ENG-DRB: PDTB-style Discourse Relation Bank on Engineering Tutorial Video Scripts

Cheng Zhang[1], Rajasekhar Kakarla[1], Kangda Wei[2], Ruihong Huang[2]
[1]Department of Construction Science and Organizational Leadership,
Purdue University Northwest, Hammond, IN
[2]Department of Computer Science and Engineering,
Texas A&M University, College Station, TX
{zhan4168, kakarlar}@pnw.edu, {kangda, huangrh}@tamu.edu

Abstract

Discourse relation parsing plays a crucial role in uncovering the logical structure of text, yet existing corpora focus almost exclusively on general-domain genres, leaving specialized fields like engineering under-resourced. We introduce ENG-DRB, the first PDTB-style discourse relation corpus derived from transcripts of hands-on engineering tutorial videos. ENG-DRB comprises 11 tutorials spanning civil, mechanical, and electrical/electronics engineering (155 minutes total) with 1,215 annotated relations. Compared to general-domain benchmarks, this dataset features a high proportion of explicit senses, dense causal and temporal relations, and frequent overlapping and embedded senses. Our benchmarking experiments underscore the dataset's difficulty. A top parser (HITS) detects segment boundaries well (98.6% F1), but its relation classification is more than 11 F1 percentages lower than on the standard PDTB. In addition, state-of-the-art LLMs (OpenAI o4-mini, Claude 3.7, LLaMA-3.1) achieve at best 41% F1 on explicit relations and less than 9% F1 on implicit relations, revealing systematic errors in temporal and causal sense detection. The dataset can be accessed at: https://doi.org/10.57967/hf/6895. Code to reproduce our results is available at: https://github.com/chengzhangedu/ENG-DRB

1 Introduction

Discourse relation parsing is a core NLP task for understanding the logical or rhetorical relations between textual units. However, state-of-the-art models trained on general-domain corpora like the Penn Discourse Treebank (PDTB) (Prasad et al., 2019) exhibit a significant performance drop when applied to specialized domains. While domain-specific corpora exist for biomedicine (e.g., Bio-DRB (Prasad et al., 2011) and BioDCA (Gopalan and Lalitha Devi, 2016)), the engineering domain remains underexplored. Engineering discourse presents a unique challenge, characterized by a high density of procedural steps, causal chains, and nested justifications that are poorly represented in existing datasets.

To address this gap, we introduce ENG-DRB, the first PDTB-style discourse relation corpus for the engineering domain. We source our data from a uniquely rich and authentic context: transcripts of hands-on engineering tutorial videos. This captures the dynamic, verbalized thought processes of instructors, providing explicit links between actions, preconditions, and rationale. The resulting corpus contains 1,215 annotated discourse relations across civil, mechanical, and electrical engineering, with a high proportion of explicit connectives (63.1%) that ground the procedural and causal phenomena as demonstrated by the example in Figure 1, at significantly higher rates than in general-domain corpora such as the PDTB2.

A key contribution of ENG-DRB is that it provides a new benchmark for complex, naturally-occurring discourse. ENG-DRB introduces a challenging benchmark whose difficulty mainly arises from two sources. First, as speech transcripts, the text is more informal and conversational than edited instructional texts. Second, the engineering content features frequent overlapping senses (one or more spans participate in multiple relations) and embedded senses (one relation is hierarchically nested entirely within the argument of another relation) that explain procedural steps. Our experiments confirm this dual challenge. A top parser (Liu et al., 2023) detects segment boundaries well (98.6 F1), but its relation classification drops by 11+ points compared to its performance on the standard PDTB 3.0 corpus. We also find that even state-of-the-art Large Language Models (LLMs) struggle with the domain complexity and long-range dependencies inherent in specialized text (Chen et al., 2024; Cheng et al., 2025).

Our contributions are:

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1318–1330
December 20-24, 2025 ©2025 Association for Computational Linguistics

- We introduce **ENG-DRB**, the first PDTB-style discourse corpus built from hands-on engineering tutorial video scripts.
- We provide a detailed analysis of the corpus, highlighting the unique discourse phenomena present in the engineering domain.
- We benchmark HITS model and modern LLMs on ENG-DRB, demonstrating their current limitations and establishing a new challenge for domain-specific discourse parsing.

2 Related Work

The Penn Discourse TreeBank (PDTB) provides a comprehensive inventory of relations for general-domain text (Prasad et al., 2019). Its framework has been shown to benefit numerous downstream applications, including summarization and machine translation (Cohan et al., 2018). The success of the PDTB framework has inspired adaptations for many other languages, including Chinese (Zhou and Xue, 2012), Arabic (Alsaif et al., 2018), and Turkish (Zeyrek and Kurfalı, 2018).

In parallel with these cross-lingual news-wire efforts, the PDTB framework has been extended to a variety of non-news genres. A pilot PDTB-style annotation of Twitter conversations showed that social-media arguments often fall outside full clauses and that Expansion and Contingency relations dominate (Scheffler et al., 2019). The TED Multilingual Discourse Bank (TED-MDB) applied PDTB-3.0 to prepared public speeches in six languages of TED Talks (Zeyrek et al., 2020). The GDTB presented a 16-genre benchmark for PDTB-style shallow discourse parsing (Liu et al., 2024). More recently, the DISRPT2023 shared task produced three multi-genre or conversational PDTB-style corpora—Italian LUNA (spoken dialogue), Portuguese CRPC (news, fiction, scientific) and Turkish TDB—which each adapted the scheme to dialogue units or mixed-genre texts (Braud et al., 2023).

On the domain-specific front, the Biomedical Discourse Relation Bank (BioDRB) extends the PDTB framework into the biomedical domain by refining its sense hierarchy to include relations such as Evidence and Hypothesis (Prasad et al., 2011). This dataset contains 5830 annotated senses. Other PDTB-style biomedical corpus exist, such as BioDCA (Gopalan and Lalitha Devi, 2016). Yet Riccardi (Stepanov and Riccardi, 2014) showed that even when PDTB-trained discourse parsers are applied to BioDRB, they still reflect strong domain specificity: cross-domain transfers demand careful adaptation. Despite these advances in broadening both genre and domain coverage, no PDTB-style corpus has been created for engineering, and the distinctive discourse of procedural, hands-on instructional texts in built-environment engineering remains completely uncharted.

Significant recent research has focused on evaluating the capacity of Large Language Models (LLMs) in handling discourse reasoning tasks, including the identification of discourse relations and the interpretation of logical connections (Chen et al., 2024). Evaluations show that even advanced LLMs struggle with discourse relation extraction (Wei et al., 2024), their performance diminishes with increased domain complexity, longer contexts, and tasks involving intricate logical, deductive, inductive, or abductive reasoning (Cheng et al., 2025; Lin et al., 2025; Li et al., 2024). The effectiveness of LLMs in specialized engineering contexts remains largely unexplored.

3 Data Collection and Pre-processing

3.1 Video Selection

We carefully select engineering tutorial videos from YouTube to ensure the quality and consistency necessary for effective discourse annotation. The selected items are hands-on "how-to / troubleshooting / installation" tutorials rather than theoretical lectures (e.g., water main break repair, pouring concrete footings, furnace troubleshooting). Videos represent three major engineering subdomains—civil, mechanical, and electrical engineering—chosen for their distinct problem-solving and reasoning methods. Specifically, civil engineering tutorials emphasize spatial reasoning, mechanical engineering videos illustrate procedural troubleshooting, and electrical engineering tutorials highlight diagnostic reasoning.

We manually identify the videos to annotate to ensure a high quality of selected videos: they had to clearly explain engineering concepts or engineering procedures, presented predominantly in monologue form. Monologue-oriented tutorials minimize dialogue fragmentation, facilitating clear identification of discourse relations and logical coherence. We also considered audience engagement metrics, including like-to-dislike ratios higher than 95% and overwhelmingly positive user comments,

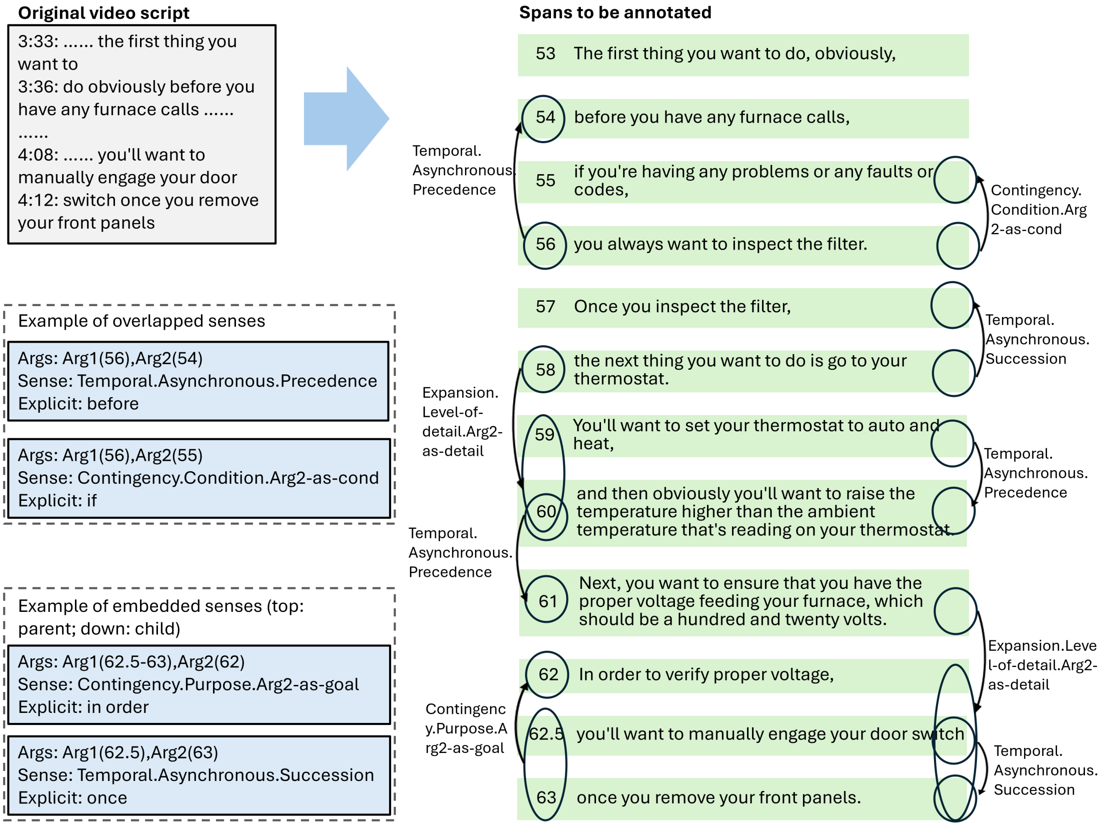
Original video script
3:33: ...... the first thing you want to
3:36: do obviously before you have any furnace calls ......
......
4:08: ...... you'll want to manually engage your door
4:12: switch once you remove your front panels
Spans to be annotated
53 The first thing you want to do, obviously,
54 before you have any furnace calls,
55 if you're having any problems or any faults or codes,
56 you always want to inspect the filter.
57 Once you inspect the filter,
58 the next thing you want to do is go to your thermostat.
59 You'll want to set your thermostat to auto and heat,
60 and then obviously you'll want to raise the temperature higher than the ambient temperature that's reading on your thermostat.
61 Next, you want to ensure that you have the proper voltage feeding your furnace, which should be a hundred and twenty volts.
62 In order to verify proper voltage,
62.5 you'll want to manually engage your door switch
63 once you remove your front panels.
Temporal. Asynchronous Precedence
Contingency. Condition.Arg 2-as-cond
Temporal. Asynchronous. Succession
Expansion. Level-of-detail.Arg2-as-detail
Temporal. Asynchronous. Precedence
Temporal. Asynchronous. Precedence
Expansion.Level-of-detail.Arg2-as-detail
Contingency.Purpose.Arg2-as-goal
Temporal. Asynchronous. Succession
Example of overlapped senses
Args: Arg1(56),Arg2(54)
Sense: Temporal.Asynchronous.Precedence
Explicit: before
Args: Arg1(56),Arg2(55)
Sense: Contingency.Condition.Arg2-as-cond
Explicit: if
Example of embedded senses (top: parent; down: child)
Args: Arg1(62.5-63),Arg2(62)
Sense: Contingency.Purpose.Arg2-as-goal
Explicit: in order
Args: Arg1(62.5),Arg2(63)
Sense: Temporal.Asynchronous.Succession
Explicit: once

Figure 1: An excerpt of annotated hands-on engineering tutorials. Decimal sub-span labels (e.g.,62.5) mark additional clause-level splits made whenever annotators detect discourse relations left un-separated in the initial segmentation. Examples of overlapping and embedded senses are also visualized here. This example highlights the features of this dataset: richness of temporal/causal relations, overlapping and embedded senses.

to assess each video's credibility, clarity and educational value. Additionally, we only included videos with clear audio, thereby enhancing transcription reliability. More metadata about selected videos (e.g., video links) are in Appendix A. Ultimately, our dataset comprises the scripts of 11 videos, totaling roughly 155 minutes of instructional content.

3.2 Data Pre-processing

We converted automatically generated YouTube transcripts into clause-level spans suitable for PDTB-style discourse annotation using a three-stage pipeline: (1) automatic cleanup, (2) LLM-based candidate segmentation, and (3) human editing with final verification.

1) Automatic cleanup. We used ChatGPT[1] to insert punctuation and remove timestamps and boilerplate (e.g., "Thank you for watching, please subscribe!"), while preserving technical content. This pass improved readability but was not sufficient for complete accuracy.

2) Candidate EDU segmentation. Guided by a detailed prompt, ChatGPT[1] split sentences into elementary discourse units (EDUs) aligned with the PDTB-3 sense inventory. The prompt instructed the model to (i) isolate clauses linked by potential explicit or implicit relations (temporal, contingency, comparison, expansion), (ii) use connectives and punctuation as likely boundaries, and (iii) avoid over-segmentation of closely related elaborations. Multiple examples and edge cases were included to calibrate granularity.

3) Human editing and ID scheme. Annotators treated LLM output as editable scaffolding: they freely split or relocated boundaries. Each span received a unique integer ID; when additional relations were identified within a span, annotators created decimal sub-spans (e.g., 62.1, 62.2). We did not encode an explicit hierarchy and "embedded" structures arise when one relation's arguments subsume another pair.

Bias mitigation and consistency. Because boundaries were under annotator control, the ma-

[1] ChatGPT o1-mini and then O4-mini were used because data collection and preprocessing spanned more than six months, during which GPT discontinued o1-mini.

chine proposals did not constrain the final segmentation. To assess systematicity and guard against LLM artifacts, we applied a top non-LLM segmenter (HITS) to the finalized corpus and observed near-perfect boundary/connective detection (98.6 F1, §6), indicating that the corpus reflects consistent segmentation rather than overfitting to LLM outputs.

Final verification and corpus. The authors manually reviewed transcripts against the original videos, correcting residual errors, validating technical content, and removing non-essential segments (e.g., greetings/closings). The resulting dataset contains 2,259 annotated spans drawn from 11 videos.

4 Annotation Scheme and Process

4.1 Annotation Scheme

Our annotation scheme adapts the Penn Discourse TreeBank 3.0 (PDTB-3) (Prasad et al., 2018) to the characteristics of engineering discourse while reducing annotation ambiguity. Concretely, we omit the *SpeechAct* attribute layer and drop or modify a small set of senses in the Expansion category with limited utility for procedural texts; we retain all other PDTB-3 sense definitions and guidelines.

SpeechAct. We do not distinguish *SpeechAct* senses. These encode meta-communicative intentions (e.g., advice, question) that are peripheral to modeling logical and procedural relations in instructional scripts and add complexity without analytic gain.

Expansion.Conjunction / Disjunction. We omit *Expansion.Conjunction* and *Expansion.Disjunction*. As noted in PDTB-3, these senses often mark arguments that stand in parallel to an external situation rather than directly to each other. In engineering tutorials, such patterns typically surface as inventories or enumerations (e.g., options, pros/cons, multiple reasons) and contribute little to causal or procedural interpretation. For example, in "Span 1: we will do process A for two reasons. Span 2: the first reason is … Span 3: the second reason is …," PDTB-3 would tag Spans 2 and 3 as *Expansion.Conjunction*. Instead, we capture the underlying reasoning as a single causal relation with a multi-span argument: *Arg1* = Span 1; *Arg2* = {*Span 2*, *Span 3*}; *Sense* = *Contingency.Cause.Reason*. This represents the reasons as jointly supporting Span 1, avoiding a parallelism tag that adds little procedural/causal signal in this domain.

Expansion.Manner. PDTB-3 notes frequent overlap between *Expansion.Manner* and *Contingency.Purpose*. In our corpus, "manner/means" expressions in procedural language almost always paraphrase a teleological goal (e.g., "Do X *to* achieve Y"), which led to annotator confusion with limited payoff. We therefore collapse *Expansion.Manner* into *Contingency.Purpose*.

Expansion.Equivalence. We found *Expansion.Equivalence* difficult to operationalize because it often coincides with differences in technical specificity. Rather than retain it as a distinct label, we annotate such cases as *Expansion.Level-of-detail* to foreground specificity contrasts (e.g., "Arg1: but it's just laziness." vs. "Arg2: Somebody didn't want to actually cut pieces of wire and do it right." → *Expansion.Level-of-detail.Arg2-as-detail*).

These targeted modifications keep our scheme faithful to PDTB-3 while aligning labels with the reasoning structures prevalent in engineering tutorial videos.

4.2 Annotation process

The annotation process consisted of three main phases: annotator training, independent annotation with span adjustment, and consensus-based adjudication.

4.2.1 Annotator Training

Both annotators received comprehensive training on the three-level PDTB-3 taxonomy, covering the top-level sense categories and the two subsequent levels of fine-grained types and subtypes. The annotators will carefully read the examples to ensure solid understanding of the PDTB-3 taxonomy. Training combined conceptual instruction with practical exercises in annotation tasks on separate tutorial video scripts, that are not included in the final dataset. Following PDTB guidelines, annotators are trained to mark only the minimal and sufficient spans to interpret the relations. Special emphasis was placed on selecting minimal spans—segments sufficient to unambiguously convey a discourse sense within its context.

4.2.2 Span Refinement and Sense Annotation

Following training, annotators independently reviewed candidate spans. Annotators were encouraged to adjust segment boundaries and further subdivide spans as needed to accurately capture discourse structures. For example, when a span contained both a general statement and a causal explanation, annotators split it at the explicit connective ("because") to ensure each span corresponded to a single, coherent discourse relation. New sub-spans were assigned decimal-based IDs to maintain traceability (e.g., span no. 62 and 62.5 in Figure 1).

After finalizing span boundaries, annotators assigned sense labels drawn from the modified PDTB-3 sense taxonomy. A minor modification to our annotation task, compared to PDTB-3, is the removal of the constraint restricting implicit senses to adjacent sentences. This adjustment is necessary because the original video transcripts lack punctuation, making sentence boundaries ambiguous and thus unreliable for identifying adjacency. Consequently, we permit annotation of implicit relations whose arguments are non-adjacent (i.e., separated by at least one intervening span). Such non-adjacent implicit relations frequently occur in engineering documentation, especially when detailed explanations or rationales interrupt sequences of actions, resulting in temporal connections spanning multiple sentences. Neglecting these relations would limit a comprehensive analysis of logical and sequential structures in engineering processes. About 11% of implicit relations in our dataset involve arguments spanning non-adjacent segments.

4.2.3 Annotation Adjudication

To reduce superficial consensus and ensure quality, we introduce a two-round adjudication protocol. In Round 1, Annotator 1 (A1) shares their annotations with Annotator 2 (A2). A2 reviews each instance and records either "agree" (no substantive difference; e.g., typos or initial misunderstandings) or "disagree" (a genuine interpretive difference), providing a brief rationale for every "disagree." A1 then reviews these cases and, for any remaining interpretive disagreements, adds a response and rationale. Most cases are resolved at this stage. In Round 2, unresolved items are discussed synchronously to reach final consensus.

4.3 Inter-Annotator Agreement Metrics

We evaluated inter-annotator agreement at the sense level using both strict and partial criteria. For strict agreement, we treated each annotated span pair as a discrete instance and computed precision, recall and F1 (Brants, 2000), omitting Cohen's Kappa since the vast number of true negatives in span-selection renders it uninformative.[2] We evaluate inter-annotator agreement using two metrics: strict and partial matching.

	Prec.	Recall	F1
Strict	0.74	0.70	0.72
Partial	0.78	0.74	0.76

Table 1: Inter-annotator agreement scores between two annotators.

Strict Match: A true positive is defined as a case where both annotators identify the identical character spans for both Argument 1 and Argument 2, and assign the same Level-3 sense label.

Partial Match: To capture agreement where annotators agree on the substance of an argument but differ slightly on its boundaries, we employ a partial-matching criterion. We quantify the degree of span overlap using a score, *P*, which is a micro-averaged Jaccard index over the argument spans:

$$P = \frac{|\mathcal{A}_1^{\text{Arg1}} \cap \mathcal{A}_2^{\text{Arg1}}| + |\mathcal{A}_1^{\text{Arg2}} \cap \mathcal{A}_2^{\text{Arg2}}|}{|\mathcal{A}_1^{\text{Arg1}} \cup \mathcal{A}_2^{\text{Arg1}} \cup \mathcal{A}_1^{\text{Arg2}} \cup \mathcal{A}_2^{\text{Arg2}}|}$$

For an annotation pair to be considered a true positive under this scheme, it must have an identical Level-3 sense label and an overlap score of $P \geq 0.5$.

As summarized in Table 1, the strict F1 of 0.72 indicates moderate agreement given the technical and subjective nature of span boundaries, while the higher partial F1 of 0.76 shows that a small proportion discrepancies arise from minor boundary shifts rather than substantive sense differences.

5 Statistics of the ENG-DRB Dataset

Our dataset is constructed from 11 engineering tutorial videos covering domains including civil, mechanical, and electrical/electronics engineering. Each video's script was thoroughly annotated for a range of discourse features, including implicit and explicit relations, non-adjacent spans, long-distance relations, and alternative lexicalizations (AltLex).

5.1 Overview

The ENG-DRB dataset contains 11 engineering tutorials, totaling 155 minutes of instructional video,

[2] For a document of length N, the number of ordered pairs of non-overlapping contiguous segments is $T(N) = \frac{N(N+1)(N+2)(N-1)}{12}$, e.g., $T(100) = 8,499,150$

Domain	No.	Duration	Tokens	Spans
C	4	57:10	9021	864
E/E	3	53:07	8218	763
M	4	45:15	6362	632
Total	**11**	**155:32**	**23601**	**2259**

Table 2: Per-domain summary of number of videos (No.), duration, token count and spans. C, M, E/E refers to Civil, Mechanical, and Electrical/Electronics, respectively

from which we annotated 2,259 discourse spans and 1,215 senses. The corpus is organized into three domains—Civil (C), Electrical/Electronics (E/E), and Mechanical (M)—with a detailed breakdown provided in Table 2.

Despite differences in topic and length, the data reveals remarkable structural consistency across the engineering domains. This is evident in two key metrics. First, the density of discourse annotation is stable, ranging from 14.0 to 15.1 spans per minute. Second, the proportion of spans that form a discourse relation is also highly consistent, with 52% to 56% of spans having an annotated sense across the three domains (calculated from Tables 2 and 3). This consistency in both span density and relational structure strengthens the dataset's utility for building generalizable models of instructional discourse.

The distribution of discourse relation types is shown in Table 3. Explicit relations are predominant, constituting 767 instances (63.1%) of the total. Implicit relations account for 393 instances (32.3%), while AltLex relations[3] are the least frequent, with 55 occurrences (4.5%). This strong preference for explicit markers suggests that technical instructional discourse in engineering prioritizes the unambiguous signaling of procedures and causal dependencies.

5.2 Distribution of Senses at the First and Second Levels of the PDTB-3 Hierarchy

Table 4 presents the distribution of annotated discourse relations. Of the annotated discourse senses, **Contingency** relations are the most frequent, constituting 50.2% of the total. Following are **Temporal** (21.5%), **Expansion** (18%), and **Comparison** (10%) relations. This skew underscores the prominence of causal explanation and stepwise sequencing characteristic of the corpus's instructional content. In contrast, news corpora such as PDTB-3 exhibit a more balanced distribution with a greater emphasis on **Expansion** relations for event and entity elaboration.

Domain	Exp.	Imp.	AltLex	Total
C	299	130	21	450
E/E	267	141	19	427
M	201	122	15	338
Total	767	393	55	1215

Table 3: Distribution of discourse relations in the ENG-DRB dataset by domain. Exp., Imp., C, M, and E/E refer to Explicit, Implicit, Civil, Mechanical, and Electrical/Electronics, respectively

2nd-level Senses	No.	%
Temporal.Asynchronous	232	19.09%
Temporal.Synchronous	29	2.39%
Contingency.Cause	395	32.51%
Contingency.Purpose	48	3.95%
Contingency.Condition	158	13%
Contingency.Negative-condi	9	0.74%
Comparison.Concession	109	8.97%
Comparison.Contrast	10	0.82%
Comparison.Similarity	2	0.16%
Expansion.Exception	0	0%
Expansion.Instantiation	6	0.49%
Expansion.Level-of-detail	204	16.79%
Expansion.Substitution	13	1.07%

Table 4: Distribution of Senses in Discourse Relations

At the second level, the **Cause** relation, a subtype of Contingency, is the most prominent, accounting for 32.5% of all annotated senses. Other significant senses include **Asynchronous** (19.1%) and **Level-of-detail** (16.8%). Relations like **Condition** (13.0%), **Concession** (9.0%), and **Purpose** (4.0%) are also noteworthy. Conversely, senses such as *Negative-condition*, *Contrast*, *Similarity*, *Instantiation*, and *Substitution* appear infrequently. These findings suggest that engineering tutorials rely heavily on causal, temporal, and elaborative discourse strategies.

[3] AltLex refers to cases where an implicit discourse relation is already lexicalized by alternative expressions, making any added connective redundant.

5.3 Overlapped and Embedded Senses in ENG-DRB: Compared with the PDTB-2 Dataset

In addition to the basic discourse structure, ENG-DRB includes annotations for two complex phenomena: overlapping and embedded senses. Overlapping senses occur when two or more discourse relations share one or more common text spans, while embedded senses refer to hierarchical structures where one discourse relation (child) is entirely contained within a single argument of another (parent).

Analysis reveals that ENG-DRB exhibits significantly higher complexity compared to the PDTB-2 dataset in both phenomena. Specifically, ENG-DRB annotations have, on average, 2.05 overlapping senses per annotated sense, compared to 1.36 in PDTB-2. At the 95th percentile, ENG-DRB annotations include up to five overlapping senses per relation, substantially exceeding the maximum of three overlaps per relation observed in PDTB-2.

Similarly, embedded senses are also notably more frequent and complex in ENG-DRB. On average, each sense in ENG-DRB participates in 0.65 parent-child relations, with 31% of senses functioning as children and 43% as parents. By contrast, PDTB-2 shows fewer embedded structures, averaging only 0.29 parent-child relationships per sense, with only 22% of senses functioning as either parents or children.

These findings highlight the distinctive complexity of ENG-DRB. The high density of overlapping and embedded annotations enables the dataset to explicitly represent hierarchical discourse structures, traditionally a characteristic feature of the Rhetorical Structure Theory (RST) framework rather than PDTB. This augmentation allows PDTB-based annotations in ENG-DRB to capture richer structural information, thus bridging the gap between shallow discourse annotation frameworks like PDTB and deeper, hierarchical frameworks exemplified by RST.

6 Benchmarking the HITS model on ENG-DRB

We benchmark a strong, open-source baseline by replicating the HITS system from the DISRPT 2023 shared task for discourse segmentation, connective detection, and relation classification (Liu et al., 2023). Following DISRPT, we evaluate (i) *Discourse Segmentation + Connective Detection* (Tasks 1&2; F1), and (ii) *Discourse Relation Classification* (Task 3; accuracy). We use our standard split into 8/1/3 documents for train/dev/test, respectively, with 1,487, 116, and 707 labeled instances. For Tasks 1&2, HITS employs a BiLSTM+CRF sequence tagger on top of a RoBERTa encoder. For Task 3, it uses a RoBERTa-based classifier that predicts the discourse relation between adjacent units.

All runs use a single NVIDIA H100 (80 GB) with 64 GB RAM and seed `106524`. *Tasks 1&2:* `roberta-base`, max sequence length 512, train/eval batch sizes 16/32, learning rate 3×10^{-5}, dropout 0.1, 10 epochs, warmup ratio 0.06, weight decay 0.1, max gradient norm 2.0. *Task 3:* `roberta-base`, max sequence length 512, batch sizes 16/32, learning rate 2×10^{-5}, dropout 0.1, 5 epochs, warmup ratio 0.06, weight decay 0.1.

HITS reaches near-ceiling performance on segmentation and connective detection (98.62 F1 on the test set), indicating that identifying discourse unit boundaries and connectives is *not* the bottleneck in ENG-DRB. In contrast, relation classification is markedly harder: despite a strong 80.70% development accuracy, the corresponding test accuracy is 63.73%, a ∼17-point generalization gap. Relative to the system's reported accuracy on the standard `eng.pdtb.pdtb` benchmark (74.75%), ENG-DRB yields an ∼11-point drop, underscoring the domain's difficulty.

7 Benchmarking LLMs on ENG-DRB dataset

7.1 Experimental Setup

For benchmarking the LLMs on the ENG-DRB dataset, we chose three LLMs: (1) OpenAI o4-mini-2025-04-16 (OpenAI et al., 2024), (2) Claude-3-7-sonnet-20250219 anthropic2024claude3, and (3)LLAMA-3.1-8B (Grattafiori et al., 2024) [4]. We separated the implicit and explicit senses in the golden dataset and conduct separate experiments.

We evaluate LLM annotation by sliding a fixed-size window of 20 spans over each document's annotated spans, moving it forward by 10 spans at each step (so that consecutive windows overlap by 10 spans). Because 97% of all arguments in our dataset are 10 spans or shorter, this overlap ensures that at least one window will *fully contain* every

[4] Llama's prompt was shortened because it failed to follow the longer instructions used for other models and returned unstructured outputs.

such span. Consequently, 97% of arguments are guaranteed to be captured in their entirety by at least one window.

For each window, we constructed an LLM prompt consisting of the relevant spans (formatted as a JSON string) and a system instruction, prompted the model and recorded its output. This approach ensures each span receives rich contextual information.

Algorithm 1 Sliding Window Prompting

```
for each document $d$ in the dataset do
    Retrieve the list of annotated spans $S = [s_1, \dots, s_n]$
    for window start $i = 1$ to $n - w + 1$ step $s$ do
        $W \leftarrow [s_i, \dots, s_{i+w-1}]$
        Assign unique window identifier $ID$
        Format $W$ as a JSON string
        Construct LLM request with system instruction, $W$, and model parameters
        Submit request to LLM
        Record response with $ID$
        Write result (response or error) to the output file (JSONL)
    end for
end for
Summarize results: report window-level statistics (success/failure)
```

The prompts used in this study closely follow human expert annotation protocols. We employ separate prompts for explicit and implicit sense detection, each tailored to our modified PDTB-3 guidelines. For explicit sense annotation, the model is instructed to label only those relations signaled by an explicit connective, always assigning the span containing the connective as Argument 2. For implicit sense annotation, the model is directed to identify only relations where no explicit connective is present. In both cases, the LLM selects minimal, contiguous spans for each argument, assigns an appropriate PDTB-3 sense label, and outputs a confidence score (0–1) based on topic familiarity, logical clarity, and potential sense ambiguity. To facilitate few-shot learning, each prompt includes a 40-span annotation example—covering two consecutive sliding windows—as an illustrative demonstration of the annotation process. Confidence scores are used to resolve disagreements when the same argument pair receives different sense labels across overlapping windows. All prompts provide clear sense definitions, stepwise annotation instructions, and enforce a structured JSON output format, ensuring transparent and reproducible evaluation against human expert annotations.

Setting	**Precision**	**Recall**	**F1**
GPT-o4-mini			
Explicit/Partial	**0.3491**	**0.5022**	**0.4119**
Explicit/Strict	**0.2712**	**0.3914**	**0.3204**
Implicit/Partial	**0.0544**	**0.2180**	**0.0871**
Implicit/Strict	**0.0408**	**0.1639**	**0.0653**
Claude-3.7			
Explicit/Partial	0.2694	0.3964	0.3208
Explicit/Strict	0.1939	0.2862	0.2312
Implicit/Partial	0.0480	0.1956	0.0771
Implicit/Strict	0.0258	0.1056	0.0415
Llama-3.1-8B			
Explicit/Partial	0.0360	0.0391	0.0375
Explicit/Strict	0.0078	0.0084	0.0081
Implicit/Partial	0.0457	0.1244	0.0668
Implicit/Strict	0.0134	0.0366	0.0196

Table 5: Benchmarking LLM Annotation Performance on ENG-DRB.

7.2 Results

Table 5 reports the performance of GPT o4mini and Claude 3.7 on the ENG-DRB dataset, evaluated under both exact match and partial agreement criteria for explicit and implicit discourse relation sense annotation. Overall, both models demonstrate substantially higher performance on explicit relations compared to implicit relations, mirroring trends observed in prior work on PDTB-style corpora. Under the partial agreement setting, GPT o4mini achieves an F1 score of 0.41 for explicit senses, outperforming Claude 3.7 (F1 = 0.32), with a similar advantage observed for exact match (0.22 vs. 0.18 F1). For implicit senses, both models exhibit marked performance degradation, with GPT o4mini attaining F1 scores of 0.09 (partial) and 0.07 (strict), while Claude 3.7 trails slightly (0.08 partial, 0.04 strict). Precision remains low across implicit sense detection, reflecting the intrinsic challenge of reliably inferring discourse relations in engineering instructional text absent explicit connectives.

7.2.1 Explicit Senses Extraction

Our error analysis on the ENG-DRB dataset uncovered three dominant patterns in discourse relation extraction, each pointing toward specific avenues for model refinement.

First, the model shows an overly permissive bias toward temporal connectives: it predicts 208 false positives for Temporal.Asynchronous relations while correctly identifying every genuine instance. In practice, clause-initial "when" or "after" often serve as descriptive or sequential markers rather than explicit Asynchronous links, yet the model treats them uniformly.

Second, for Expansion.Level-of-Detail, precision and recall trade off markedly (83 false positives vs. 56 false negatives). Common coordinate conjunctions such as "and" spuriously trigger expansion labels, whereas explicit cues ("This can be done by...") are occasionally overlooked. This suggests that simple keyword heuristics are insufficient to distinguish narrative continuation from true Expansion.Level-of-Detail sense.

Third, Contingency.Cause exhibits balanced but still substantial confusion (20 false positives vs. 25 false negatives). Weak causal markers such as "so" are over-trusted as 46.71% of "So" in this dataset are not associated with any sense. Prototypical causal sequences ("X then Y") are sometimes missed.

7.2.2 Implicit Senses Extraction

All models struggle profoundly with implicit sense extraction, with F1 scores remaining below 0.09 when considering partial agreement (Table 5). Performance is crippled by extremely low precision (under 0.055), indicating that the models generate a high volume of false positives. This failure suggests that without explicit connectives, current LLMs lack the contextual reasoning necessary to reliably infer nuanced discourse relations in specialized engineering text.

8 Conclusion

This study introduced ENG-DRB, the first PDTB-style discourse relation corpus for engineering tutorial transcripts, addressing a gap in specialized-domain discourse resources. The corpus includes 1,215 annotated relations across civil, mechanical, and electrical/electronics tutorials, highlighting a high proportion of explicit senses, dense causal and temporal relations, and frequent overlapping senses. Benchmarking three state-of-the-art LLMs revealed significant challenges in accurately parsing specialized discourse. Future work should focus on developing models better suited for the complex discourse structures and domain-specific nuances found in engineering contexts.

Limitations

While ENG-DRB fills an important gap in engineering discourse resources, several factors constrain its scope and utility. First, the corpus comprises only 11 tutorial videos (155 minutes; 1,215 relations). Although its scale is comparable to other domain-specific resources such as BioDRB (5,830 senses), it remains modest compared to large, general-domain discourse banks and may limit large-scale model training. Second, annotations were produced by two annotators with a two-round adjudication; our strict F1 (0.72) indicates reasonable but not perfect consistency, reflecting inherent subjectivity in span boundary and sense decisions. Third, current LLMs exhibit substantial performance degradation on long-context inputs (e.g., full-length engineering transcripts), which constrains the reliability of our discourse parsing evaluation over extended instructional texts. Future work should scale up ENG-DRB with more topically diverse tutorial videos, involve a larger annotator pool with enhanced adjudication protocols to boost consistency, and explore models and prompting techniques specifically designed for long-context understanding.

Acknowledgements

We thank the anonymous reviewers and area chairs of IJCNLP–AACL 2025 for their thoughtful feedback and constructive suggestions.

References

Amal Alsaif, Tasniem Alyahya, Madawi Alotaibi, Huda Almuzaini, and Abeer Algahtani. 2018. Annotating attribution relations in Arabic. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Thorsten Brants. 2000. Inter-annotator agreement for a German newspaper corpus. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Chloé Braud, Yang Janet Liu, Eleni Metheniti, Philippe Muller, Laura Rivière, Attapol Rutherford, and Amir Zeldes. 2023. The DISRPT 2023 shared task on elementary discourse unit segmentation, connective detection, and relation classification. In *Proceedings* *of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 1–21, Toronto, Canada. The Association for Computational Linguistics.

Weize Chen, Chenfei Yuan, Jiarui Yuan, Yusheng Su, Chen Qian, Cheng Yang, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2024. Beyond natural language: LLMs leveraging alternative formats for enhanced reasoning and communication. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10626–10641, Miami, Florida, USA. Association for Computational Linguistics.

Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert van Rooij, Kun Zhang, and Zhouchen Lin. 2025. Empowering llms with logical reasoning: A comprehensive survey. *Preprint*, arXiv:2502.15652.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *CoRR*, abs/1804.05685.

Sindhuja Gopalan and Sobha Lalitha Devi. 2016. BioDCA identifier: A system for automatic identification of discourse connective and arguments from biomedical text. In *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)*, pages 89–98, Osaka, Japan. The COLING 2016 Organizing Committee.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Chuyuan Li, Chloé Braud, Maxime Amblard, and Giuseppe Carenini. 2024. Discourse relation prediction and discourse parsing in dialogues with minimal supervision. In *Proceedings of the 5th Workshop on Computational Approaches to Discourse (CODI 2024)*, pages 161–176, St. Julians, Malta. Association for Computational Linguistics.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. Zebralogic: On the scaling limits of llms for logical reasoning. *Preprint*, arXiv:2502.01100.

Wei Liu, Yi Fan, and Michael Strube. 2023. HITS at DISRPT 2023: Discourse segmentation, connective detection, and relation classification. In *Proceedings of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 43–49, Toronto, Canada. The Association for Computational Linguistics.

Yang Janet Liu, Tatsuya Aoyama, Wesley Scivetti, Yilun Zhu, Shabnam Behzad, Lauren Elizabeth Levine, Jessica Lin, Devika Tiwari, and Amir Zeldes. 2024. GDTB: Genre diverse data for English shallow discourse parsing across modalities, text types, and domains. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12287–12303, Miami, Florida, USA. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. GPT-4 Technical Report. *Preprint*, arXiv:2303.08774.

Rashmi Prasad, Susan McRoy, Nadya Frid, Aravind Joshi, and Hong Yu. 2011. The biomedical discourse relation bank. 12:188. Article number: 188.

Rashmi Prasad, Bonnie Webber, and Alan Lee. 2018. Discourse annotation in the PDTB: The next generation. In *Proceedings of the 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 87–97, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Rashmi Prasad, Bonnie Webber, Alan Lee, and Aravind Joshi. 2019. Penn discourse treebank version 3.0. *LDC2019T05*.

Tatjana Scheffler, Berfin Aktaş, Debopam Das, and Manfred Stede. 2019. Annotating shallow discourse relations in Twitter conversations. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 50–55, Minneapolis, MN. Association for Computational Linguistics.

Evgeny Stepanov and Giuseppe Riccardi. 2014. Towards cross-domain PDTB-style discourse parsing. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, pages 30–37, Gothenburg, Sweden. Association for Computational Linguistics.

Kangda Wei, Aayush Gautam, and Ruihong Huang. 2024. Are LLMs good annotators for discourse-level event relation extraction? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1–19, Miami, Florida, USA. Association for Computational Linguistics.

Deniz Zeyrek and Murathan Kurfalı. 2018. An assessment of explicit inter- and intra-sentential discourse connectives in Turkish discourse bank. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Deniz Zeyrek, Amália Mendes, Yulia Grishina, Murathan Kurfalı, Samuel Gibbon, and Maciej Ogrodniczuk. 2020. TED Multilingual Discourse Bank

(TED-MDB): A parallel corpus annotated in the PDTB style. 54(2):587–613.

Yuping Zhou and Nianwen Xue. 2012. PDTB-style discourse annotation of Chinese text. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–77, Jeju Island, Korea. Association for Computational Linguistics.

A Video metadata

We show the metadata of the videos we used in Table 6.

B Hyperparameter Grid Search for HITS Benchmarking experiments.

We conducted a grid search over key training hyperparameters for to identify effective configurations for fine-tuning. Specifically, we varied the learning rate in $\{2 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$, the number of training epochs in $\{5, 10\}$, and the training batch size in $\{8, 16, 32\}$, resulting in 18 total configurations. All other parameters were kept fixed: we used the roberta-base backbone with a maximum sequence length of 512, evaluation batch size of 32, dropout rate of 0.1, weight decay of 0.1, warmup ratio of 0.06, and maximum gradient norm of 2.0. Each run used the same random seed (106524) and was trained on a single NVIDIA H100 (80 GB) GPU. We use the same grid search settings for Task1&2 and Task 3. The best-performing model was selected based on the highest validation score observed during training as reported in Section 6.

C Hyperparameter for LLM Benchmarking Experiments

For the experiment using the LLaMA-3.1-8B-Instruct model, we configured the model to generate up to 1,024 new tokens per window, with a maximum input length of 2,048 tokens including the context. To ensure deterministic and fully reproducible generation, we set the temperature to 0.0 and top-p to 0.0, with sampling disabled. This configuration guarantees that the same input consistently yields the same output. All experiments were conducted on a single NVIDIA H100 GPU.

For the experiment using the OpenAI o4-mini-2025-04-16, we configured the model to generate up to 100,000 new tokens per window, with a maximum input length of 200,000 tokens including the context. To ensure deterministic and fully reproducible generation, we set the temperature to 0.0 and top-p to 0.0, with sampling disabled. The experiment consumes 2,178,419 tokens with batch API and $4.65 (with preliminary trial and error process).

For the experiment using the Claude-3-7-sonnet-20250219, we configured the model to generate up to 3,000 new tokens per window. To ensure deterministic and fully reproducible generation, we set the temperature to 0.0, with sampling disabled and all other parameters using their default values (top k of 40 maintained). The experiment consumes 2,692,805 tokens as input and 432,053 tokens as output, and $14.56 (with preliminary trial and error process).

D A Prompt for separating scripts into EDUs

# Splitting Sentences into Argument Candidate Spans in a Tutorial Discourse

# 1. Purpose

The purpose of this stage is to separate the sentences into potential clause-level spans so that the potential senses (i.e., logical relationships) within sentences can be later annotated.

# 2. Separating Spans Within a Sentence When It Contains Logical Relationships (i.e., Senses)

When a sentence contains one of the following logical relationships (senses), you should separate the sentence into multiple spans corresponding to each argument of the sense. The senses are adopted from the PDTB-3 framework and include temporal relation, contingency relation, comparison relation, and expansion relation. This section summarizes the PDTB-3 sense labels, explaining each type of discourse relation with simplified definitions and typical examples. Discourse relations describe how two parts of text (arguments) are connected logically.

## 1. Temporal Relations

Definition: Temporal relations connect events in time, indicating when they occur relative to each other.

### 1.1 Temporal.Synchronous

- **Explanation**: Both events happen at the same time or overlap. - **Examples**: - "*While* she reads, he listens to music." - "The crowd cheered *as* the team scored."

......

Video Title	Domain	Duration	Likes	Dislikes	Tokens	Spans	Senses	Video Link
Water Main Break Repair	C	0:14:52	771	13	2570	264	131	https://youtu.be/wGJNFfqP2y8?si=5qhxT46MCacmnx2J
Pouring Concrete Footings	C	0:18:24	11000	192	2459	223	105	https://www.youtube.com/watch?v=qo7eL5yp56A
Garage Floor Crack	C	0:14:38	22000	765	2453	227	133	https://youtu.be/bXDYgxM-PTc?si=VpPfeZDm_VvyZ2kv
Dig a Basement	C	0:09:16	3100	44	1539	150	81	https://www.youtube.com/watch?v=rAmAoxmWkLI
Occupancy Sensor (truncated)	E	0:12:29	2100	88	1546	157	91	https://youtu.be/9lZUP-Fe9to?si=hmOCXyLcKisCra__
PLC Output	E	0:07:24	3100	19	1228	116	57	https://www.youtube.com/watch?v=U3fj4tHHS8M
Voltage Drop	E	0:33:14	15000	573	5444	490	279	https://youtu.be/DfLyh43iihM?si=m30Jz0cPJSN_3BP2
Simple Boiler Maintenance	M	0:01:58	676	8	471	52	36	https://youtu.be/xtuzsK6RFO0?si=4pw0CPO_NJRir1iP
RV Plumbing	M	0:07:23	1600	55	1037	112	67	https://www.youtube.com/watch?v=MjyU2eClPcA
Furnace Troubleshooting	M	0:26:32	17000	503	3392	331	177	https://youtu.be/dJzNrw6L_YU?si=uqZ4IqHurvClHM7G
Welding Techniques Steel Columns	M	0:09:22	–	–	1462	137	58	https://www.youtube.com/watch?v=-8D_sPGBstI
Total	–	2:35:32	–	–	23601	2259	1215	–

Table 6: Summary of Technical Videos with Metadata

(brief introduction of all senses used in this study)

# 3. Step-by-Step Instructions

Step 1: Start from the first sentence.

Step 2: Read the current sentence carefully.

Step 3: Split the spans in the current sentence.

- **3.1 Identify Logical Relationships (Senses):**

- **Looking for senses (logical relationships) in the current sentence according to the senses listed in Section 2**

- **If a sentence contains sections (e.g., clauses, phrases, or "to + verb" forms) that are connected by logical relationships (senses) listed above, separate the sentence into multiple spans corresponding to each argument of the sense.**

- **Look for conjunctions and connective phrases** such as "and," "but," "if," "because," "so," "when," "after," "before," "while," "although," "for example," " instead of", "rather", "otherwise", etc., which often signal the presence of a sense.

- **Identify the arguments** of the sense. Each argument should be a separate span.

- **Examples:**

- (some brief examples for each category of senses)

- **3.2 If No Logical Relationships Are Present:**

- - If the sentence does not contain any of the specified logical relationships and the entire sentence serves a single function, annotate the sentence as a single span with its function. - **3.3 Handling Irrelevant Content:**

- If a span has no function relevant to the tutorial (e.g., the author made a joke or included other irrelevant content), label it as N/A.

Step 4: Review the previous several spans.

- **Double-check the annotated functions.** If you have new understanding about the function of the spans or the logical relationships, update them accordingly.

Step 5: Move to the next sentence and repeat Steps 2 to 4 until the entire discourse is separated into spans and functions are annotated.

—

# Additional Notes

- **Be Vigilant for Multiple Senses:**

- A sentence may contain multiple senses. In such cases, ensure that each sense is considered, and the sentence is split accordingly. - **Hierarchy of Senses:**

- If multiple senses are present and overlap, prioritize splitting based on the most prominent or governing sense in the context. - **Use of Connectives:**

- Words like "if," "because," "so," "when," "after," "before," "although," "for example," etc., are strong indicators of logical relationships and should prompt you to consider splitting the sentence. Punctuations (e.g., commas and semicolons) are also potential indicators of logical relationships - **Context Matters:**

- Always consider the context to accurately determine the sense and function of each span. - **Consistency:**

- Apply these guidelines consistently throughout the annotation to maintain uniformity.

### When Not to Split Spans

- **Coordinating Conjunctions:**

- The mere presence of coordinating conjunctions (and, but, or) does NOT automatically warrant splitting spans. On the other hand, regardless of the conjunction used (and, but, or), if a span contains coordinated clauses that each serve as an argument in a logical relationship, split them into separate spans. - **Examples of when NOT to split:**

1. (some examples) ......

- **Rule of Thumb:** When any span contains more than one clause that plays a role in a logical relationship, split the span so that each clause is isolated as an individual argument of that relationship. If you are not sure, just split them into separate spans.

# Examples

### Example 1

#### Input - Original transcript:

All right, here we go. Today we're going to do a simple steam boiler maintenance. This is a real basic one; we're not going to get into any ignition or anything like that.

......

#### Output - Annotated spans and functions:

1 All right, here we go. 2 Today we're going to do a simple steam boiler maintenance. 3 This is a real basic one; we're not going to get into any ignition or anything like that.

......

E A Prompt for Benchmarking LLMs

Note: This is the version for benchmarking explicit senses. Implicit version can be modified accordingly

You are an experienced, professional, rigorous computational linguist. You are now work as the annotator of the tutorial discourse PDTB sense dataset. Your task is extract the senses in the given spans from hands-on engineering tutorial discourses. Specifically, you only identify the senses where the explicit connective is shown in the argument. Also, please make sure the argument with connective is assigned as Argument 2, regardless of its position. When annotating arguments in PDTB-3 for a specific sense, the annotated spans for each argument should be the minimal span(s) sufficient to unambiguously convey that sense. This minimal span must preserve the original meaning within its discourse context. Please annotate the senses based on the given PDTB sense definition. In addition, please also indicate a confidence level (0-1) for each sense you detected in the final output based on 1) your familiarity with the topic, 2) the logic clarity of the candidate arguments, 3) possible similar senses that may cause confusion. The definition of senses, example input, annotation steps, output format, example output, and the discourse section to be annotated is given below:

# Definition of senses: " ## 1.1 PDTB3 Sense Classification Here we provide definitions of the PDTB-3 sense labels.

### 1.1.1 Temporal Relations......

### 1.1.2 Contingency Relations......

### 1.1.3 Comparison Relations......

### 1.1.4 Expansion Relations......

"

# Example inputs: " (an exmaple showing the input format) "

# Annotation Steps

1. **Initial Reading**: Review the provided discourse to understand the content and context.

2. **Identify Connectives**: Locate explicit connectives and determine their corresponding Arg1 and Arg2 spans. Always assign the span containing the explicit connective as Argument 2.

3. **Determine Sense**: Use the definitions from the PDTB-3 sense hierarchy to classify the senses between arguments (e.g., Temporal, Contingency, Comparison, Expansion).

4. **Minimal Span Identification**: Ensure the chosen spans for each argument are the minimal portions needed to convey the intended sense correctly.

5. **Confidence Assessment**: Evaluate your confidence level (0-1) for each identified sense, considering familiarity with the topic, clarity of logic, and potential for similar sense confusion.

6. **Annotate**: Structure annotations clearly, specifying argument spans, sense, explicit connective, and confidence score.

# Output Format

Each sense annotation should be in the following JSON format: “‘json (an exmaple showing the input format) "

# The discourse section to be annotated is as follows: