Explicating the Implicit: Argument Detection Beyond Sentence Boundaries

Paul Roit[1] **Aviv Slobodkin**[1] **Eran Hirsch**[1] **Arie Cattan**[1]
Ayal Klein[1] **Valentina Pyatkin**[2,3] **Ido Dagan**[1]
[1]Bar-Ilan University [2]Allen Institute for Artificial Intelligence
[3]University of Washington
plroit@gmail.com

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.

“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 other sentences in the document. (3) A proposition with some arguments (the house) placed in an incorrect syntactic position that does not align with its original semantic role. (4) A proposition with an incorrect argument according to the document. Both (3) and (4) are not supported by the document.

1 Introduction

Identifying which entities in a text play specific semantic roles with respect to a predicate word 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 than a single sentence. Consider the example in Figure 1. The location that the boat left, ‘*the house*’, and the destination where it was headed to, ‘*the port*’, can be deduced by associating the trip with the leave event in the preceding sentence. Such examples showcase where our semantic interpretation 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.

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

[1]Our codebase, dataset, and models can be found at https://github.com/plroit/semquest

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16394–16409
August 11-16, 2024

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 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 (Michael, 2023) 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 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.

The cross-sentence task has been notably underexplored in the literature, largely due to the extreme difficulties in constructing suitable datasets (Gerber and Chai, 2010; Moor et al., 2013; Ruppenhofer et al., 2010). Under this context, we suggest a significantly more feasible approach that leverages only existing resources designed for in-sentence argument detection to detect semantic arguments across sentence boundaries. Our distantly supervised method achieves higher performance than some fully supervised models on a document-level dataset (Elazar et al., 2022) for noun-phrase relations, and outperforms other zero and few-shot 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 (Fillmore, 1986; Gerber and Chai, 2010; Klein et al., 2020; Roit et al., 2020; Pyatkin et al., 2021). 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). And Pyatkin et al. (2020) used QA pairs to represent implicit and explicit discourse relations, also across sentences (Pyatkin et al., 2023).

TNE is a dataset for modeling semantic relations between noun phrases (NPs) across a document and is annotated on top of Wikinews. A relation consists of an anchor and complement phrases that are labeled with a preposition, i.e. [the investigation]$_{\text{ANCHOR}}$ *by* [the police]$_{\text{COMPLEMENT}}$. 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 derived from news articles in the development and train partitions of OntoNotes (Pradhan and Xue, 2009). The dataset contains 390 instances selected from 260 documents, and annotates *five* different verbal predicates. The original annotators have inspected only core roles (i.e. numbered: ARG0, ARG1, etc.) that were missing an explicit filler argument in OntoNotes, and filled the role with the closest phrase to the predicate that fit the role description. We have re-annotated this dataset to close the coverage gap for modifier roles and retrieve all fitting phrases (see §6.1).

3 Task Definition

Given a text document $\mathbf{D} = [\mathbf{s}_1, \ldots, \mathbf{s}_n]$, consisting of n sentences, and a target predicate word p from the document (a verb or deverbal noun), our task is to detect the set of semantic arguments $\mathcal{A}_\mathrm{p} = \{a_i\}$ of the predicate. In this work, we adopt the QA-SRL definition of a semantic argument and apply it to the document-level task. An argument of p is any phrase a in the document that pertains to the event referred by p and correctly answers a simple Wh-question revolving around the predicate word. For example, *What did something leave?* asks about *the house* argument in Figure 1.

We will mark the sentence where the predicate resides as $\mathbf{s}_p$ and define the set of local, in-sentence arguments as $\mathcal{L}_\mathrm{p} = \{a_i | a_i \in \mathcal{A}_\mathrm{p} \wedge a_i \subset \mathbf{s}_p\}$, where the subset notation is overloaded here to represent a sub-sequence. While past works have focused on detecting $\mathcal{L}_\mathrm{p}$, we will show a method that leverages $\mathcal{L}_\mathrm{p}$ to detect the rest of cross-sentence arguments.

4 Method

Our approach is based on synthesizing a simple sentence that encodes candidate phrases as arguments for the predicate p, and testing if the sentence is entailed from the document. The synthesized sentence $\mathcal{H}$, named the *semantic hypothesis*, is created from templates by placing different phrases in subject or object positions around the predicate verb.[2] For example, in proposition no. 2 in Figure 1 *'The boat'* is assigned to the subject position and represents the LEAVER role, and *'on the day of the bombing'* is assigned to an adjunct and represents a TEMPORAL role. If the semantic hypothesis is entailed from the passage, we conclude that the predicate-argument relations encoded within $\mathcal{H}$ are manifested in the document. On the other hand, if the hypothesis is not entailed, we can conclude that the document does not express one or more of the encoded argument relations. For example, proposition no. 4 incorrectly assigns *'the investigators'* to the LEAVER role, which should not be entailed from the passage.

Our method iterates over candidate phrases extracted from across the document (Figure 2, left). To determine if a candidate phrase c is a semantic argument of the predicate, we construct several hypotheses that incorporate the candidate and

[2]If p is a nominalization, its verbal form will be used instead.

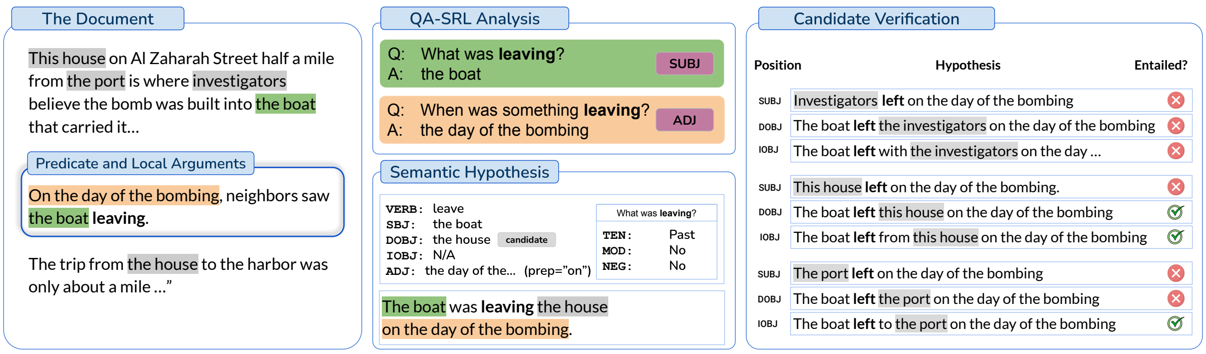

Figure 2: *Left:* An excerpt from the document with the predicate marked in bold in the circled sentence. Candidate phrases from across the document are highlighted in gray. For completeness, we also highlight local arguments and their co-referent mentions in color. *Center-Top:* A QA-SRL parser analyzes the predicate's sentence and outputs local arguments, their questions, and their syntactic position (in purple). *Center-Bottom:* Hypothesis fields are assigned with argument and candidate phrases to different syntactic positions. Grammatical attributes are extracted from the question of the first local argument. The generated hypothesis sentence is shown in the box below. *Right:* Each candidate (highlighted in gray) is inserted into three different position fields and the resulting hypothesis is verified with an NLI model against the full document. The second candidate demonstrates two correct alternations.

check for entailment against the document (Figure 2, right). The hypotheses combine local, in-sentence, arguments of the predicate with the candidate phrase, and differ in the syntactic position that the candidate assumes. Conceptually, each position encodes the candidate into a different semantic role. Moreover, the hypothesis sentence associates the candidate with the specific target event by incorporating local arguments of the event, the remote candidate, and the target predicate into a single sentence. If $\mathcal{H}$ is entailed, it suggests that the candidate participates in the event in the document, and that the candidate's semantic role in the document is the same as its role in $\mathcal{H}$.

Our synthetic hypothesis sentence is generated from syntactic templates. They accommodate the main verb (VERB) with up to 4 argument phrases in the following unique syntactic positions: subject (SUBJ), direct object (DOBJ), indirect object (IOBJ), and adjunct (ADJ). It can be modified using the following grammatical attributes: tense (TEN), negation (NEG), and modality (MOD). These fields are inspired by the QA-SRL question syntax that was similarly used to generate simple propositions in Pyatkin et al. (2021). They present high versatility and can capture any semantic role of a verbal predicate in English, yet maintain a compact representation requiring only a few assignments.

In this work, unless specified otherwise, we consider as candidates noun phrases and named entities from the document that do not overlap with local arguments, including those from $\mathbf{s}_p$. Each candidate is placed into different hypotheses in the subject, direct and indirect object positions. We use the probability of entailment as predicted by an NLI model as the score of the hypothesis $\mathrm{NLI}(\mathbf{D}, \mathcal{H})$ and define the score of each candidate to be

$$\max_{\mathrm{pos} \in \{\mathtt{S},\mathtt{D},\mathtt{I}\}} \mathrm{NLI}\left(\mathbf{D}, \mathcal{H}[\mathrm{pos} = c]\right)$$

the maximal score of any of its hypotheses. Our final predicted set of arguments includes all candidate phrases that scored above a configurable threshold and all local arguments predicted by the parser. In the next subsections, we will describe how we populate the hypothesis syntactic positions and attributes given the predicate and its sentence (§4.1), and how we then generate the actual hypothesis sentence (§4.2).

4.1 Populating Hypothesis Fields

We initialize the hypothesis fields with the predicate as the main verb and assign its local arguments to syntactic positions. The local arguments are extracted from the predicate's sentence along with their question labels using a high-performing QA-SRL parser (Klein et al., 2022) (see Figure 2, center-top). The syntactic position of each argument is determined by applying a heuristic from Klein et al. (2020) that inspects the question and maps it to one of the argument positions (Figure 2, top-left). Generally, each local argument is assigned to a field according to its syntactic position (Figure 2, center-bottom). However, to mitigate QA-SRL parsing errors and resolve collisions, we score each local argument, and select the top argument per position. To score a local argument, we generate a singleton hypothesis containing only the argument, and compute its entailment probability against the predicate's sentence using an NLI model. Refer to §G for more details.

Tense, modality, and negation attributes are determined by inspecting the question of the first local argument in the hypothesis. For example, *'Who might have left?'* indicates that the event should be described in the past tense with the modal verb 'might' (see §G).

Finally, given a candidate phrase, we generate different hypotheses, assigning the candidate to a different syntactic position in each, possibly overriding a local argument.

4.2 Generating the Semantic Hypothesis

Given an assignment to the hypothesis fields, we construct the hypothesis sentence by filling a syntactic template and using English grammar rules for inflection. If a subject phrase is specified, we use the active voice template. Otherwise, we resort to the passive voice, placing the direct object in the subject position as follows:

$$\text{Active} \implies \texttt{SUBJ-AUX-VERB-DOBJ-IOBJ-ADJ}$$
$$\text{Passive} \implies \texttt{DOBJ-AUX-VERB}_{\texttt{passive}}\texttt{-IOBJ-ADJ}$$

The corresponding auxiliary (AUX) and main verbs are automatically assigned or inflected[3] based on the grammatical attributes of the hypothesis, the active or passive voice, and after considering the subject-verb agreement.

Notably, a valid declarative sentence in English must contain a subject phrase, while for some instances there is no apparent local subject or object argument. E.g. '*the 2 PM* ***presentation***' implies that someone is presenting something at 2 PM. To write a valid hypothesis, we insert abstract placeholder arguments instead of concrete ones, placing 'someone' in an empty subject position or 'something' in empty object positions when necessary (see §G for details).

The prepositions for indirect objects or adjuncts are assigned by either syntactic analysis or predicted with a masked language model. For local arguments, we look for a connecting preposition between the predicate and the argument by inspecting the original sentence or the declarative form of the QA pair. Otherwise, we use a masked language model (Devlin et al., 2019) to rank prepositions given the full passage and the hypothesis. For more details see Appendix G.

[3] We use https://github.com/bjascob/pyinflect

5 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.

6 Experiment Setup

6.1 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 (§2). 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. The average document in TNE is 8 sentences long, where each deverbal anchor is related to 4.5 complement entities on average, 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.

The TNE task pre-specifies a list of noun-phrases as candidates in each document[4], and we are tasked to select out of those the complement phrases for each deverbal anchor in the document.

When using generative methods to form arguments, we consider a specific NP candidate as predicted if it matches one of the generated argument phrases. 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, using the unified set of predicates from both train and development partitions as our evaluation data. The documents in this dataset have gold coreference annotations, which are necessary for our evaluation protocol (see §6.2). We use cross-fold validation over 4 folds and report average and standard deviation on the test fold. The search for arguments is limited to a context window of 7 sentences, with 5 preceding and 1 subsequent sentence around the predicate, a scope that was found to be sufficient to locate more than 98% of all originally annotated arguments. We use noun phrases and named entities as candidates for cross-sentence arguments, they are extracted using Spacy's (Honnibal et al., 2020) NER and dependency parsers. These candidate phrases cover 80% to 90% of all cross-sentence arguments found in Moor et al. (2013) and Gerber and Chai (2010) respectively.

To close the coverage gap (see section 2) 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 C for more details regarding the annotation protocol.

6.2 Evaluation

We follow the evaluation methodology proposed by Ruppenhofer et al. (2010) and adapt it to our schema-free task setting. It states that credit for a semantic argument should be assigned only once, irrespective of the number of multiple mentions it has in the reference or system output. Since a salient argument can dominate the argument set with multiple mentions, it becomes imperative to disentangle the evaluation of argument detection from co-reference resolution.

Specifically, our evaluation procedure employs externally provided co-reference data and maps predicted and reference arguments to their entity clusters.[5] An entity is considered as predicted if it has at least one mention in the system output, and likewise for an entity in the reference set. We calculate the standard precision and recall metrics over entities, summing entity counts over all instances in the dataset.

A predicted or reference argument mention is mapped to a co-reference cluster if its match score is above 0.5 with one of the mentions in the cluster. The score between two spans in the document is defined as $\mathrm{score}(a, m) = \max\{\mathbb{I}[\mathrm{h}(a) == \mathrm{h}(m)], \mathrm{IOU}(a, m)\}$ where $\mathrm{h}(a)$ is the index of the syntactic head token[6] of a and IOU is the tokenwise intersection over union score. If an argument is matched to multiple clusters, we select the one with the higher match score. Arguments that do not correspond to any pre-existing cluster form a new singleton

[4]Note that in an ordinary SRL setup, a candidate list is not provided.

[5]Our evaluation benchmarks provide gold co-reference chains.

[6]Head indices are retrieved with the spaCy dependency parser (Honnibal et al., 2020)

cluster.[7] A pseudo-code of the protocol is provided in §6.2

This evaluation procedure measures unlabeled argument detection. Evaluating labeled accuracy is challenging since our method does not produce explicit labels, but rather provides some signal about the semantic role through the generated hypothesis. Moreover, our evaluation benchmarks use different label sets that cannot be mapped easily. Instead, we suggest an analysis in section 8 that provides a measure of label accuracy.

6.3 Baselines

NP-SpanBERT (Elazar et al., 2022) is a classification model over anchor-complement pairs that was trained directly over TNE, and based on SpanBERT-Large (Joshi et al., 2020). We apply the 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 a 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 E. 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.

[7]The evaluation procedure maps matching system and reference arguments to the same singleton cluster where necessary.

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 (**Arg**) asks the model to produce a list of semantically related arguments of the marked predicate, while the second variant (**QA**) asks for a combined representation of an argument and its semantic role represented as a natural language question-answer pair (refer to Appendix F 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 width is set to 4.

6.4 Our Method

Our entailment-based approach is applied using fine-tuned or zero-shot NLI models. All models are tuned on the development set for TNE, or using cross-fold validation for ON5V, to find the best-performing classification threshold for candidate phrases. We use the same NLI model for both phases of local argument and non-local candidate verification. Noteworthy, the premise in the non-local case is significantly longer, but is limited to the scope of the search, which is 7-8 sentences on average. This scope is within the reach of NLI models trained on contemporary datasets as evident in several related works (Honovich et al., 2022; Schuster et al., 2022).

NLI We use an off-the-shelf NLI model[8] based on DeBERTA-V3-Large (He et al., 2021; Laurer et al., 2024) 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.

SRL-NLI We fine-tune our NLI model using the predicate-argument aware dataset (section 5) with the same architecture as the aforementioned

[8]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)	$16.90_{\pm 2.4}$	$\mathbf{52.13}_{\pm 3.3}$	$25.47_{\pm 3}$
SRL-NLI (DeBERTa-LG)	$25.41_{\pm 5.5}$	$36.10_{\pm 5.2}$	$\mathbf{29.28}_{\pm 3}$

Table 2: Results on the ON5V unified evaluation set on *cross-sentence* arguments (see Appendix B 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.

NLI model and initialized to the same weights. Our model is trained for 3 epochs, with batch size 32 and 5e-6 learning rate.

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 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".

7 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 B 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 (§5), 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 just for the arguments (**ARG**) consistently achieves higher performance on both TNE and ON5V, than asking it to produce arguments in the form of QA pairs (**QA**), which could have been more fitting for an instruction tuned model.

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.

8 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. For further analysis refer to Appendix D.

9 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 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 augment any 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 to the extracted proposition to get a label for the captured argument. The propositions 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

First, our method relies on a robust entailment engine that is sensitive to the syntactic argument structure of the hypothesis and has a strong 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 that is correctly entailed due to one event, but incorrect with respect to the target event. This problem is inherent to the entailment task. Entailment is not conditioned on a specific event in the premise but rather verifies the hypothesis against all the information in the premise text. We tried to address this issue by incorporating the candidate phrase into a hypothesis with other local arguments of the target event, yet this is not a foolproof method.

Lastly, this method may seem computationally intensive, as every candidate phrase in the document is used for entailment multiple times. 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. In contrast, a generative approach makes a forward pass at inference time for each *token* of a predicted argument.

Acknowledgements

We would like to thank our diligent reviewers for their constructive suggestions and comments. This work was supported by the Israel Science Foundation (grant no. 2827/21), and a grant from the Israel Ministry of Science and Technology.

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.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

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.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

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.

Julian Michael. 2023. The case for scalable, data-driven theory: A paradigm for scientific progress in NLP. In *Proceedings of the Big Picture Workshop*, pages 40–52, Singapore, Singapore. 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.

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.

Valentina Pyatkin, Paul Roit, Julian Michael, Yoav Goldberg, Reut Tsarfaty, and Ido Dagan. 2021. Asking it all: Generating contextualized questions for any semantic role. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1429–1441, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Valentina Pyatkin, Frances Yung, Merel C. J. Scholman, Reut Tsarfaty, Ido Dagan, and Vera Demberg. 2023. Design choices for crowdsourcing implicit discourse relations: Revealing the biases introduced by task design. *Transactions of the Association for Computational Linguistics*, 11:1014–1032.

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.

Tal Schuster, Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, and Donald Metzler. 2022. Stretching sentence-pair NLI models to reason over long documents and clusters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 394–412, Abu Dhabi, United Arab Emirates. 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 Evaluation Procedure

The following excerpt is an example of a partial Python implementation of our evaluation procedure described in §6.2. It counts the number of true positive, false positive, and false negative entities in a predicted instance.

```
def count_entities(
  refs, preds,
  all_mentions
) -> tuple[int,int,int]:
  for ref in refs:
    all_mentions = map_cluster(
      ref, all_mentions
    )
  for pred in preds:
    all_mentions = map_cluster(
      pred, all_mentions
    )
  pred_ccs = {a['cc_id'] for a in preds}
  ref_ccs = {a['cc_id'] for a in refs}
  n_tp = len(pred_cs & ref_ccs)
  n_fp = len(pred_ccs - ref_csc)
  n_fn = len(ref_ccs - pred_ccs)
  return n_tp, n_fp, n_fn

THRESH = 0.5
def map_cluster(arg, all_mentions):
  score, idx = best_match(
    arg, all_mentions
  )
  if score >= THRESH:
    cc_id = all_mentions[idx]['cc_id']
    arg['cc_id'] = cc_id
  else:
    cc_id = new_cc_id(all_mentions)
    arg['cc_id'] = cc_id
    all_mentions.append(arg)
  return all_mentions

def best_match(arg, all_mentions):
  scores = [
    score(ref, m) for m in all_mentions
  ]
  score, idx = argmax(scores)
  return score, idx
```

The input is a list of predicted and reference arguments, and a list of known co-reference mentions. Each argument and mention item have a start and end token index and also a head token index. The mentions also have an entity identifier (cc_id).

B ON5V Results

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.

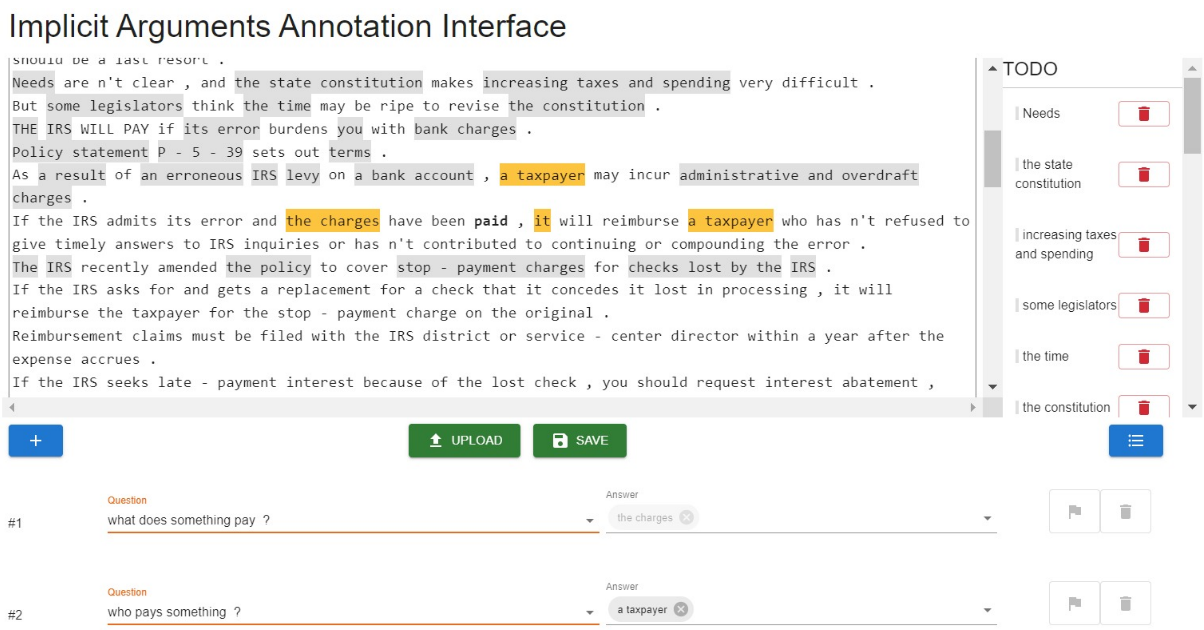

Figure 3: Our implicit arguments annotation interface. The yellow highlighted phrases depicts the current set of arguments, phrases in grey are candidates that need to be either removed from the TODO list or selected as an answer to a QA-SRL question. The interface validates that the question is formatted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

System	Precision	Recall	F1
Baselines			
QA-SRL Parser (T5-LG)	89.38	37.48	52.81
Mistral (Arg, 7B)	$17.01_{\pm 4.9}$	$21.45_{\pm 5.4}$	$18.16_{\pm 1.7}$
Mistral (QA, 7B)	$15.37_{\pm 3.6}$	$15.49_{\pm 1.1}$	$15.27_{\pm 1.9}$
Entailment-based models			
Instruct-NLI (Mistral 7B)	$33.97_{\pm 1.5}$	$61.41_{\pm 4.5}$	$16.34_{\pm 1.1}$
NLI (DeBERTa-LG)	$31.28_{\pm 1.6}$	$69.01_{\pm 2.9}$	$43.03_{\pm 1.9}$
SRL-NLI (DeBERTa-LG)	$46.59_{\pm 7.9}$	$61.49_{\pm 2.4}$	$52.64_{\pm 4.1}$

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.

C 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 not to 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 into 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 6, 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.

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.

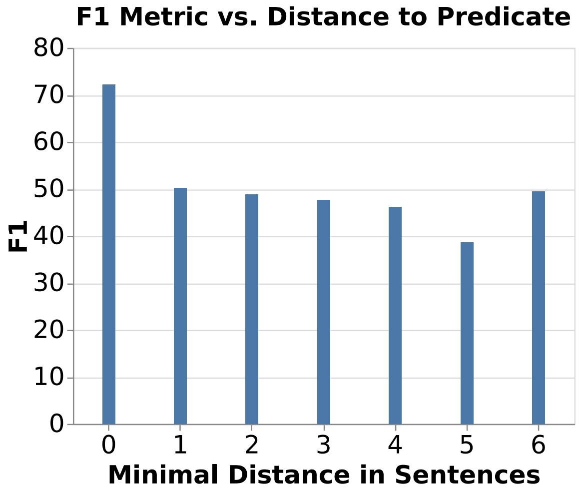

Figure 4: Stratification of our results on the test-set of TNE according to distance counted in sentence between the entity and the predicate. The distance of an entity from the predicate is defined as the absolute difference between the sentence index of the closest mention to the predicate and the sentence index of the predicate.

D Extended Analysis

To further investigate the performance of our method, we present in Figure 4 a stratified view of our TNE test set predictions according to distances from the predicate. Since our evaluation protocol counts entities (clusters of argument mentions), to stratify the test set we first group arguments into entities, compute the minimal entity distance to the predicate, and divide accordingly. Specifically, given a target distance, we select a subset of all predicted and reference arguments whose minimal entity distance to the predicate equals the target value. We count distances in sentences, where the minimal distance of an entity to the predicate is defined as follows:

$$\mathrm{MinDistance}(\mathbf{ent}, \mathbf{p}) = \min_{m \in \mathrm{ent}} |\mathrm{sent}(m) - \mathrm{sent}(\mathbf{p})|$$

where $\mathrm{sent}(a)$ returns the sentence index of the mention. All presented strata in the figure have at least 100 gold entities in the evaluated subset.

The results for in-sentence arguments (first column) can mostly be attributed to the QA-SRL parser. We observe a slight degradation in performance as the distance of the entity from the predicate increases.

E QA-SRL Parser Evaluation

We re-train the joint QA-SRL parser (Klein et al., 2022) on a T5-Large model and report performance

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.

```
[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 5: The Mistral-specific prompts are formatted both as QA generation (top) and argument extraction (bottom). Blue highlighting indicates chat instructions, green is our task-specific instruction, orange is for the query, and yellow is our example of a suitable response.

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.

F Prompt Examples

We provide prompt templates for both the QA prompt and the argument prompt formatted specifically as a chat for the Mistral model in Figure 5.

G Implementation Details

Preposition Reranker We use a masked language model `bert-large-cased` (Devlin et al., 2019) to select a preposition for a prepositional phrase (`ADJ` or `IOBJ` slots) in the hypothesis. The input to the language model is a concatenation of the document $\mathbf{D}$ with the hypothesis $\mathcal{H}$. The hypothesis places `[MASK]` tokens to indicate the masked preposition token. We call the language model and extract the output distribution over the vocabulary for the masked token. Then, we select the highest-ranked preposition by probability out of the following list if it appears in the top 10 ranked words in the vocabulary. The prepositions include: *on, at, for, to, from, about, as, against, in, with, off, over, into, after, while, before* and *by*. Otherwise, we do not prepend a preposition to the phrase.

Grammatical Attributes The question in QA-SRL can encode different attributes of the event, such as its tense, modality, and negation. We parse the question of the first local argument in each hypothesis to copy over those attributes from the question to the hypothesis itself. To determine negation we look for a '*not*' or '*n't*' in the question. Similarly, to determine modality, we look for the presence of one of the following modal verbs in the auxiliary field of the question: 'may', 'should', 'would', 'can', or 'might'. We implement a heuristic that uses the inflection of the auxiliary and main verb in the question to determine the tense.

Local Argument Verificaiton All NLI-based methods use the QA-SRL Parser internally to extract local arguments (see §4.1). Some instances are prone to parsing errors, specifically, question-labels tend to have higher error rates. This negatively affects our process which is sensitive to the syntactic position of each argument sourced from the question. In other cases, more than one local argument may share the same syntactic position.

These problems can be remedied by filtering incorrect arguments and selecting the top-ranked argument based on its entailment score. We determine the argument's score using the following procedure. We build *singleton* hypotheses per argument that assign the phrase to a syntactic position according to its question-label. We also put placeholder arguments ('someone', 'something', see subsection 4.2) on empty `SUBJ` and `DOBJ` fields, creating hypotheses with an intransitive and transitive usage. We score the hypotheses using our entailment model with the predicate's sentence as premise, and assign the argument the highest NLI score one of its hypotheses receives.

It is assumed that if a local argument does not pass a strict NLI threshold then, most likely, it is due to an error associated with its syntactic position (we refer to the high *unlabeled* precision for the parser at Appendix E). Hence, the failing argument phrase is appended to the candidate list for further processing and placement in a different syntactic position. We set a strict threshold for local argument verification of 0.5 for the `NLI` and `Instruct-NLI` models and 0.95 for the semantics-aware model `SRL-NLI`. These were determined empirically on a sample of the QA-SRL development set.

Transitive and Intransitive usage The valence pattern of a predicate within a sentence can change the semantic interpretation of its arguments. Consider the differences between: '*Salaries* **increased** across the sector' and '*The board* **increased** the salaries across the sector'. The subject phrase is interpreted differently given the valence of the predicate.

The valence pattern in our synthetic hypotheses is determined by the presence or absence of the `DOBJ` argument. In some cases, the `DOBJ` slot is left unassigned for various reasons, however, its presence is necessary to interpret the predicate in the correct sense according to the document. In such cases, we add another hypothesis assigning the abstract placeholder 'something' to `DOBJ` and score it according to the usual flow of our method.