# SDA: SEPARATING DISCOVERY FROM ANNOTATION FOR LLM-BASED STRUCTURED EXTRACTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Structured extraction organizes semantic elements and their relations in free-form text into predefined records. Large language models (LLMs) support this task through in-context learning (ICL), yet often fall short of task-specific supervised models. Prior work addresses this gap through better task instructions and demonstration selection. Our analysis shows that demonstration patterns change predicted record counts for unchanged inputs, suggesting that examples may constrain the model's interpretation of source content. Inspired by this finding, we propose SDA, a structured extraction framework that separates source-based semantic discovery from demonstration-guided annotation. Specifically, the framework comprises three modules. Semantic Discovery constructs natural-language candidates through complementary source readings without labeled demonstrations. Candidate Projection uses relevant precedents from an annotation memory to normalize candidates and delimit admissible annotation choices. Joint Compilation jointly resolves labels and record selection while preserving candidate meanings and previously fixed fields. Extensive experiments on three structured extraction tasks across five datasets show that SDA with GPT-4o-mini outperforms the strongest evaluated same-backbone LLM baseline on each dataset, with an average gain of 20.78 F1 points.

## 1 INTRODUCTION

Structured extraction identifies task-relevant semantic elements and their relations in free-form text and organizes them into predefined records. It encompasses tasks such as relation extraction, event extraction, and aspect-based sentiment analysis (Wadhwa et al., 2023; You et al., 2022; Pontiki et al., 2014). By making information expressed in language accessible to downstream systems, structured extraction supports applications such as knowledge base construction and fine-grained opinion understanding (Huguet Cabot & Navigli, 2021; Pontiki et al., 2014).

Early studies developed task-specific supervised models through tagging-based prediction (Zheng et al., 2017; Shang et al., 2022a), table filling (Wang & Lu, 2020), and sequence-to-sequence generation (Zhang et al., 2021a; Huguet Cabot & Navigli, 2021). These approaches achieved strong performance, but their representations and decoding procedures were tailored to particular extraction schemas, making it costly to accommodate new tasks and annotation requirements. Unified supervised extraction models subsequently introduced shared representations and extraction operations across heterogeneous schemas, enabling joint prediction of the elements and links within target records (Lu et al., 2022; Lou et al., 2023). This reduces task-specific architecture design, but supervised performance still relies on annotated data to learn schema-conditioned decisions.

The emergence of large language models (LLMs) has introduced a new paradigm for structured extraction through in-context learning (ICL), allowing models to perform extraction using task instructions and a few demonstrations without updating their parameters (Brown et al., 2020; Wadhwa et al., 2023). Despite this flexibility, empirical studies have reported substantial performance gaps between few-shot ICL and task-specific supervised models (Ma et al., 2023; Han et al., 2023). Prior work has linked these gaps to insufficient task specification and introduced learned annotation guidelines (Pang et al., 2023). Task-aware retrieval and label-informed explanations address weak demonstration relevance and unclear input–label mappings (Wan et al., 2023). A remaining question is whether demonstrations guide annotation alone or also change which source meanings are extracted.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A controlled analysis in Section 3.1 shows that predicted record counts shift with demonstration patterns, even for unchanged source texts. On Rest15, single-record demonstrations tend to induce under-prediction on inputs containing multiple records, whereas demonstrations with more records increase the mean predicted count. This suggests that demonstrations may constrain the model's interpretation of semantic content, beyond specifying annotation conventions.

Inspired by this finding, we propose SDA, an LLM-based structured extraction framework that separates source-based semantic discovery from demonstration-guided annotation. Its core idea is to establish a natural-language candidate space of elements and relations before introducing dataset-specific annotation conventions. Specifically, the framework comprises three modules. Semantic Discovery combines complementary, task-guided readings to construct this space without labeled demonstrations. Candidate Projection retrieves relevant precedents from an annotation memory by matching candidate anchors, then uses their paired natural relations and standard records to normalize candidates and delimit remaining annotation choices. Joint Compilation uses relevant precedents to jointly resolve labels and record selection, preserving candidate meanings and the fields fixed during projection.

Extensive experiments on three structured extraction tasks, namely Aspect Sentiment Quad Prediction (ASQP), Relational Triple Extraction (RTE), and Event Detection (ED), across five datasets demonstrate the effectiveness of SDA. Using GPT-4o-mini, SDA outperforms the strongest evaluated same-backbone LLM baselines on all five datasets by an average of 20.78 F1 points. Ablation studies show the benefits of multi-view discovery and candidate-specific annotation evidence, while further analyses demonstrate advantages under limited annotation budgets.

Our contributions are as follows:

- We identify demonstration-dependent changes in extraction outputs and motivate separating source-based discovery of elements and relations from demonstration-guided annotation.
- We propose SDA, a framework that constructs a natural-language candidate space, projects it into finite annotation choices, and jointly compiles the selected candidates into task records.
- Extensive experiments across three tasks and five datasets demonstrate SDA's extraction gains, with ablation studies supporting the contributions of multi-view discovery and candidate-specific annotation evidence.

## 2 Related Work

### 2.1 Structured Extraction

Structured extraction identifies semantic elements and their relations to produce predefined records. Representative tasks include Aspect Sentiment Quad Prediction (ASQP), Relational Triple Extraction (RTE), and Event Detection (ED) (Zhang et al., 2021a; Wei et al., 2020; Gui et al., 2024). Task-specific supervised methods encode elements and their associations through tagging-based prediction (Zheng et al., 2017; Wei et al., 2020; Shang et al., 2022a), table-filling and joint architectures (Wang & Lu, 2020; Wang et al., 2020; Zheng et al., 2021; Shang et al., 2022b), or sequence-to-sequence generation (Zeng et al., 2018; Zhang et al., 2021b;a; Huguet Cabot & Navigli, 2021; Gou et al., 2023). These methods model dependencies among the components of complete records. Their extraction formulations and supervision remain closely tied to individual tasks and annotation schemas.

Unified extraction models share representations and extraction operations across tasks. UIE encodes heterogeneous structures in a common extraction language and controls generation through schema prompts (Lu et al., 2022). USM distinguishes structuring from conceptualizing, implementing both through token-linking operations over jointly encoded text and schema (Lou et al., 2023). TAGPRIME adopts a shared sequence-tagging formulation, using condition-specific priming words to guide relational extraction (Hsu et al., 2023). UIE and USM also report transfer under low-resource settings (Lu et al., 2022; Lou et al., 2023). These advances reduce task-specific architectural design, while the shared extractor must still learn how target records are composed. ASQP, for example, jointly predicts an aspect category, aspect term, opinion term, and sentiment polarity in a quadruple (Zhang et al., 2021a). In supervised use, learning these linked fields includes their associations, span conventions, and label mappings, which continue to depend on task annotations.

### 2.2 LLM-BASED STRUCTURED EXTRACTION

Large language models (LLMs) offer a complementary route in which natural-language instructions specify extraction targets and output schemas. This route includes both instruction tuning and in-context learning (ICL). IEPile aggregates datasets into a schema-conditioned instruction corpus to train models for extraction across tasks (Gui et al., 2024). ICL specifies the task through instructions and demonstrations while keeping model parameters frozen (Brown et al., 2020; Wadhwa et al., 2023). Empirical studies report substantial gaps between few-shot ICL and task-specific supervised models (Ma et al., 2023; Han et al., 2023). Accordingly, recent work enriches the context with explicit task rules, relevant demonstrations, and explanations of input–label mappings.

Guideline Learning derives reusable task guidelines from extraction errors, making annotation criteria explicit (Pang et al., 2023). GPT-RE combines task-aware demonstration retrieval with gold label-induced reasoning to clarify input–label mappings (Wan et al., 2023). SimRP selects syntactically and semantically similar demonstrations for ASQP (Jian et al., 2025), while $I^2$CL combines triple-semantics-aware selection with tabular prompting for RTE (Li et al., 2024). These approaches improve the task information supplied for prediction. SDA instead changes when annotation examples are introduced: Semantic Discovery first proposes source-grounded candidates without labeled examples, and Candidate Projection and Joint Compilation then use candidate-specific precedents to normalize fields and resolve open annotation choices.

## 3 METHOD

### 3.1 DEMONSTRATIONS SHAPE EXTRACTION OUTPUTS

As shown in Figure 1, ICL's predicted record counts on Rest15 vary across demonstration sets containing different numbers of records per example. Single-record demonstrations yield mean predictions of 1.74 for two-record inputs and 2.90 for inputs with at least three gold records (gold mean: 3.36). For the latter group, the mean rises to 3.29 and 3.60 with two- and three-record demonstrations, respectively, while the inputs and five-example budget remain fixed. In contrast, the mean prediction for single-record inputs remains close to one across all three conditions, indicating stronger count shifts on relation-rich inputs. These shifts suggest that demonstration patterns may influence what the model extracts, beyond specifying annotation conventions. This observation motivates discovering source-supported elements and relations before applying demonstration-guided annotation. Appendix A provides a detailed analysis.

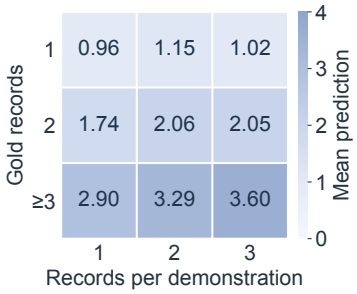

Figure 1: ICL count shifts on Rest15 ASQP.

### 3.2 TASK DEFINITION

Given a source text $x = (w_1, \ldots, w_n)$ and a predefined schema $\mathcal{S}$, structured extraction identifies semantic elements and their relations as a variable-sized set of records:

$$\mathcal{Y}(x) = \{y_i\}_{i=1}^{N}, \qquad y_i = \big(a_i^{(1)}, \ldots, a_i^{(m)}, \ell_i^{(1)}, \ldots, \ell_i^{(q)}\big). \tag{1}$$

Here, $a_i^{(j)}$ denotes a source-grounded element or a permitted implicit placeholder, and $\ell_i^{(k)}$ is a schema-defined attribute. A record may contain one element, such as an event trigger paired with its type, or several linked elements. The schema specifies $m$ element fields and $q$ attribute fields, together with their meanings and legal labels. Before assigning these annotations, we represent a candidate as $r_i = \langle \mathbf{u}_i, d_i, e_i \rangle$, where $\mathbf{u}_i$ identifies one or more source-grounded elements or anchors, $d_i$ describes the expressed event or connection in natural language, and $e_i$ records source evidence. This representation captures a candidate's meaning without fixing its exact annotation boundaries or target labels. A natural-language task descriptor $\tau$ specifies the relevant kinds of information, such as evaluations or factual connections.

### 3.3 OVERVIEW

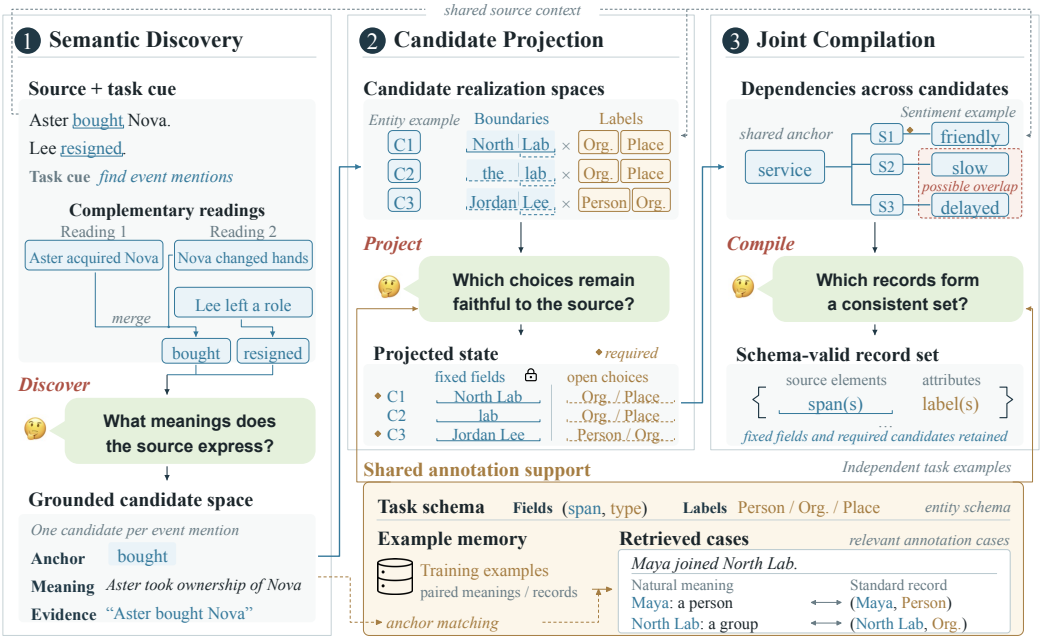


Figure 2: SDA framework with independent event, entity, and sentiment examples. Blue denotes source content and candidate states, and gold denotes annotation choices and support. Locks mark fixed fields, and diamonds mark required candidates.

As shown in Figure 2, SDA consists of three modules: Semantic Discovery, Candidate Projection, and Joint Compilation. Semantic Discovery reads the source through multiple task-guided perspectives to form a semantic candidate set $\mathcal{C}(x)$ without annotation examples. Candidate Projection matches relation anchors against the annotation memory $\mathcal{A}$ and uses the paired standard annotations to construct the projected state $\mathcal{Z}(x)$, which specifies fixed fields and remaining choices. Joint Compilation uses the retrieved example collection $\mathcal{E}(x)$ to resolve labels and record selection jointly, producing the predicted record set $\widehat{\mathcal{Y}}(x)$. Together, these modules connect source-based discovery with dataset-specific annotation while preserving the discovered meanings and normalized decisions:

$$\underbrace{x}_{\text{source text}} \xrightarrow[\tau,\mathcal{V}_\tau]{\textsc{Discover}} \underbrace{\mathcal{C}(x)}_{\text{semantic candidates}} \xrightarrow[x,\mathcal{A}]{\textsc{Project}} \underbrace{\mathcal{Z}(x)}_{\text{projected state}} \xrightarrow[x,\mathcal{S},\mathcal{E}(x)]{\textsc{Compile}} \underbrace{\widehat{\mathcal{Y}}(x)}_{\text{structured records}} . \tag{2}$$

Here, $\mathcal{C}(x) = \{r_i\}_{i=1}^{M}$ is the discovered candidate set, and $\mathcal{A}$ pairs training natural relations with standard annotations. We denote by $K$ the number of annotation examples provided to Joint Compilation per input; $\mathcal{E}(x)$ retains the candidate associations of the retrieved examples. The projected state $\mathcal{Z}(x)$ stores each candidate's admissible record set $\widetilde{\Omega}_i$ and the required candidate indices $I_{\text{keep}}$. Model parameters remain frozen throughout.

### 3.4 SEMANTIC DISCOVERY

Semantic Discovery identifies task-relevant elements and their connections before applying dataset annotation conventions. Generating complete records directly combines deciding which meanings to propose with deciding how to annotate them, allowing examples to influence both decisions. We therefore first ask the model to propose semantic candidates from the source and task description.

To discover these connections individually, we expand the task description into reading questions that attend to different semantic content. A source may connect several elements through different expressions, and one element may participate in multiple connections. A single reading can leave some of these meanings unexplored. Each perspective seeks distinct connections or event mentions through its own question, without prescribing a candidate count. Their outputs are then combined to recover meanings that an individual reading may miss.

Let $\mathcal{V}_\tau$ denote the reading perspectives, $P_v$ the prompt constructor for perspective $v$, and $L_\theta$ the frozen LLM. Each perspective receives only the source $x$ and task description $\tau$:

$$\mathcal{C}(x) = \mathrm{GroundMerge}\left(\bigcup_{v\in\mathcal{V}_\tau} L_\theta(P_v(x,\tau))\right). \tag{3}$$

$\mathrm{GroundMerge}$ aligns candidates with source expressions and consolidates repeated readings according to their participating elements and meanings. Candidates sharing an element but expressing different connections remain separate. Exact annotation boundaries and classification labels remain unresolved.

### 3.5 Candidate Projection

After discovery, Candidate Projection uses training precedents to convert natural relations into representations that follow dataset conventions. A connection supported by the source may require different element boundaries or fall outside the dataset's annotation scope. The model uses the transformations illustrated by the examples to make these normalization decisions for each candidate.

The annotation memory $\mathcal{A}$ is organized by training source. Let $b$ index a training example and $h$ a record within it. Each standard record $y^{\mathrm{tr}}_{b,h}$ is presented alongside the natural relation $r^{\mathrm{tr}}_{b,h}$ that expresses its annotated meaning, with the source $x_b$ retained as context. These correspondences remain explicit when a source contains multiple records. The natural relation carries anchors $\mathbf{u}^{\mathrm{tr}}_{b,h}$; its standard record uses the element fields $a^{(j)}_{b,h}$ and classification fields $\ell^{(k)}_{b,h}$ defined in Eq. (1). Each discovered candidate initiates retrieval by comparing corresponding anchors within the two natural relations:

$$s_{i,b,h} = \mathrm{sim}_{\mathrm{anchor}}\left(\mathbf{u}_i, \mathbf{u}^{\mathrm{tr}}_{b,h}\right). \tag{4}$$

These matches associate relevant training records with each candidate, forming candidate evidence $\mathcal{E}_i$. Anchor matching establishes relevance to the current relation, while the paired standard records demonstrate its annotation boundaries and scope. The model applies these transformation patterns to $r_i$, using the source to resolve element fields and retain the remaining annotation choices. We represent the projection of $r_i$ by an admissible record set $\widetilde{\Omega}_i$, whose elements $y$ have the field structure in Eq. (1). This set is represented through fixed fields and unresolved options: resolved element boundaries remain fixed, while open classification and other annotation choices are deferred to Compilation. Let $I_{\mathrm{keep}}$ contain the indices of candidates that the task requires to be retained; final recordability remains open for the others. The projected state is

$$\mathcal{Z}(x) = \left\langle \{(i, r_i, \widetilde{\Omega}_i)\}_{i=1}^{M}, I_{\mathrm{keep}} \right\rangle. \tag{5}$$

Projection thus specifies which fields are fixed, which choices remain open, and which candidates must be retained.

### 3.6 Joint Compilation

Joint Compilation completes the projected candidates as a set while preserving the decisions made during normalization. Candidates may share elements while expressing distinct meanings, or normalize to duplicate records. Considering them together allows the compiler to resolve the remaining labels and record choices while checking that required candidates are represented and distinct meanings remain separate. The compiler uses $K$ annotation examples $\mathcal{E}(x)$ selected at input level (Appendix E.1), examining their classification fields $\ell^{(k)}_{b,h}$ to resolve the current record's open fields $\ell^{(k)}$. It checks each proposed record against the source $x$ and candidate evidence $e_i$: the participating elements, expressed connection, and direction must match the candidate's meaning. The examples guide how that meaning is labeled under the dataset conventions, with each label drawn from the schema-legal set $\mathcal{L}^{(k)}_{\mathcal{S}}$.

Compilation decisions form an assignment set $\mathcal{T}$, where $(i, y)$ realizes candidate $r_i$ as a record $y \in \widetilde{\Omega}_i$. This restriction preserves element fields $a^{(j)}$ and other fields fixed during Projection. The prompt requires a record for every candidate in $I_{\mathrm{keep}}$; candidates with open recordability may be omitted when the source meaning or annotation scope does not support them. Task requirements

determine whether one or multiple records are allowed per candidate, with distinct meanings retained and redundant realizations consolidated. The compiler makes these decisions in one call over the source and complete projected state:

$$\widehat{\mathcal{T}}(x) = L_\theta(P_{\text{compile}}(x, \mathcal{Z}(x), \mathcal{E}(x)))\,. \tag{6}$$

The prompt requests semantic and preservation checks, while the parser verifies candidate and option ownership, label legality, source positions, and field completeness. Validated assignments are serialized into the record structure in Eq. (1), with candidate indices removed; stable sorting and exact deduplication produce $\widehat{\mathcal{Y}}(x)$. Unparsable responses count as empty predictions during evaluation.

# 4 Experimental Setup

## 4.1 Tasks and Datasets

The experiments cover three structured extraction tasks across five benchmarks: **Aspect Sentiment Quad Prediction (ASQP)** jointly extracts aspects, opinions, categories, and polarities on the processed **Rest15** and **Rest16** ASQP splits released by Zhang et al. (2021a); **Relational Triple Extraction (RTE)** extracts directed subject–relation–object triples on the 24-relation **NYT** split released with CopyRE and the 216-relation **WebNLG** split distributed in the standard ETL/TPLinker format (Zeng et al., 2018; Wang et al., 2020; Zheng et al., 2021); and **Event Detection (ED)** identifies typed event triggers on the ACE2005 English split converted and released by **IEPile** (Walker et al., 2006; Gui et al., 2024). For ACE2005, records sharing the same source sentence are merged into one input instance before inference.

**Table 1:** Input and label counts.

| Dataset | Train | Dev | Test | #Labels |
|---|---|---|---|---|
| Rest15 | 834 | 209 | 537 | 13 |
| Rest16 | 1,264 | 316 | 544 | 13 |
| NYT | 56,196 | 5,000 | 5,000 | 24 |
| WebNLG | 5,019 | 500 | 703 | 216 |
| ACE2005 | 3,257 | 319 | 293 | 33 |

Label counts in Table 1 denote aspect categories, relation types, or event types; ASQP additionally uses three sentiment labels.

## 4.2 Baselines

**Supervised baselines.** The ASQP comparisons include GAS (Zhang et al., 2021b), Paraphrase (Zhang et al., 2021a), MvP (Gou et al., 2023), and SimRP-SFT (Jian et al., 2025). The RTE comparisons include TPLinker (Wang et al., 2020), PRGC (Zheng et al., 2021), OneRel (Shang et al., 2022a), and OD-RTE (Ning et al., 2023). The ED comparisons include DMCNN (Chen et al., 2015), $S^2$-JDN (Li et al., 2021), and the instruction-tuned extraction models OneKE (Gui et al., 2024), YAYI-UIE (Xiao et al., 2023), and InstructUIE (Wang et al., 2023a).

**LLM-based baselines.** The generic comparisons include ICL (Brown et al., 2020), CoT (Wei et al., 2022), Self-Consistency (Wang et al., 2023b), ProgCo (Song et al., 2025), and SR-LLM (Zhang et al., 2025). Task-specific comparisons are SimRP-ICL (Jian et al., 2025) for ASQP, TableIE (Li et al., 2024) for RTE, and KeyCP++ (Li & Deng, 2025) for ED.

## 4.3 Metrics

Following prior work on these tasks (Zhang et al., 2021a; Wang et al., 2020; Gui et al., 2024), the primary metrics are micro-averaged precision, recall, and F1. The main scores require exact matching of complete task records. To assess intermediate stages, the analysis uses broad precision (BP), broad recall (BR), and broad F1 (BF1), which omit schema labels and allow compatible span-boundary variation. The matching and counting rules are given in Appendix B.

### 4.4 Implementation Details

Supervised baseline experiments use two NVIDIA L40S GPUs with 48 GB each. LLM experiments use APIs, primarily GPT-4o-mini, with additional GPT-5.6-Luna evaluations of SDA. LLM parameters remain frozen. Generic prompting baselines reuse a fixed random set of five training examples on ASQP or 24 on RTE for every input; temperature is zero except where sampling is part of the method. SimRP-ICL uses the published GPT-4o-mini five-shot F1 scores. The evaluated ED LLM baselines use five fixed examples per schema batch. SDA builds its annotation memory from the training split and uses $K = 5$ compilation examples by default. Appendix D provides task-specific prompt templates.

## 5 Results and Analysis

### 5.1 Main Results

Table 2: Extraction results (%). Bold marks the best LLM-based scores.

**(a) Aspect Sentiment Quad Prediction**

| | Rest15 | | | Rest16 | | |
|---|---|---|---|---|---|---|
| Method | P | R | F1 | P | R | F1 |
| Supervised reference | | | | | | |
| GAS | 45.31 | 46.70 | 45.98 | 54.54 | 57.62 | 56.04 |
| Paraphrase | 46.16 | 47.72 | 46.93 | 56.63 | 59.30 | 57.93 |
| MvP | – | – | 51.04 | – | – | 60.39 |
| SimRP-SFT | 53.12 | 53.50 | 53.30 | 62.74 | 64.12 | 63.42 |
| LLM-based methods (GPT-4o-mini) | | | | | | |
| ICL | 33.37 | 34.21 | 33.79 | 41.83 | 43.55 | 42.67 |
| CoT | 29.22 | 23.52 | 26.06 | 35.63 | 33.67 | 34.62 |
| Self-Consistency | 34.00 | 32.08 | 33.01 | 44.81 | 43.80 | 44.30 |
| ProgCo | 30.30 | 26.79 | 28.44 | 42.46 | 40.55 | 41.49 |
| SR-LLM | 28.59 | 29.56 | 29.07 | 38.30 | 39.55 | 38.92 |
| SimRP-ICL | – | – | 33.33 | – | – | 40.44 |
| Ours | | | | | | |
| SDA (GPT-4o-mini) | 48.77 | 39.75 | 43.80 | **53.63** | 47.18 | 50.20 |
| SDA (GPT-5.6-Luna) | **49.86** | **42.49** | **45.89** | 52.68 | **53.15** | **52.91** |

**(b) Relational Triple Extraction**

| | NYT | | | WebNLG | | |
|---|---|---|---|---|---|---|
| Method | P | R | F1 | P | R | F1 |
| Supervised reference | | | | | | |
| TPLinker | 91.40 | 92.60 | 92.00 | 88.90 | 84.50 | 86.70 |
| PRGC | 93.50 | 91.90 | 92.70 | 89.90 | 87.20 | 88.50 |
| OneRel | 93.20 | 92.60 | 92.90 | 91.80 | 90.30 | 91.00 |
| OD-RTE | 94.20 | 93.60 | 93.90 | 92.80 | 92.10 | 92.50 |
| LLM-based methods (GPT-4o-mini) | | | | | | |
| ICL | 24.11 | 15.69 | 19.01 | 39.32 | 39.39 | 39.35 |
| CoT | 14.71 | 7.59 | 10.01 | 38.52 | 38.95 | 38.74 |
| Self-Consistency | 25.39 | 15.80 | 19.48 | 40.10 | 39.58 | 39.84 |
| ProgCo | 20.29 | 6.93 | 10.34 | 39.37 | 30.18 | 34.17 |
| SR-LLM | 24.12 | 19.31 | 21.45 | 41.24 | 40.45 | 40.84 |
| TableIE | 38.48 | 25.00 | 30.31 | 43.16 | 45.55 | 44.32 |
| Ours | | | | | | |
| SDA (GPT-4o-mini) | 93.01 | **75.75** | **83.50** | 66.67 | **76.91** | 71.42 |
| SDA (GPT-5.6-Luna) | **99.04** | 61.07 | 75.55 | **69.71** | 73.47 | **71.54** |

Table 2 compares SDA with same-backbone GPT-4o-mini baselines and supervised performance references. We additionally evaluate SDA with GPT-5.6-Luna to examine how backbone choice affects the framework's extraction performance.

**Gains span all three tasks and are largest for relation extraction.** With GPT-4o-mini, SDA outperforms the strongest evaluated same-backbone LLM baseline on every dataset in Table 2, with F1 gains ranging from 5.90 to 53.19 points. NYT and WebNLG show the largest improvements, with gains in both precision and recall. On NYT, for example, precision increases from 38.48 to 93.01 and recall from 25.00 to 75.75 relative to TableIE. The framework thus recovers more correct relations while also improving the accuracy of its predictions.

**(c) Event Detection**

| | ACE2005 | | |
|---|---|---|---|
| Method | P | R | F1 |
| Supervised reference | | | |
| DMCNN | 66.67 | 68.94 | 67.79 |
| $S^2$-JDN | 67.80 | 68.26 | 68.03 |
| OneKE | 65.41 | 70.99 | 68.09 |
| YAYI-UIE | 68.94 | 68.94 | 68.94 |
| InstructUIE | 78.16 | 78.16 | 78.16 |
| LLM-based methods (GPT-4o-mini) | | | |
| ICL | 14.05 | 40.27 | 20.83 |
| CoT | 22.16 | 27.30 | 24.46 |
| Self-Consistency | 14.60 | 40.61 | 21.48 |
| ProgCo | – | – | – |
| SR-LLM | 12.78 | 40.61 | 19.44 |
| KeyCP++ | 24.85 | 42.32 | 31.31 |
| Ours | | | |
| SDA (GPT-4o-mini) | **32.22** | 49.49 | 39.03 |
| SDA (GPT-5.6-Luna) | 31.23 | **84.30** | **45.57** |

**Changing the backbone improves event recall, while precision remains a bottleneck.** On the full ACE2005 test set, Luna increases recall from 49.49 to 84.30 and F1 from 39.03 to 45.57, while precision decreases slightly from 32.22 to 31.23. SDA recovers more correct events but still produces many false positives. On NYT, Luna achieves 99.04 precision and 61.07 recall, with missed relations being the main limitation.

**A gap to supervised extraction remains.** On the full test sets, supervised references outperform GPT-4o-mini SDA on all five datasets. On ACE2005, Luna reaches 45.57 F1 versus InstructUIE's 78.16; limited precision still constrains extraction despite higher event recall. Overall, SDA outperforms the evaluated same-backbone baselines across all three tasks, while switching backbones changes the precision–recall trade-off without consistent F1 gains.

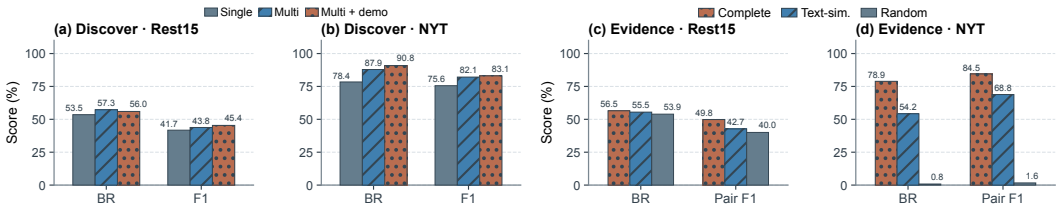


Figure 3: Semantic Discovery (a–b) and Candidate Projection (c–d) ablations.

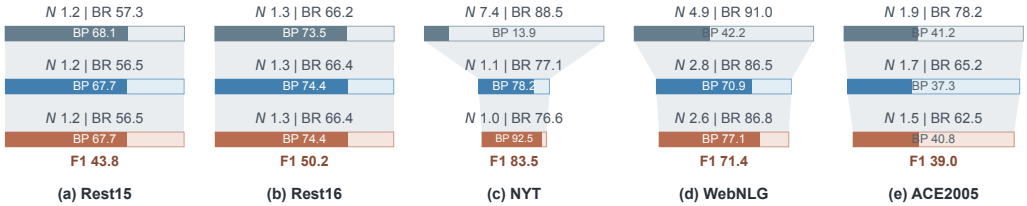


Figure 4: Candidate evolution on five datasets. Rows correspond to Semantic Discovery, Candidate Projection, and Joint Compilation, from top to bottom.

### 5.2 Ablations and Stage Analysis

In this section, we investigate the internal mechanisms of SDA. Specifically, we ask three questions: (1) How do multiple readings and demonstrations affect candidate coverage in Semantic Discovery? (2) How does annotation evidence affect normalization in Candidate Projection? (3) How do candidate coverage and precision evolve across the three modules?

**Multiple readings improve coverage in Semantic Discovery.** The ablation compares example-free single and multiple readings, then adds five labeled examples to the multi-view setting to measure their incremental effect. We evaluate these settings on Rest15 and NYT. Figure 3(a,b) reports broad recall (BR) at this stage and final exact-match F1. Multiple readings increase BR from 53.5 to 57.3 and final F1 from 41.7 to 43.8 on Rest15; on NYT, BR increases from 78.4 to 87.9 and F1 from 75.6 to 82.1. Thus, multi-view discovery improves coverage by 3.8 and 9.5 points, respectively, without annotation examples.

**Demonstrations offer limited gains in Semantic Discovery.** Adding examples changes BR from 57.3 to 56.0 on Rest15 and from 87.9 to 90.8 on NYT. These changes ($-1.3$ and $+2.9$ points) are smaller than the gains from multiple readings. Final F1 increases by 1.6 and 1.0 points, respectively, with the latter gain not recurring in both NYT input blocks. Additional examples thus have a limited effect within multi-view discovery; the Rest15 F1 gain does not reflect broader semantic coverage.

**Candidate Projection depends on annotation evidence.** The evidence comparison in Figure 3(c,d) uses complete, whole-sentence-similarity, and random-example evidence on the full Rest15 and NYT test sets. Pair F1 requires exact endpoints but omits labels. Replacing complete evidence with whole-sentence similarity reduces pair F1 from 49.8 to 42.7 on Rest15 and from 84.5 to 68.8 on NYT; random examples reduce it further to 40.0 and 1.6 (three fixed draws on NYT, one on Rest15). Rest15 BR changes by only 1.0 point despite a 7.1-point pair F1 drop, indicating an effect on exact boundary normalization; on NYT, BR also falls from 78.9 to 54.2, indicating lost relation coverage. These configurations change the evidence available for annotation and do not isolate retrieval ranking alone.

**Candidate evolution across modules differs by task.** Figure 4 reports the mean number of unique label-free candidates per input ($N$), broad precision (BP), broad recall (BR), and final F1 on five full test sets. ASQP preserves pairs from Candidate Projection to Joint Compilation while completing classification fields. RTE filters further: NYT contracts from 37,019 candidates in total to 5,719 and then 4,796, raising BP from 13.9 to 92.5 as BR falls from 88.5 to 76.6; WebNLG contracts from 3,436 to 1,796 in total while retaining 86.8 BR. On ACE2005, mean candidate counts fall from 1.9 to 1.5, but BP decreases from 41.2 to 40.8 as BR falls from 78.2 to 62.5. Thus, ASQP preserves pairs through Joint Compilation, RTE trades coverage for precision, and ED loses coverage without a precision gain.

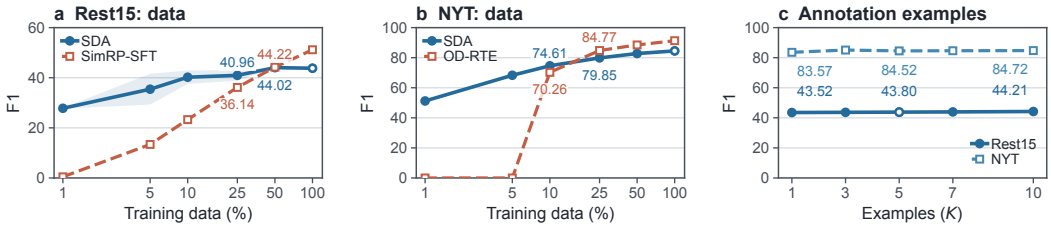


Figure 5: Effects of training fraction (a–b) and $K$ (c) on Rest15 and NYT.

Table 3: Mean per-input inference cost ($n = 100$). Accounting: Appendix E.2.

| | Rest15 ($n = 100$) | | | | | NYT ($n = 100$) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | F1 | Input | Output | Time (s) | Tok./F1 | F1 | Input | Output | Time (s) | Tok./F1 |
| ICL | 37.06 | 396 | 25 | 1.47 | 11.36 | 22.30 | 2,952 | 30 | 1.87 | 133.67 |
| CoT | 29.00 | 456 | 152 | 2.78 | 20.99 | 8.20 | 2,981 | 108 | 2.97 | 376.80 |
| Self-Consistency | 36.18 | 1,189 | 73 | 4.38 | 34.86 | 20.08 | 8,856 | 89 | 6.02 | 445.53 |
| ProgCo | 29.35 | 8,306 | 685 | 22.09 | 306.31 | 14.10 | 37,672 | 701 | 25.54 | 2,722.08 |
| SR-LLM | 32.59 | 851 | 114 | 3.95 | 29.60 | 19.59 | 3,800 | 83 | 4.02 | 198.16 |
| SimRP-ICL | 38.96 | 367 | 24 | 2.98 | 10.03 | – | – | – | – | – |
| TableIE | – | – | – | – | – | 29.55 | 3,292 | 50 | 2.11 | 113.12 |
| SDA | 49.65 | 6,801 | 330 | 46.12 | 143.62 | 83.39 | 7,446 | 1,979 | 38.58 | 113.02 |

### 5.3 Annotation Efficiency and Sensitivity to $K$

In this section, we ask two questions: (1) How effective is SDA with limited annotations? (2) How sensitive is it to the number of retrieved examples $K$?

**SDA is effective under limited annotation budgets.** Figure 5(a,b) varies the training fraction for annotation memory or supervised training, averaging two subset seeds at reduced fractions. Rest15 uses its full test set; NYT uses 300 fixed inputs alongside a full-test OD-RTE reference. At 10%, SDA reaches 40.21 F1 versus SimRP-SFT's 23.29 on Rest15 and 74.61 on NYT. Full-data supervised references score higher.

**Performance is relatively insensitive to $K$.** With the full annotation memory, Figure 5(c) varies $K \in \{1, 3, 5, 7, 10\}$, fixing Candidate Projection for Rest15 and rerunning Candidate Projection and Joint Compilation on the same 300 NYT inputs. F1 spans 43.52–44.21 on Rest15 and 83.57–85.09 on NYT, changing by only 0.69 and 1.52 points, respectively. At $K = 1$, the scores are already 43.52 and 83.57, indicating limited sensitivity within the tested range.

### 5.4 Inference Cost

To quantify the inference cost associated with SDA's accuracy gains, Table 3 compares F1 and recorded inference costs on 100 shared test inputs per dataset. On NYT, SDA reaches 83.39 F1 versus TableIE's 29.55 at nearly identical reported tokens per F1 point (113.02 versus 113.12). On Rest15, a 10.69-point gain over SimRP-ICL requires 7,131 versus 391 tokens per input. Mean summed request time is 46.12 seconds on Rest15 and 38.58 on NYT.

## 6 Conclusion

In this paper, we introduce SDA, an LLM-based structured extraction framework that separates source-based semantic discovery from demonstration-guided annotation. Motivated by the influence of demonstration patterns on extraction outputs, SDA discovers natural-language candidates through complementary readings, retrieves candidate-relevant annotation evidence, and jointly compiles the normalized candidates into structured records. Experiments with GPT-4o-mini on five datasets spanning ASQP, RTE, and ED show consistent gains over the evaluated same-backbone LLM baselines. Ablation studies highlight the complementary roles of multi-view discovery in improving semantic coverage and annotation evidence in guiding candidate normalization. Remaining limitations in inference cost motivate future work toward more reliable and efficient structured extraction.

### AI USE STATEMENT

In this work, we used generative AI tools to refine research hypotheses; provide feedback on research methodology and experimental design; implement and inspect code; interpret results; search and summarize literature; and draft, edit, and format parts of the manuscript. All AI-assisted outputs were reviewed by the authors and, where applicable, checked against experiment scripts, saved predictions, evaluation artifacts, and primary bibliographic sources. No model-generated judgment served as ground-truth annotation. The authors take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with AI assistance.

### REPRODUCIBILITY STATEMENT

The method and inference procedure are described in Section 3; datasets, baselines, metrics, and implementation settings are documented in Section 4; and prompt templates, metric details, additional diagnostics, and case studies are provided in the appendices. The anonymous supplementary package will include the code, frozen prompts, split identifiers, model and decoding settings, per-sentence predictions, evaluation scripts, and cost records required to reproduce the reported experiments.

## REFERENCES

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 167–176. Association for Computational Linguistics, 2015. doi: 10.3115/v1/P15-1017. URL https://aclanthology.org/P15-1017/.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. MvP: Multi-view prompting improves aspect sentiment tuple prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4380–4397. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.240. URL https://aclanthology.org/2023.acl-long.240/.

Honghao Gui, Lin Yuan, Hongbin Ye, Ningyu Zhang, Mengshu Sun, Lei Liang, and Huajun Chen. IEPile: Unearthing large scale schema-conditioned information extraction corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 127–146, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.13. URL https://aclanthology.org/2024.acl-short.13/.

Ridong Han, Chaohao Yang, Tao Peng, Prayag Tiwari, Xiang Wan, Lu Liu, and Benyou Wang. An empirical study on information extraction using large language models. *arXiv preprint arXiv:2305.14450*, 2023. URL https://arxiv.org/abs/2305.14450.

I-Hung Hsu, Kuan-Hao Huang, Shuning Zhang, Wenxin Cheng, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. TAGPRIME: A unified framework for relational structure extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12917–12932, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.723. URL https://aclanthology.org/2023.acl-long.723/.

Pere-Lluís Huguet Cabot and Roberto Navigli. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2370–2381. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.204. URL https://aclanthology.org/2021.findings-emnlp.204/.

Zhongquan Jian, Yanhao Chen, Jiajian Li, Shaopan Wang, Xiangjian Zeng, Junfeng Yao, Xinying An, and Qingqiang Wu. SimRP: Syntactic and semantic similarity retrieval prompting enhances aspect sentiment quad prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (23):24248–24256, 2025. doi: 10.1609/aaai.v39i23.34601. URL https://ojs.aaai.org/index.php/AAAI/article/view/34601.

Guozheng Li, Wenjun Ke, Peng Wang, Zijie Xu, Ke Ji, Jiajun Liu, Ziyu Shang, and Qiqing Luo. Unlocking instructive in-context learning with tabular prompting for relational triple extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 17131–17143. ELRA and ICCL, 2024. URL https://aclanthology.org/2024.lrec-main.1488/.

Rui Li, Wenlin Zhao, Cheng Yang, and Sen Su. Treasures outside contexts: Improving event detection via global statistics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 2625–2635. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.206. URL https://aclanthology.org/2021.emnlp-main.206/.

Ziheng Li and Zhi-Hong Deng. Keyword-centric prompting for one-shot event detection with self-generated rationale enhancements. In *ECAI 2025: 28th European Conference on Artificial Intelligence*, pp. 4105–4112, 2025. doi: 10.3233/FAIA251301. URL https://doi.org/10.3233/FAIA251301.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Universal information extraction as unified semantic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 13318–13326, 2023. doi: 10.1609/aaai.v37i11.26563. URL https://ojs.aaai.org/index.php/AAAI/article/view/26563.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5755–5772, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.395. URL https://aclanthology.org/2022.acl-long.395/.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10572–10601. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.710. URL https://aclanthology.org/2023.findings-emnlp.710/.

Jinzhong Ning, Zhihao Yang, Yuanyuan Sun, Zhizheng Wang, and Hongfei Lin. OD-RTE: A one-stage object detection framework for relational triple extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11120–11135. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.623. URL https://aclanthology.org/2023.acl-long.623/.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. Guideline learning for in-context information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15372–15389. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.950. URL https://aclanthology.org/2023.emnlp-main.950/.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 27–35. Association for Computational Linguistics, 2014. doi: 10.3115/v1/S14-2004. URL https://aclanthology.org/S14-2004/.

Yu-Ming Shang, Heyan Huang, and Xian-Ling Mao. OneRel: Joint entity and relation extraction with one module in one step. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11285–11293, 2022a. doi: 10.1609/aaai.v36i10.21379. URL https://ojs.aaai.org/index.php/AAAI/article/view/21379.

Yu-Ming Shang, Heyan Huang, Xin Sun, Wei Wei, and Xian-Ling Mao. Relational triple extraction: One step is enough. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 4360–4366. International Joint Conferences on Artificial Intelligence Organization, 2022b. doi: 10.24963/ijcai.2022/605. URL https://www.ijcai.org/proceedings/2022/605.

Xiaoshuai Song, Yanan Wu, Weixun Wang, Jiaheng Liu, Wenbo Su, and Bo Zheng. ProgCo: Program helps self-correction of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 944–959, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-short.73. URL https://aclanthology.org/2025.acl-short.73/.

Somin Wadhwa, Silvio Amir, and Byron C. Wallace. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15566–15589, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.868. URL https://aclanthology.org/2023.acl-long.868/.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. ACE 2005 Multilingual Training Corpus LDC2006T06, 2006. URL https://catalog.ldc.upenn.edu/LDC2006T06.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. GPT-RE: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3534–3547. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.214. URL https://aclanthology.org/2023.emnlp-main.214/.

Jue Wang and Wei Lu. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1706–1721. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.133. URL https://aclanthology.org/2020.emnlp-main.133/.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. InstructUIE: Multi-task instruction tuning for unified information extraction, 2023a. URL https://arxiv.org/abs/2304.08085.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. TPLinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1572–1582. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.138. URL https://aclanthology.org/2020.coling-main.138/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1476–1488. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.136. URL https://aclanthology.org/2020.acl-main.136/.

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. YAYI-UIE: A chat-enhanced instruction tuning framework for universal information extraction, 2023. URL https://arxiv.org/abs/2312.15548.

Huiling You, David Samuel, Samia Touileb, and Lilja Øvrelid. EventGraph: Event extraction as semantic graph parsing. In *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, pp. 7–15, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.case-1.2. URL https://aclanthology.org/2022.case-1.2/.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 506–514. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1047. URL https://aclanthology.org/P18-1047/.

Jiahuan Zhang, Tianheng Wang, Hanqing Wu, Ziyi Huang, Yulong Wu, Dongbai Chen, Linfeng Song, Yue Zhang, Guozheng Rao, and Kaicheng Yu. SR-LLM: Rethinking the structured representation in large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3443–3462, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.172. URL https://aclanthology.org/2025.acl-long.172/.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9209–9219. Association for Computational Linguistics, 2021a. doi: 10.18653/v1/2021.emnlp-main.726. URL https://aclanthology.org/2021.emnlp-main.726/.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. Towards generative aspect-based sentiment analysis. 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)*, pp. 504–510. Association for Computational Linguistics, 2021b. doi: 10.18653/v1/2021.acl-short.64. URL https://aclanthology.org/2021.acl-short.64/.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. PRGC: Potential relation and global correspondence based joint relational triple extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6225–6235. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.486. URL https://aclanthology.org/2021.acl-long.486/.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1227–1236. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1113. URL https://aclanthology.org/P17-1113/.

# A Demonstration-Count Diagnostics

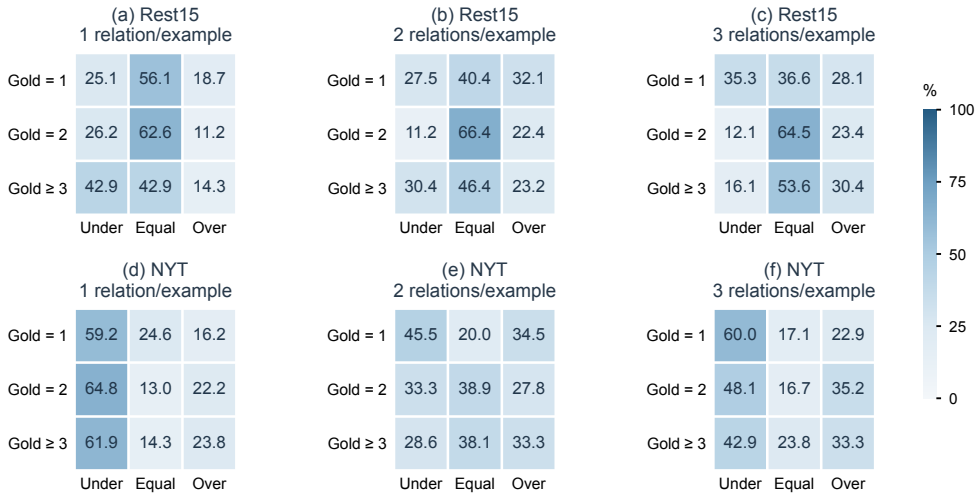


Figure 6: Direct-ICL count deviations under different demonstration cardinalities.

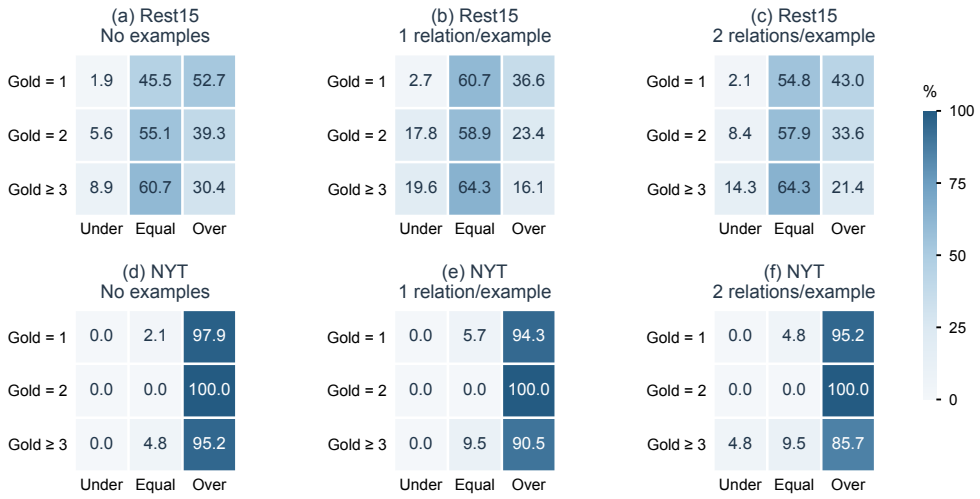


Figure 7: Natural-relation discovery count deviations without and with demonstrations.

This experiment tests whether demonstrations change output count and gold-relation coverage in direct ICL and natural-relation discovery in Section 3.1. Both use GPT-4o-mini on the same 537 Rest15 and first 600 NYT inputs. In the shared contrast, each interface uses the same five training texts with either one or two relations per text; ICL shows complete records, whereas Discovery shows natural readings without labels. Discovery also has no-example and 1/1/2/2/3-balanced controls; three-relation ICL is supplementary. Predictions and gold are reduced to distinct label-free endpoint pairs. Within the one-, two-, and $\geq$ 3-gold-pair groups, the matrices show under-, equal-, and over-output relative to each input's exact count; Table 4 gives one-to-one broad recall (BR). Failed calls contribute empty outputs without excluding inputs. Each interface keeps its reading procedure fixed across conditions.

**Output quantity responds to demonstration composition.** In Figure 6, changing from one to two relations per example increases ICL over-output on single-pair inputs from 18.7% to 32.1% on Rest15 and from 16.2% to 34.5% on NYT. Figure 7 shows that Discovery also changes its candidate count: its mean rises from 1.82 to 2.00 on Rest15 but falls from 7.84 to 7.26 on NYT. These count shifts alone do not establish a change in gold coverage. Discovery deliberately proposes a broader candidate set, so exceeding the annotated count has a different role than in complete-record prediction.

**Natural-relation coverage changes little in the shared contrast.** From one to two relations per example, discovery BR changes by $+0.80/-0.47/0.00$ points across the Rest15 strata and by $-0.19/-0.93/+4.00$ on NYT. The latter dataset's multi-pair change is three additional matches (60 to 63 of 75 gold pairs) across 21 inputs. In contrast, NYT ICL changes by $+8.95/+9.26/+25.33$ points; on Rest15, ICL coverage also varies only modestly. Natural discovery thus preserves similar coverage under these two example compositions, with a clearer contrast to direct ICL on NYT. The additional controls qualify this stability: balanced examples change two-pair recall from 71.03% to 61.68% on Rest15 and from 80.56% to 91.67% on NYT relative to no examples. Each condition uses one fixed support set, so content and relation count remain coupled. The results concern coverage sensitivity of the complete reading procedures, not an isolated causal effect of natural-language representation or a downstream F1 gain.

Table 4: Broad recall by gold-pair count under fixed demonstration conditions.

| | | | ICL BR (%) | | Discovery BR (%) | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | Gold pairs | Inputs / pairs | One | Two | None | One | Two | Balanced |
| Rest15 | 1 | 374 / 374 | 52.41 | 49.73 | 73.26 | 69.79 | 70.59 | 71.93 |
| | 2 | 107 / 214 | 59.35 | 60.75 | 71.03 | 67.29 | 66.82 | 61.68 |
| | $\geq 3$ | 56 / 189 | 60.85 | 61.90 | 69.31 | 71.96 | 71.96 | 70.37 |
| NYT | 1 | 525 / 525 | 22.48 | 31.43 | 92.19 | 92.57 | 92.38 | 91.62 |
| | 2 | 54 / 108 | 16.67 | 25.93 | 80.56 | 86.11 | 85.19 | 91.67 |
| | $\geq 3$ | 21 / 75 | 32.00 | 57.33 | 78.67 | 80.00 | 84.00 | 85.33 |

## B Broad Metric Protocol

Broad evaluation checks whether a candidate space contains the annotated semantic unit before asking whether it assigned the complete task labels. The units are ordered aspect–opinion pairs for ASQP, unordered endpoint pairs for RTE, and trigger mentions for ED. Category, polarity, predicate, predicate direction, and event type are omitted only for this diagnostic. Duplicate units are collapsed within each input.

For each input, connect a predicted unit to every gold unit it is allowed to match under Table 5, then take a maximum one-to-one bipartite matching. Let $M$ be the total number of such matches and $N_p, N_g$ the total unique predicted and gold units across inputs. Micro-averaged broad precision, recall, and F1 are

$$\mathrm{BP} = \frac{M}{N_p}, \qquad \mathrm{BR} = \frac{M}{N_g}, \qquad \mathrm{BF1} = \frac{2\mathrm{BPBR}}{\mathrm{BP} + \mathrm{BR}}. \tag{7}$$

An undefined denominator gives zero; an empty/empty input contributes no units. A grouped opinion or expanded boundary can match at most one gold relation. The same rule is applied after each stage. P/R/F1 in the main table instead require a complete task record, including its labels, under the dataset's exact scorer.

Table 5: Boundary-compatible, label-free matching by task.

| Task | Permitted broad match |
|---|---|
| ASQP | Match aspect and opinion in their original roles, with exact NULL agreement. Accept identical token sequences or contiguous containment with unchanged negation presence; aspect extensions must not add "and", "or", "but", or "nor". Alternatively, equal-length sequences may differ in one aligned token pair if both tokens have at least five characters and their Levenshtein distance is one. |
| RTE | Ignore predicate and direction. Both endpoints must match, in either order, by equality or contiguous token-sequence containment after lowercasing and extracting alphanumeric tokens. |
| ED | Ignore event type. Trigger token sequences must be identical or one must contain the other contiguously, using the same normalization as RTE. |

ASQP uses case-insensitive alphanumeric tokens, retaining internal apostrophes and hyphens. Negation is detected by "not", "no", "never", "neither", "nor", "without", "hardly", "barely", or a token containing "n't". Empty normalized endpoints do not match.

# C CASE STUDIES

Table 6 follows one source through the three stages for each task. It shows the candidate state before and after projection, including a discarded alternative where that decision determines the final record. The ASQP trace changes the pair boundary without changing its relation; the RTE trace filters an unsupported pair before assigning two legal predicates to the retained one; the ED trace keeps a discovered mention separate from the event type recorded by the dataset.

Table 6: Stage-wise traces for ASQP, RTE, and ED.

| | |
|---|---|
| **(a) ASQP: Rest15 – boundary selection before labeling** | |
| **Source** | "The entree was also very good." |
| **Discovery** | Natural pair: (entree, very good). |
| **Projection** | Grounded pair: (entree, good); the optional degree word is removed. |
| **Compilation** | (entree, good, food quality, positive). |
| **(b) RTE: NYT – one connection, two predicates** | |
| **Source excerpt** | "Bobby Fischer ... at a news conference in Reykjavik, Iceland." |
| **Discovery** | Natural connections include (Reykjavik, Iceland) and the conference-presence pair (Bobby Fischer, Iceland). |
| **Projection** | KEEP Reykjavik–Iceland; DROP Bobby Fischer–Iceland because presence at the conference does not establish the precedent's nationality relation. |
| **Compilation** | (Iceland, `/location/country/capital`, Reykjavik); (Iceland, `/location/location/contains`, Reykjavik). |
| **(c) ED: ACE2005 – mention discovery versus recordable event** | |
| **Source excerpt** | "have already been sued ... and they face investigations" |
| **Discovery** | Two natural event frames: "sued" and "investigations". |
| **Projection** | Offers `sue`/"sued" for the first frame; no legal event option for the second. |
| **Compilation** | EMIT `sue`/"sued"; select NONE for "investigations". No argument roles are predicted. |

# D PROMPT TEMPLATES AND RELATION FAMILIES

The prompt boxes use the same format throughout: supplied fields, model instruction, and returned state. Bracketed fields are populated at inference time; per-instance JSON serialization is omitted. The templates show the default setting $K = 5$.

## D.1 SEMANTIC DISCOVERY

Discovery prompts ask for source-grounded natural relations or events, without final task labels. NYT uses six fixed semantic questions; WebNLG maps predicate names and descriptions from the legal ontology to thirteen question templates. Table 7 summarizes these relation families, which may overlap; NYT paths are shortened to their final label name. Stage 1 receives the natural-language question cards, not the mapped predicate labels.

**ASQP (Rest15 / Rest16)**
**Given:** [restaurant review]; for the atomic reading, also [source-only coverage scaffold].
**Instruction:** First read the review as a whole to find every evaluated thing and evaluative expression, including implicit overall attitudes and return or avoidance intentions. Then inspect each coverage unit for local, coordinated, negated, and shared evaluations. Quote the smallest complete source expressions; use NULL for an implicit evaluated thing. Do not use ASQP categories or polarity labels.
**Return:** `relations: [{thing, expression}].`

**RTE (NYT)**
**Given:** [sentence] and [natural relation-family cards], each with a title, discovery question, inclusion guidance, and exclusion guidance.
**Instruction:** Read the sentence under each supplied family question. For every meaningful connection, copy its two named endpoints from the source, explain the connection in ordinary language, and quote source evidence. Do not predict a NYT predicate.

**RTE (NYT) (continued)**
**Return:** `associations: [{left, right, connection, evidence, family_id, support}].`

**RTE (WebNLG)**
**Given:** [sentence] and [applicable natural relation-family cards] with the same fields as NYT.
**Instruction:** Examine the complete sentence through each supplied question. Copy grounded endpoint spans, describe their connection in ordinary language, and provide source evidence. The cards guide discovery; the mapped WebNLG predicate names are not shown in this prompt.
**Return:** `associations: [{left, right, connection, evidence, family_id, support}].`

**ED (ACE2005)**
**Given:** [sentence], [indexed source tokens], and [six broad event-family cards].
**Instruction:** Find each lexical event mention, including event verbs or nouns that are embedded or coordinated. Describe what happened and identify the smallest source-token anchor; do not extract arguments or assign a fine ACE event type.
**Return:** `frames: [{family_id, what_happened, anchor_options, evidence}].`

Table 7: NYT and WebNLG relation families with representative mapped labels.

**(a) NYT: six relation families**

| Natural family | Representative mapped labels (family size) |
|---|---|
| Identity / classification | nationality, industry (7) |
| Spatial location | contains, place_lived (12) |
| Part / whole / control | major_shareholders, contains (3) |
| Role / affiliation | company, advisors (9) |
| Creation / production | founders, place_founded (2) |
| Biography / social | children, place_of_birth (3) |

**(b) WebNLG: thirteen relation families**

| Natural family | Representative mapped labels (family size) |
|---|---|
| Identity / classification | language, ethnicGroup, demonym (33) |
| Spatial location | country, capital, birthPlace (39) |
| Part / whole / control | isPartOf, owner, operator (12) |
| Role / affiliation | club, leader, operatingOrganisation (31) |
| Creation / production | creator, author, architect (13) |
| Biography / social | birthPlace, deathPlace, awards (17) |
| Event / participation | season, champions, starring (10) |
| Time / order | established, precededBy, completionDate (11) |
| Measurement / identifier | runwayLength, floorCount, LCCN_number (39) |
| Composition / property | ingredient, currency, floorArea (23) |
| Transport / route | cityServed, operator, aircraftFighter (19) |
| Work / topic / content | officialLanguage, series, firstAppearanceInFilm (7) |
| Other recordable attribute | product, keyPerson, sportsOffered (19) |

### D.2 Candidate Projection

Projection takes the frozen Discovery state and dataset-specific legal choices or annotation precedents as input. Its returned state remains available to set-level compilation. For NYT and WebNLG, a candidate with no precedent-supported choice is not queried and is not passed forward.

**ASQP (Rest15 / Rest16)**
**Given:** [review], [one discovered natural evaluation], [legal aspect/opinion source spans], [pair-boundary precedents].
**Instruction:** The evaluation relation already exists, but its description strings are provisional. Interpret the complete evaluated-thing–expression meaning, then rewrite it as exactly one aspect–opinion pair under the annotation standard. Copy one legal source span for each endpoint. Prefer the shortest complete expression;

**ASQP (Rest15 / Rest16) (continued)**
preserve proper names, compounds, negation, particles, and indispensable complements. Do not reject, add, split, merge, or label the relation.
**Return:** `grounded:{id, aspect, opinion}`.

**RTE (NYT)**
**Given:** [sentence], [one frozen natural endpoint connection], [finite supported predicate/direction choices], [matching annotations in five retrieved training sentences].
**Instruction:** Examine the source first. Compare the candidate's exact endpoints, relation meaning, direction, and source evidence with each annotated precedent. Topic or entity-type overlap alone is insufficient. KEEP only when a shown precedent supports an applicable annotation choice; otherwise DROP. Do not decide the final triple set or assign its predicate here.
**Return:** `{decision: KEEP|DROP, support_precedent_id: ID|null, reason}`.

**RTE (WebNLG)**
**Given:** [sentence], [one frozen natural endpoint connection], [WebNLG predicate/direction choices], [matching annotations in five retrieved WebNLG training sentences].
**Instruction:** Apply the same source-first, exact-endpoint and relation-meaning comparison as for NYT. Cite an applicable annotated precedent to KEEP; otherwise DROP. Similar topic or entity types are not support. Do not create a choice outside the finite board or select the final triple set.
**Return:** `{decision: KEEP|DROP, support_precedent_id: ID|null, reason}`.

**ED (ACE2005)**
**Given:** [indexed sentence], [all natural event frames], [public event-type definitions], [five typed-trigger precedents].
**Instruction:** For each frame, propose source-grounded event-type–trigger options for that same event. The Discovery anchor is a clue, not a fixed trigger boundary. Prefer the minimal lexical event head, retain genuine same-event alternatives, and exclude topical matches. Do not select the final event set or predict arguments or roles.
**Return:** each `frame_id` with zero or more `event_options` and source-token spans.

### D.3 JOINT COMPILATION

Compilation receives the complete projected state for one input and returns the task record set. Its prompt is organized by task below. For ASQP, the legal semantic axes are mapped to each dataset's category inventory after the call.

**ASQP (Rest15 / Rest16)**
**Given:** [review], [all frozen aspect–opinion pairs], [legal target/focus axes and sentiment values], [five annotated reviews].
**Instruction:** Classify every supplied pair exactly once. What kind of thing is evaluated? What property of it is evaluated? What sentiment does its expression convey? Infer an implicit target from the full review and preserve local negation. Treat examples as annotation evidence. Do not add, delete, split, merge, or rewrite a pair.
**Return:** one `compiled` item per `pair_id`, with `target_kind`, `focus`, and `sentiment`.

**RTE (NYT)**
**Given:** [sentence], [complete projected candidate board], [globally unique choice IDs and readable predicate cards], [the five training examples bound in Stage 2].
**Instruction:** Jointly select every supplied choice that faithfully realizes a discovered meaning. Check predicate, direction, endpoint identity, and source evidence. Distinct predicates on the same endpoints may coexist. Omit genuine meaning or endpoint mismatches; do not import a training fact or invent an endpoint, predicate, relation, or choice ID.
**Return:** `{selected_choice_ids: [ID, ...]}`, possibly empty.

**RTE (WebNLG)**
**Given:** [sentence], [complete projected candidate board], [WebNLG choice IDs and readable predicate cards], [the five training examples bound in Stage 2].
**Instruction:** Make the same source-grounded set decision as for NYT. Keep each candidate relation visible while selecting zero or more legal predicate/direction choices for it. A choice is not a fact merely because it appeared in a training example; reject mismatched endpoints, direction, predicate, or evidence.

**RTE (WebNLG) (continued)**
**Return:** one `relation_id` decision per candidate, each with a possibly empty `selected_choice_ids` list.

**ED (ACE2005)**
**Given:** [sentence], [all projected event frames and their type–trigger option IDs], [the same five trigger precedents].
**Instruction:** For every frame, choose EMIT with one supplied choice ID or NONE. Compare overlapping frames jointly and retain distinct event mentions without a fixed count. Do not invent a trigger, event type, argument, or role.
**Return:** an EMIT row with `event_choice_id`, or a NONE row with only the frame ID and decision.

# E Evidence Selection and Cost Accounting

## E.1 Input-Level Example Selection

For ASQP and NYT, the budget $K$ counts distinct complete annotated examples supplied to Joint Compilation for one input, rather than records independently retrieved for every candidate. Candidate-local boundary evidence in Projection is distinct from this budget. The Rest15 sensitivity experiment keeps projected boundaries fixed and varies compilation examples; NYT shares the selected bank across Projection and Compilation, so both modules are rerun when $K$ changes.

For ASQP compilation, each training sentence $b$ receives the fixed score

$$J_b + 4A_b + 5O_b + 8E_b^a + 10E_b^o + 0.75U_b - 0.1|M - H_b|. \tag{8}$$

Here, $J_b$ is the token-set Jaccard similarity between input and training sentence; $A_b$ and $O_b$ are the maximum Jaccard similarities between input candidates and training records for aspect and opinion token sets, respectively. $E_b^a$ and $E_b^o$ indicate whether any corresponding endpoint sets are identical, and $U_b$ indicates agreement on the presence of an implicit target. $M$ and $H_b$ count input candidates and training records. Sentences are sorted by descending score, with training index breaking ties, and the first $K$ are selected.

NYT selects a single input-level bank using candidate-linked record coverage. It greedily adds the training sentence that covers the largest number of still-uncovered candidate-specific predicate/direction choices. Lexical relevance breaks ties, followed by training index. Remaining slots are filled by lexical ranking and then training order, excluding the input text and already selected examples. The selected bank retains complete annotations together with their source context.

## E.2 Recorded Inference Costs

Table 3 evaluates each method on the same 100 inputs per dataset. Input and output tokens are means per input. Time is the mean recorded request duration per input, summing requests across stages and subcalls; it is not parallel wall-clock latency or a controlled throughput measurement. Tok./F1 divides the unrounded mean total tokens by micro-F1 on those inputs, expressed on a 0–100 scale. It is a descriptive ratio, not the marginal token cost of increasing F1.

Token accounting excludes provider usage that was not returned. Rest15 SDA includes usage measured by replaying five reciprocal subcalls, while retaining the original predictions and recorded durations; an unreported rejected request remains excluded. NYT SDA excludes usage from one refused compilation request, so its token counts and Tok./F1 are lower bounds. Its request-duration records are complete. These conventions apply only to cost accounting, not to the evaluation denominator.