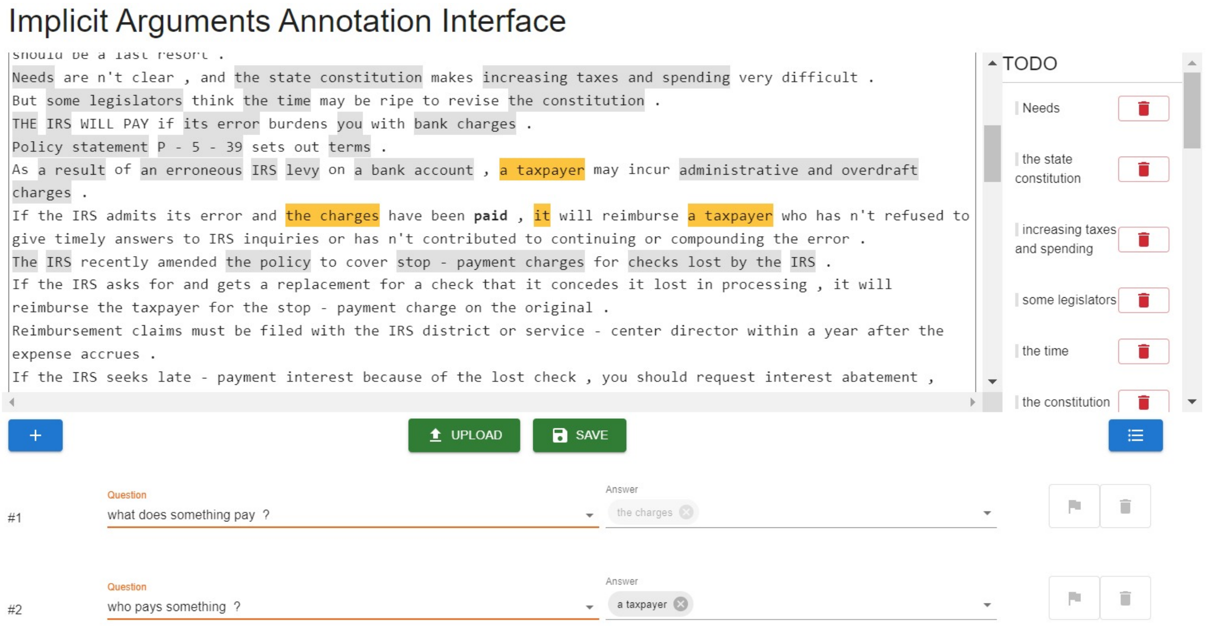
System	Dataset	Precision	Recall	F1
T5-Large, retrained	Verbal	91.36	64.27	75.46
T5-Large, retrained	Nominal	76.16	63.73	69.39
T5-Large, retrained	ON5V	76.48	84.35	80.22
T5-Small (Klein et al., 2022)	Verbal	76.20	62.40	68.60
T5-Small (Klein et al., 2022)	Nominal	64.30	54.80	59.20

Table 4: Results for single sentence evaluation of the retrained parser on QA-SRL and ON5V evaluation sets.

We recruited 5 in-house annotators, four with a strong background in linguistics and one native English speaker who excelled on our qualification assignment. We presented them the QA-SRL annotation guidelines from Roit et al. (2020), and conducted a short training round of 10-15 predicates, after which we provided personal and detailed feedback. Each predicate took on average 5 minutes to annotate. During the annotation period, one of the authors examined 10-20% of each annotator's workload to verify correctness and proper coverage. We paid each annotator an hourly rate of 14$, and annotation took about 10 minutes per predicate.

C Evaluation Procedure

An argument is mapped to the entity of the highest-ranking mention in any chain according to an overlap score, as long as it passes a certain threshold. Scoring between two phrases is calculated based on syntactical head equivalence[3], which accounts for a full match, or the token-wise intersection-over-union (IOU) which ranges between 0 and 1. The threshold for a match is set to 0.5 based on standard argument evaluation criteria (Roit et al., 2020). For each evaluated predicate, we add arguments that did not map to any existing coreference cluster to singleton clusters. We verify that a predicted argument that matches some gold argument according to the criteria above, will always be mapped to the same entity with the matching gold argument.



Implicit Arguments Annotation Interface

should be a last resort .
Needs are n't clear , and the state constitution makes increasing taxes and spending very difficult .
But some legislators think the time may be ripe to revise the constitution .
THE IRS WILL PAY if its error burdens you with bank charges .
Policy statement P - 5 - 39 sets out terms .
As a result of an erroneous IRS levy on a bank account , a taxpayer may incur administrative and overdraft charges .
If the IRS admits its error and the charges have been **paid** , it will reimburse a taxpayer who has n't refused to give timely answers to IRS inquiries or has n't contributed to continuing or compounding the error .
The IRS recently amended the policy to cover stop - payment charges for checks lost by the IRS .
If the IRS asks for and gets a replacement for a check that it concedes it lost in processing , it will reimburse the taxpayer for the stop - payment charge on the original .
Reimbursement claims must be filed with the IRS district or service - center director within a year after the expense accrues .
If the IRS seeks late - payment interest because of the lost check , you should request interest abatement ,

TODO
Needs
the state constitution
increasing taxes and spending
some legislators
the time
the constitution

#1 Question: what does something pay ? Answer: the charges

#2 Question: who pays something ? Answer: a taxpayer

Figure 3: Our implicit arguments annotation interface. The yellow highlighted phrases depicts the current set of arguments, phrases in grey are candidates that need to be either removed from the TODO list or selected as an answer to a QA-SRL question. The interface validates that the question is formatted correctly.

D QA-SRL Parser Evaluation

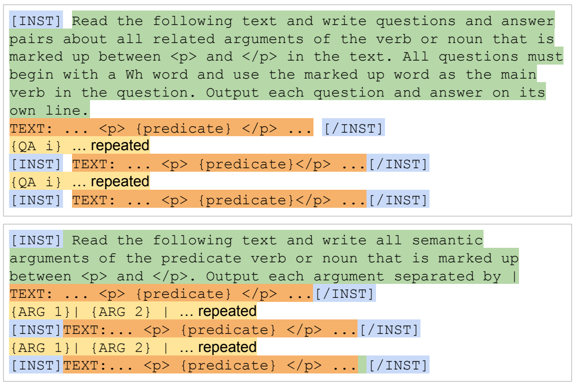
We re-train the joint QA-SRL parser (Klein et al., 2022) on a T5-Large model and report performance metrics on single sentences. Evaluation is conducted with unlabeled mention-level metrics that match spans between reference and predicted arguments. Results are shown in Table 4. Verbal and Nominal refer to the gold-standard evaluation sets of Roit et al. (2020) and Klein et al. (2020) respectively. A span match threshold of IOU >= 0.3 was used to match previously published metrics.

E Prompt Examples

We provide the prompt templates for both the QA prompt and the argument prompt formatted specifically as a chat for the Mistral model in Figure 4.

F Implementation Details

The prepositions for `Indirect` or `Adjunct` phrases can be pre-assigned, as in the case of local arguments, or can be inferred using an auxiliary model. To assign a preposition to an argument sourced from the QA-SRL analysis, we inspect the dependency structure of either the original sentence or the declarative sentence that is formed by its QA pair. If the argument in one of these sentences is connected to the predicate via a preposition, we re-use it in our semantic hypothesis as well. For example, consider the "on the day of the bombing" argument in Figure 2, it connects with the predicate "leave" via the preposition "on" in both sentences. In the general case, we use a masked language model (Devlin et al., 2019) to output the highest-ranking *preposition* word given the context and the masked hypothesis. We use bert-large-cased as the underlying ranking model.



```
[INST] Read the following text and write questions and answer
pairs about all related arguments of the verb or noun that is
marked up between <p> and </p> in the text. All questions must
begin with a Wh word and use the marked up word as the main
verb in the question. Output each question and answer on its
own line.
TEXT: ... <p> {predicate} </p> ... [/INST]
{QA i} ... repeated
[INST] TEXT: ... <p> {predicate}</p> ...[/INST]
{QA i} ... repeated
[INST] TEXT: ... <p> {predicate}</p> ...[/INST]
```

```
[INST] Read the following text and write all semantic
arguments of the predicate verb or noun that is marked up
between <p> and </p>. Output each argument separated by |
TEXT: ... <p> {predicate} </p> ...[/INST]
{ARG 1}| {ARG 2} | ... repeated
[INST]TEXT:... <p> {predicate} </p> ...[/INST]
{ARG 1}| {ARG 2} | ... repeated
[INST]TEXT:... <p> {predicate} </p> ... [/INST]
```

Figure 4: The Mistral specific prompts formatted both as QA generation (top) and argument extraction (bottom). Blue highlighting is to indicate chat instructions, green is our task specific instruction, orange is for the query, yellow is our example of a suitable response.

We also inspect the local QA-SRL analysis for

[3] The head of a phrase is captured by Spacy's dependency parser (Honnibal et al., 2020)

linguistic attributes such as tense, modality, and negation that affect the main verb inflection and auxiliaries. For example, the question "Who did not pay something" uncovers that the event is described in the past tense and that the event did not unfold in the sentence. We parse these attributes from the QA-SRL question of the first argument in the template and initialize our hypothesis accordingly. When required, we modify the `VERB` field of the template to include the modal verb ‘might’ or negate the auxiliary verb and use the proper inflection.