Leveraging Codebook Knowledge with NLI and ChatGPT for Zero-Shot Political Relation Classification

Yibo Hu*, **Erick Skorupa Parolin**†, **Latifur Khan**†, **Patrick T. Brandt**†,
Javier Osorio‡, **Vito J. D'Orazio**§
*Georgia Institute of Technology, †The University of Texas at Dallas,
‡The University of Arizona, §West Virginia University
yibo.hu@gatech.edu, erickparolin@gmail.com, {lkhan,pbrandt}@utdallas.edu,
josorio1@email.arizona.edu, vito.dorazio@mail.wvu.edu

Abstract

Is it possible accurately classify political relations within evolving event ontologies without extensive annotations? This study investigates zero-shot learning methods that use expert knowledge from existing annotation codebook, and evaluates the performance of advanced ChatGPT (GPT-3.5/4) and a natural language inference (NLI)-based model called ZSP. ChatGPT uses codebook's labeled summaries as prompts, whereas ZSP breaks down the classification task into context, event mode, and class disambiguation to refine task-specific hypotheses. This decomposition enhances interpretability, efficiency, and adaptability to schema changes. The experiments reveal ChatGPT's strengths and limitations, and crucially show ZSP's outperformance of dictionary-based methods and its competitive edge over some supervised models. These findings affirm the value of ZSP for validating event records and advancing ontology development. Our study underscores the efficacy of leveraging transfer learning and existing domain expertise to enhance research efficiency and scalability. The code is publicly available[1].

1 Introduction

Event coding is a crucial task in political violence research for both academic and policy communities. It transforms unstructured text from news articles into structured event data, represented as source-action-target triplets, achieved through entity extraction and relation classification. It provides a structured record of interactions among political actors and serves as input for monitoring, understanding, and forecasting political conflicts and mediation processes worldwide (Schrodt and Gerner, 1996; Schrodt et al., 2003, 2004; Schrodt, 1997, 2006a, 2011; Shellman and Stewart, 2007; Shearer, 2007; Brandt et al., 2011, 2013, 2014). However, manually coding events from extensive datasets is labor-intensive.

To streamline this, experts have developed event ontologies and knowledge bases (McClelland, 1978; Azar, 1980; Gerner et al., 2002; Bond et al., 2003; Schrodt, 2006b; Boschee et al., 2016; Lu and Roy, 2017; Osorio and Beltrán, 2020; Osorio et al., 2019). Yet, traditional pattern-matching models based on static dictionaries suffer from inflexibility, low recall, and high maintenance costs. Recent advancements in deep learning and pre-trained language models (PLMs) offer promising supervised learning solutions (Glavaš et al., 2017; Büyüköz et al., 2020; Olsson et al., 2020; Örs et al., 2020; Parolin et al., 2020, 2021, 2022b; Hu et al., 2022a). Yet, their reliance on extensively annotated datasets introduces significant challenges, especially for in-depth and subnational studies requiring nuanced categorization and non-exclusive labeling within political event ontologies. Moreover, labeled datasets lack flexibility and may require frequent relabeling as ontologies evolve. Thus, much of the current PLM-based research in event coding targets broad, coarse-grained categorizations, often constrained by limited evaluation sets. Costs, time, and effort associated with developing training data have foiled the large-scale adoption and ready deployment of PLMs by government security agencies, researchers, and practitioners in need of monitoring and understanding rapidly-changing conflict processes around the world.

In light of these challenges, we pose the following questions: (1) Is it possible to leverage existing expert knowledge to enhance the efficiency of event coding without extensive annotation of new data? (2) Is it possible to create an interpretable and adaptable system that easily accommodates ontology or schema changes?

To tackle these questions, our paper focuses on relation classification, a key aspect of event coding. The goal is to categorize events in a source-

[1] https://github.com/snowood1/Zero-Shot-PLOVER

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 583–603

CAMEO Root.	PLOVER Root.	Quad.
01- Make Public Statement	dropped	
02- Appeal	dropped	
03- Express Intent to Cooperate	AGREE	1. V-Coop.
04- Consult	CONSULT	1. V-Coop.
05- Engage in Diplomatic Cooperation	SUPPORT	1. V-Coop.
06- Engage in Material Cooperation	COOPERATE	2. M-Coop.
07- Provide Aid	AID	2. M-Coop.
08- Yield	YIELD	2. M-Coop.
09- Investigate	ACCUSE	3. V-Conf.
10- Demand	REQUEST	3. V-Conf.
11- Disapprove	ACCUSE	3. V-Conf.
12- Reject	REJECT	3. V-Conf.
13- Threaten	THREATEN	3. V-Conf.
14- Protest	PROTEST	4. M-Conf.
15- Exhibit Force Posture	MOBILIZE	4. M-Conf.
16- Reduce Relations	SANCTION	4. M-Conf.
17- Coerce	COERCE	4. M-Conf.
18- Assault	ASSAULT	4. M-Conf.
20- Unconventional Mass Violence	ASSAULT	4. M-Conf.

Table 1: CAMEO/PLOVER's Rootcodes and Quadcodes (1-Verbal Cooperation, 2-Material Cooperation, 3-Verbal Conflict, and 4-Material Conflict).

target pair following a predefined event ontology PLOVER (Open Event Data Alliance, 2018) without external labeled data. We achieved this by combining the transferred semantic knowledge of PLMs with expertise derived from annotation codebooks. The codebook, as depicted in Figure 2, contains label descriptions and guidelines for resolving confusing labels. To unlock this knowledge, we explore two zero-shot methods: the emerging ChatGPT (GPT-3.5/4), and our proposed NLI-based model called ZSP (**Z**ero-**S**hot fine-grained relation classification model for **P**LOVER ontology).

While GPT-4 showcases notable improvements over GPT-3.5, it still exhibits instability in fine-grained tasks, promising further enhancement. Conversely, ZSP, despite being built upon a smaller model, offers substantial advantages. It leverages easily constructed hypotheses from the codebook and employs a tree-query framework to capture nuanced semantics and mode distinctions within a focused set of hypotheses at each level. Additionally, ZSP's adaptability allows straightforward updates by modifying the hypothesis table or class disambiguation rules to align with evolving ontologies. This approach proves more cost-effective than maintaining extensive dictionaries or re-labeling datasets for event record validation.

In sum, the untapped potential of GPT-4 and the success of ZSP encourage experts to reevaluate the value of existing knowledge bases and inspire innovative uses of this knowledge to expedite research within the political science community.

```
Obama said he won't provide military aid to Israel.
Source:    Obama-USAGOV                Target:   Israel-ISR
Action:    1222-Reject request for military aid
Root.:     12- REJECT                  Quad.:    3. V-Conf.

Other event modes ------------------- Root. ----- Quad. -
Obama halted   military aid to Israel. SANCTION  4. M-Conf.
Obama provided military aid to Israel. AID       2. M-Coop.
Obama agreed to provide aid to Israel. AGREE     1. V-Coop.
```

Figure 1: Event coding illustration: How event modes affect Rootcode and Quadcode labeling for sentences involving identical entities.

2 Preliminaries

2.1 Event Coding, Ontology, and Mode

The ontology of the event coding system defines how to code the actors (McClelland, 1978; Azar, 1980; Jones et al., 1996; Bond et al., 2003; Raleigh et al., 2010; Mitamura and Hovy, 2015; Boschee et al., 2016). One prominent schema is **CAMEO** (Gerner et al., 2002), which incorporates knowledge from the codebook[2], action-pattern dictionaries, and actor dictionaries. It categorizes political interactions into 200+ fine-grained 4-digit codes (`01XX–20XX`). These are then aggregated into 20 more frequently utilized **Rootcodes** (`01–20`), and further into 4 high-level **Quadcodes**: 1-Verbal Cooperation, 2-Material Cooperation, 3-Verbal Conflict, and 4-Material Conflict. Later, the **PLOVER** scheme (Open Event Data Alliance, 2018) simplifies CAMEO by removing 4-digit codes, reducing Rootcodes to 16, and enhancing semantic clarity. Rootcode and Quadcode overviews of CAMEO/PLOVER are presented in Table 1, with a codebook snippet in Figure 2. Appendix Table 17 and the codebook present the Rootcode in detail.

Figure 1 illustrates an event coding scenario where the interaction between the source (Obama representing the USA government) and the target (Israel, coded as ISR) is classified using lower-level 4-digit codes as well as higher-level Rootcodes and Quadcodes. However, the classification is highly sensitive to subtle variations in what we term **event mode**—whether Obama has provided, plans to provide, has stopped, or intends to stop military aid—resulting in significant adjustments to Rootcodes and Quadcodes for identical entities.

PLOVER's codebook[3] suggested the concept of event mode through auxiliary modes—historical, future, hypothetical, or negated—to flexibly depict

[2] https://eventdata.parusanalytics.com/cameo.dir/CAMEO.Manual.1.1b3.pdf

[3] https://github.com/openeventdata/PLOVER/blob/master/PLOVER_MANUAL.pdf

event statuses and assist in labeling. However, it did so without providing strict definitions or implementations. Inspired by this, we refine these event modes and integrate them into our NLI system to enhance classification accuracy. This refinement process and its impact on enhancing event coding precision are detailed in Section 3.2 and further elaborated in Appendix A.

The shift from the dictionary-based CAMEO to the more semantically friendly PLOVER aligns with the domain's broader trend. Our focus on PLOVER is the result of careful consideration and validation with domain experts.

2.2 Related Work

Relation or event extraction has been studied across various domains (Hendrickx et al., 2019; Zhang et al., 2017; Han et al., 2018; Du and Cardie, 2020; Luan et al., 2018; Riedel et al., 2010; Fincke et al., 2022), with some studies partially overlapping in entities or categorizations relevant to political science (Doddington et al., 2004; Ebner et al., 2020; Li et al., 2021). However, our work distinguishes itself by considering event modes, a dimension not fully explored in existing works.

Our work also relates to zero-shot learning across various schemes (Huang et al., 2018; Obamuyide and Vlachos, 2018; Yin et al., 2019; Meng et al., 2020; Geng et al., 2021; Lyu et al., 2021; Sainz et al., 2021), especially socio-political event classification (Hürriyetoğlu et al., 2021; Radford, 2021; Barker et al., 2021; Haneczok et al., 2021; Halterman and Radford, 2021). However, many works focus on sentence-level classification rather than relations between multiple entity pairs. The others with complex templates **cannot be** adapted to our political ontology easily. Thus, we design our framework to efficiently integrate with the existing knowledge base.

Finally, recent large language models (LLMs) (Brown et al., 2020; Ouyang et al., 2022; OpenAI, 2022, 2023) have greatly advanced zero-shot learning in reasoning and text generation (Hu et al., 2022b; Halim et al., 2023; Jin et al., 2024). However, the application of ChatGPT for zero-shot event extraction remains underexplored and lags behind advanced supervised methods (Yuan et al., 2023; Cai and O'Connor, 2023; Li et al., 2023; Gao et al., 2023; Aiyappa et al., 2023). We will evaluate ChatGPT on PLOVER as part of our investigation.

3 Approach

We start by discussing the discovery of NLI as a potential solution and the construction process of the NLI-based ZSP framework, followed by the deployment of ChatGPT.

3.1 Limitations of NLI for Event Coding

NLI measures how likely a premise entails a hypothesis (Bowman et al., 2015; Williams et al., 2017). Initially, we explores the feasibility of using NLI to assign PLOVER codes by selecting the most probable entailed hypothesis from a set of candidates. We designed a tiny experiment with only 18 hypotheses derived from the Rootcode names[4].

Table 2 illustrates three example hypotheses, where <S> and <T> denote the source (Indonesian students) and the target (President Suharto's government), respectively. The labeled premise "THREATEN 3" indicates the intention to initiate protests. Notably, NLI accurately recognizes AID as contradictory and identifies REQUEST and PROTEST as entailments. Moreover, the tiny NLI model with only 18 hypotheses surpasses dictionary-based methods that rely on $81k$ verb patterns, with a remarkable macro F1 increase of 17.1% for Quadcode classification, thus comfirming the NLI potential as a valuable solution.

However, upon closer examination, we find that NLI models measuring semantic entailment may not directly suit our classification task, as the best entailed hypotheses do not always match our desired labels. The adaptation raises two key issues:

First, NLI disregards event mode. In Table 2, the premise labeled as THREATEN stands for a hypothetical, verbal protest event. NLI partially captures the event's context (PROTEST) but fails to consider its mode. To address this, we incorporate mode information to enhance candidate precision.

Second, event category labels lack mutual exclusivity in semantics. In Table 2, the premise correctly entails both PROTEST and REQUEST with high scores from the semantic aspect. However, in CAMEO/PLOVER's single-label schema, the context "demonstrate to demand reforms" aligns with PROTEST, a Material Conflict, rather than REQUEST, a Verbal Conflict. An easy solution is to prioritize "protest" over "request" when encountering "protest to request", following the codebook's disambiguation rules illustrated in Figure 2.

[4]See more details about this "Tiny model" experiment in Section 4.4, and Rootcode names in Table 1).

Premise: Thousands of Indonesian students said they would stage mass demonstrations Saturday, demanding political reforms from President Suharto's government.
Source <S>: Indonesian students
Target <T>: President Suharto's government
Gold Label: THREATEN 3; threaten political dissent.

(a) Basic Hypotheses	Label	Score
<S> requested <T>.	REQUEST 3	92.7
<S> protested against <T>.	PROTEST 4	92.5
<S> provided aid to <T>.	AID 2	0.8
(b) Adding Mode		
<S> threatened to protest against <T>.	✓**THREAT. 3**	**97.3**
(c) Adding Class Disambiguation		
`Override REQUEST if PROTEST exists` → ~~REQUEST~~		

Table 2: Entailment scores (%) for hypotheses on a sentence labeled as "THREATEN 3" (Rootcode text + Quadcode digit). Adding mode or class disambiguation to basic hypotheses improves prediction precision.

Premise: Thousands of Indonesian students agreed to suspend Saturday's demonstrations, demanding political reforms from President Suharto's government.
Source <S>: Indonesian students
Target <T>: President Suharto's government
Gold Label: AGREE 1; express intent to ease popular dissent.

Mode Hypotheses for "Protest"	Mode	Label	Score
<S> protested against <T>.	-	PROTEST 4	92.5
<S> increased protests against <T>.	P	PROTEST 4	0.1
<S> launched more protests against <T>.	P	PROTEST 4	0.0
<S> reduced protests against <T>.	CP	YIELD 2	95.2
<S> threatened to protest against <T>.	F	THREAT. 3	67.5
<S> promised to reduce protests against <T>.	**CF**	**AGREE 1**	**97.1**
<S> will reduce protests against <T>.	CF	AGREE 1	96.3

Table 3: Entailment scores (%) for hypotheses on a sentence labeled as "AGREE 1" (Rootcode text + Quadcode digit). Adding **Mode** (P, F, CP, CF) improves prediction precision compared to mode exclusion (-).

In summary, we identify three key dimensions to ensure accurate predictions: Context, Mode, and Class Disambiguation. Firstly, we narrow down predictions to the top candidates, PROTEST and REQUEST. Secondly, we incorporate mode information and identify the event as future, verbal, or hypothetical. Lastly, we apply the class disambiguation rule, giving precedence to PROTEST over REQUEST. By combining these dimensions, we achieve the final correct answer THREATEN. These findings motivate our NLI-based framework in Figure 2. Next, we provide detailed explanations for each component.

3.2 Enabling NLI to Classify Event Mode

NLI's inability to accurately determine the event mode often leads to misclassification. In Table 3, we present a sentence with reversed labels compared to another in Table 2. Although labeled as AGREE for Verbal Cooperation, indicating a willingness to mitigate dissent, it incorrectly scores high (76.4%) for the hypothesis "protested against". Semantically, this isn't entirely wrong, as "agreed to ease protests" implies prior protests, but it suggests cooperation rather than the implied conflict of the hypothesis.

To address this, we introduce **mode-aware hypotheses** that incorporate four event modess, adapted from PLOVER's guidelines: Past (**P**) for historical events or events that initiated or are ongoing, Future (**F**) for future, verbal, or hypothetical events, Contradict_Past (**CP**) and Contradict_Future (**CF**) for their respective contradictions. See examples in Table 3.

Labels for each mode are directly adopted from the codebook, requiring no new definitions. For example, PROTEST's CF is labeled as AGREE, mirroring "03- EXPRESS INTENT TO COOPERATE" Rootcode, specifically item "0352- Express intent to ease popular dissent." Likewise, the CP events like "reduced protests against" are labeled as YIELD, following CAMEO code 0833, which signifies yielding to demands.

To facilitate this nuanced classification approach, we have assembled a mode mapping table (Table 7) that clarifies mode transitions and associated label modifications. This streamlined process not only simplifies the task of classification but also significantly reduces the complexity of navigating the codebook. For an in-depth exploration of these event modes, please see Appendix A.

The mode-aware hypotheses in Table 3 enable NLI to accurately dismiss the Past hypotheses and correctly score the CF mode highest, underscoring its capability to discern semantic subtleties. Additionally, NLI's semantic generalization avoids the need for exact word matching, unlike the complex dictionary-based methods. For example, similar hypotheses such as "increase" and "launch more protests" receive similar scores, and phrases like "promised to reduce" are considered similar to "will reduce".

Our **mode-aware NLI** system leverages these insights, using the codebook to guide hypothesis construction. This approach enables easy conversion of present-tense label description into different modes or tenses, even for non-experts. The codebook also provides contrasting label examples, like "YIELD: ease protests" and "AGREE: agree to ease

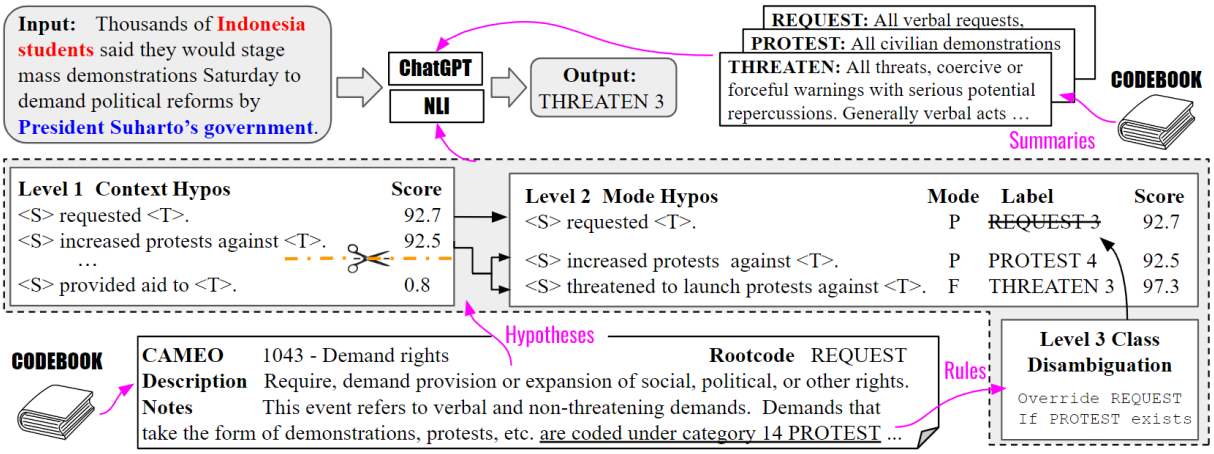


Figure 2: Two zero-shot approaches for classifying relation labels (Rootcode and Quadcode) in a source - target pair. ChatGPT employs prompts designed from the codebook's label summaries, while ZSP utilizes a pretrained NLI model and a tree-query system. Hypotheses and class disambiguation rules are derived from the codebook and enhanced with mode considerations (e.g., **P**ast, **F**uture). The tree-query framework reduces query time and improves precision by filtering candidates, determining modes, and eliminating ambiguity.

protests" for the CP and CF modes for PROTEST, significantly streamlining the engineering process. We only need to ensure hypotheses, especially in Past mode, clearly reflect event trends. For instance, in Table 3, "protested against" was rephrased to "increased/launched more protests against" for enhanced clarity. Likewise, "imposed bans" can be modified to "increased/imposed more bans".

3.3 Class Disambiguation

To address the issue of class ambiguity and overlaps in CAMEO/PLOVER, experts have documented instructions and annotation rules in the codebook. Annotators frequently consult the codebook when faced with ambiguous cases. In contrast, we can integrate this information into our machine to reduce manual annotation and effectively handling boundary cases. Note that incorporating excessive rules goes against our goal of designing a simple and adaptable system. It can lead to overfitting and inflexibility, similar to the limitations found in traditional dictionary-based methods. Therefore, we've chosen to include only the most frequent rules explicitly outlined in the codebook, considering this step as supplementary to our system.

One notable rule, referred to as the **Conflict Override**, is summarized from the codebook. This rule gives priority to labels in Material Conflict over Verbal Conflict, as depicted in Figure 2. If the top predictions include candidate labels in Material Conflict, the labels in Verbal Conflict will be overridden. For example, we label "protest to request" as material PROTEST other than verbal REQUEST, as explained in Section 3.1. Similarly, we label "convict and arrest" as material COERCE other than verbal ACCUSE, considering the more severe actions involved. These rules can be easily customized and expanded by users to accommodate changes in the schema or ontology. Additional discussions are provided in Appendix B and C.

3.4 Tree-Query NLI Framework

We enhance precision and efficiency by integrating mode-aware NLI with class disambiguation into a tree-query framework, as shown in Figure 2. This contrasts with the "flat-query" method, where hypotheses are arranged at a single level, by organizing them hierarchically.

At **Level 1 Context**, we compare 76 Past hypotheses ($\approx$ 5 hypotheses per Rootcode) to classify the premise's context. Using a customized threshold, such as selecting the top-3 candidates with scores higher than the maximum score minus 0.1, we narrow down the most probable candidates. In the example, this filtering yields two candidates related to REQUEST and PROTEST.

At **Level 2 Mode**, we compare the hypotheses in other modes for the selected candidates to determine their mode. We focus on two types of modes in the experiments: Past and Future. For instance, PROTEST leads to two branches - the existing PROTEST and a new THREATEN (PROTEST+future). However, for certain Rootcodes like REQUEST, querying their Future variants is unnecessary since the labels remain the same from Past to Future (details in Table 7 and Ap-

pendix B). This reduces the number of Future hypotheses in Level 2 to 58, and only a subset requires querying per premise. In Figure 2, we collect all necessary scores for Level 2 analysis by querying just the new THREATEN hypothesis once.

At **Level 3 Class Disambiguation**, we apply specific rules, including the Conflict Override, to eliminate REQUEST since PROTEST already exists among the top predictions from Level 2.

ZSP is interpretable, flexible, efficient, and precise. First, we split the complicated, ambiguous classification into a simple tree framework that both computer science and political science researchers can easily understand. Second, experts can quickly update ZSP by revising the hypothesis table or class disambiguation rules according to a evolving ontology, which is much cheaper than maintaining large dictionaries or relabeling a dataset (detailed in Appendix C). Third, it improves efficiency. For instance, we only query 76 times in Level 1 + one time in Level 2 without comparing all 134 hypotheses in Figure 2. Finally, NLI scores within ZSP accurately capture nuanced entailment relations within the limited scope of compared hypotheses at each level. This minimizes potential errors that can arise from mixed hypotheses in different contexts and modes. We will validate this in experiments.

3.5 ChatGPT

Besides our proposed NLI-based ZSP model, we explored the zero-shot performance of LLMs on this task. We focused on two versions of ChatGPT: GPT-3.5 and GPT-4. We used the OpenAI API and designed prompts that incorporate task descriptions and pre-defined label sets, building upon insights from previous research (Wei et al., 2023; Li et al., 2023). The label descriptions were summarized and refined from the PLOVER codebook's comprehensive Rootcode descriptions. Further insights are available through exemplified input output instances in Appendix H.

4 Experiments

4.1 Datasets

Since there were limited datasets with fine-grained annotation, we built a Rootcode-level **PLV** dataset from the CAMEO codebook and a balanced coarse-grained-labeled dataset (Parolin et al., 2022a), resulting in 1050 training examples and 1033 testing examples. We built three classification tasks with varying degrees of complexity: Binary (cooperation vs. conflict), Quadcode, and Rootcode.

Besides the political science dataset PLV, we also explored how event ontology knowledge benefits and generalizes in other NLP datasets. Thus, we built a binary **A/W** dataset from **A**CE (Doddington et al., 2004) and **W**ikiEvents (Li et al., 2021), which contain many conflict-related subjects that overlap the political ontologies. A/W consists of 802 training examples and 805 testing examples. See more details in Appendix D.

4.2 Setup

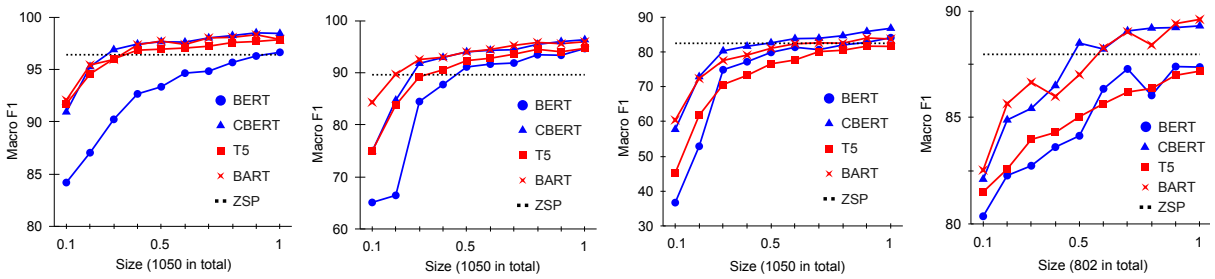
Regarding our proposed **ZSP**, we incorporated a finetuned NLI model[5] into our tree-query system. For **ChatGPT**, we used OpenAI's Chat completions API to access **GPT-3.5** and **GPT-4**. To assess the practical usefulness of these zero-shot models, we compared them with notable baselines, including Universal PETRARCH (**UP**) (Lu and Roy, 2017), a widely-used dictionary-based CAMEO event coder. We measured UP's ideal performance on relation classification by considering incomplete triplets, as detailed in Appendix E.

Additionally, we examine the performance of various supervised learning models, including masking language models (MLM) like **BERT**-base-uncased (Devlin et al., 2018) and ConfliBERT-scr-uncased (**CBERT**) (Hu et al., 2022a). Notably, CBERT reports greater effectiveness in the political science domain. We also use text generation models, namely **BART** (Lewis et al., 2020) and **T5** (Raffel et al., 2020), to generate original label texts for this classification task. We trained these supervised models on either the entire training set or sampled subsets of varying sizes using a single V-100 GPU with default hyperparameters. Subsequently, we evaluated them on the complete testing dataset. We ran each scenario with five different seeds and reported average results for reliability.

4.3 Results and Analysis

We summarized the performance of dictionary-based and zero-shot models, as well as the supervised learning models trained on the entire training datasets, in Table 4. Additionally, in Figure 3, we compared ZSP with supervised learning models trained on varying limited datasets. UP and ChatGPT were excluded from the analysis due to their significant performance gap compared to the other

[5] https://huggingface.co/roberta-large-mnli



(a) PLV Binary (b) PLV Quadcode (c) PLV Rootcode (d) A/W Binary

Figure 3: Performance vs. varying sized training datasets.

Type	Model	PLV Bin.	PLV Quad	PLV Root	A/W Bin.	Avg.
Dict. & Zero-shot	UP	80.8	51.8	46.3	67.2	61.5
	GPT-3.5	90.1	66.2	40.9	76.3	68.4
	GPT-4	93.4	76.7	61.5	87.0	79.7
	ZSP	**96.4**	**89.6**	**82.4**	**88.0**	**89.1**
Super-vised	BERT	96.6	94.6	84.0	87.4	90.7
	CBERT	**98.4**	**96.3**	**86.7**	89.3	**92.7**
	T5	97.8	94.7	81.6	87.2	90.3
	BART	97.9	95.9	83.7	**89.6**	91.8

Table 4: Macro F1 scores of models on diverse dataset-task combinations and average results.

models, to maintain focus and relevance.

Supervised learning. Among supervised models, CBERT emerged as the top performer, surpassing BERT with less data required. BART closely trailed. It outperformed T5 by exhibiting less overfitting on small, imbalanced labeled datasets.

ZSP. ZSP consistently outperformed UP and ChatGPT, and it achieved competitive results with supervised learning models in most tasks (Figures 3a, 3c, 3d). Notably, in these scenarios, ZSP matched BERT and T5, while the stronger models CBERT and BART still required 25%-50% of the training data to achieve a slight performance gap (less than 4.3%) over ZSP. The only exception was a notable 6.7% performance gap observed between CBERT and ZSP on PLV-Quadcode (Figure 3b). This difference can be attributed to the dataset's balanced and coarser-grained nature, which favors supervised learning.

However, supervised models experience a significant performance decline in more challenging fine-grained Rootcode classification (Figure 3c), emphasizing the need for sufficient and balanced annotation. Actually, our experience across multiple projects to develop event coding datasets with approximately 1,000 examples typically extends beyond several months. Creating evaluation datasets like PLV and AW, or even relabeling existing ones, proves to be far more time-consuming. In contrast, designing NLI prompts from the codebook for ZSP takes just a few days, greatly reducing annotation efforts and demonstrating clear advantages in real-world applications. Furthermore, ZSP's lack of a training phase significantly cuts down on GPU resource needs, enhancing its adaptability and enabling efficient inference on both CPUs and GPUs. This efficiency starkly contrasts with supervised models, which rely heavily on costly GPU resources for training.

We further analyzed ZSP's confusion matrix for Rootcode classification (see Figure 7 in Appendix F). The results reveal high ZSP accuracy by correctly classifying most Rootcodes, yet there are some misclassifications, particularly for AGREE, SUPPORT, AID, and YIELD labels. These labels have subtle semantic differences, with AGREE representing a future, verbal, or hypothetical version of the other three categories. For instance, consider the sentence labeled as diplomatic SUPPORT "... <S> had approved an agreement with <T> ...", ZSP produces conflicting predictions, with a score of 96.9% for the hypothesis "SUPPORT: approved an agreement" and 97.0% for "AGREE: agreed to sign an agreement". This discrepancy arises due to the fine distinction between these two labels, which even human annotators may find challenging.

ChatGPT. We observed notable differences in the performance of GPT-3.5 and the latest GPT-4 models. Specifically, GPT-3.5 exhibited inconsistent results. Despite excelling in binary tasks, it struggles with more specific labels and even performs worse than UP in Rootcode classification. These challenges align with previous research in similar tasks (Yuan et al., 2023; Cai and O'Connor,

2023; Li et al., 2023; Gao et al., 2023).

One ongoing challenge is generating formatted results and avoiding random labels outside the predefined set. To address this, we found that instructing GPT-3.5 to output digits (01-15) instead of text labels (AGREE - ASSAULT) partially alleviates these challenges and improves recall scores.

Another difficulty lies in effectively incorporating complex task descriptions and predefined label information into GPT-3.5. While our ZSP model can utilize class disambiguation rules easily, GPT-3.5 struggles to retain large amounts of information and may forget relevant details after just one round of chatting. This limitation necessitates the repetitive input of essential information in every interaction, which reduces efficiency.

Furthermore, balancing the preservation of necessary information and the compression of prompts to accommodate actual questions proves challenging. Continuous refinement of the prompts does not consistently improve performance, and it is counterintuitive that longer label descriptions with more disambiguation instructions result in performance decline. The quest for an optimal prompt design remains an open question for future research.

However, GPT-4 stands out as a significant improvement over GPT-3.5. It effectively reduces formatting errors, with few occasional issues lingering. The most significant enhancement is its ability to comprehend and process longer input tokens, allowing for better use of input information and finer class distinctions. Interestingly, class disambiguation notes were found to be effective for GPT-4 but not for GPT-3.5, further distinguishing the two models. The success of GPT-4 highlights the vast potential of LLMs. While extensive API queries can be costly, and precision may be slightly lower than ZSP, GPT-4's effectiveness with fewer prompts and superior generalization are notable advantages for future applications.

4.4 Ablation Study

We conducted an ablation study to address the following questions on ZSP: (1) Is a tree-query approach superior to a flat-query approach, which compares all hypotheses at single level simultaneously? (2) Does having more hypotheses guarantee better performance?

Table 5 displays the results of other zero-shot models, UP, GPT-3.5/4, and two variants of our ZSP models across multiple tasks. For the **Flat**-query approaches, the **Tiny** model uses 18 hypotheses derived from the Rootcode names (See Table 1). The **Full** model incorporates a complete list of 222 label descriptions from the codebook. The **Tree**-query approach consists of our ZSP model at different levels: l_1, l_2, and l_3.

Model		PLV Bin.	PLV Quad	PLV Root	A/W Bin.	Avg.
UP		80.8	51.8	46.3	67.2	61.5
GPT-3.5		90.1	66.2	40.9	76.3	68.4
GPT-4		93.4	76.7	61.5	87.0	79.7
ZSP Flat	Tiny	90.5	69.5	50.8	83.6	73.6
	Full	91.0	73.4	55.7	82.4	75.6
ZSP Tree	l_1	96.2	85.8	78.2	87.8	87.0
	$l_{1,2}$	**96.5**	87.6	79.4	87.8	87.8
	$l_{1,2,3}$	96.4	**89.6**	**82.4**	**88.0**	**89.1**

Table 5: Macro F1 scores% of ZSP with different settings vs. other zero-shot models in ablation study.

The observation that the Tiny model with 18 hypotheses outperforms UP with $81k$ inflexible patterns, confirms the effectiveness of generalized PLM features. Furthermore, Tiny surpasses GPT-3.5, highlighting the unreliability of GPT-3.5 and emphasizing the significance of expert knowledge in achieving superior results.

Despite the Tiny model's limited capacity to handle nuanced cases, adding more unorganized hypotheses does not consistently improve performance. The Full model's improper mixing and comparison of hypotheses for verbal and material events at different levels result in arbitrary NLI scores, leading to poor performance on PLV and inferior results compared to the Tiny model on A/W.

In contrast, the tree-query models outperform all flat-query models by a large margin at Level 1. Adding additional levels brings stable improvements, primarily for Quadcode and Rootcode. The tree-query framework effectively delimits the scope of candidate hypotheses and offers precise NLI scores that capture semantic differences. This ensures a more controllable and accurate result.

5 Conclusion

Future event coding tools should prioritize ease of interpretation and flexibility, making them more practical than annotating new datasets for black-box supervised models. Therefore, we explored the potential of zero-shot relation classification using ChatGPT (GPT-3.5/4) and introduced our ZSP model. While GPT-3.5 struggled with fine-grained

classification, GPT-4 showed promise in mitigating instability issues. Our ZSP offers an even more cheap, precise, and adaptable solution. The key is structuring the complex problem into an interpretable, three-level tree framework, integrating mode-aware NLI, and incorporating class disambiguation rules from the codebooks. Overall, our study highlights the value of integrating transferred knowledge with expert linguistic insights to streamline the process of verifying event records for the political science community.

6 Limitations

Balancing generalization and specificity is a common challenge across many methods. ZSP was developed to address the complexities of annotation codebooks and the inefficiencies in training annotators. This led to streamlined annotation, such as labeling an event as PROTEST instead of DEMAND for a protest related to rights, aided by the Conflict Override rule which simplifies complex annotation notes into machine-understandable rules. To maintain a balance between complexity and adaptability, we included only the most frequently used rules from the codebook.

Our approach's broader applicability, particularly in political science and related fields where codebooks are traditionally used to train annotators (Raleigh et al., 2010; Pavlick et al., 2016; Parolin et al., 2021), underscores the practicality of our method in streamlining data labeling. We are optimistic that combining established codebooks or knowledgebases with language models can extend to other domain-specific data. For instance, in legal studies, this approach can enhance the classification of legal documents by leveraging codified laws and regulations. In healthcare, it can aid in categorizing patient records and medical literature using clinical codebooks. In media studies and communication, it can support media content analysis by categorizing articles, broadcasts, and social media posts using thematic or sentiment-related codebooks.

However, challenges persist in zero-shot models when classifying semantically non-mutually exclusive fine-grained labels due to the intensive hypothesis engineering required. We addressed these challenges through the codebook's attainable expertise, but ZSP may struggle with tasks lacking accessible domain knowledge bases or those with overly nuanced and ambiguous labels. For example, classifying subcategories of ASSAULT (crime vs. attack vs. kidnap) or distinguishing peace protests from riots may require as many hypotheses as keywords (Barker et al., 2021; Radford, 2021). For such tasks, hybrid methods such as integrating ZSP or ChatGPT with few-shot learning, pattern-matching, or in-context learning could effectively address tasks of varying complexity, reducing human efforts. Future work will focus on exploring these hybrid methods.

Our exploration of ChatGPT models highlights the trade-off between generalizability and precision. While ChatGPT can adapt to any task with its chat-style format, it may sacrifice precision compared to NLI models. Due to time constraints and cost considerations, we did not investigate multi-turn interactions to enhance ChatGPT's precision, leaving this for future research. Nonetheless, both zero-shot models show promising potential to surpass traditional dictionary-based methods and annotation-driven supervised learning.

In selecting comparative methods, we included the most pertinent and recent ones that meet the specific needs of our task. Political event coding differs significantly from NLP event extraction tasks, such as those using the popular ACE dataset, in the availability of directly comparable studies. The prevailing methodologies in ACE event extraction are predominantly supervised and don't easily align with our unique ontology. Our investigation focuses on exploring the effectiveness of zero-shot models in event coding rather than achieving the highest accuracy through supervised approaches. Adapting existing zero-shot methods, designed with ACE contexts in mind, to our distinct requirements presents distinct challenges. Moving forward, we plan to expand our baseline comparisons in future studies.

While we initially considered expanding experiments to include other ontologies, we chose to focus on CAMEO/PLOVER, deferring broader explorations to future studies. This decision was influenced by practical constraints such as time and API costs, as well as a desire to pioneer within less-explored research domains. Unlike many widely-studied ontologies that rely on manually-created NLI systems and lack mode considerations, CAMEO/PLOVER presents unique challenges and opportunities. Its integration of mode features and codebooks makes it an ideal candidate for exploring PLMs in complex areas like political event

coding. By converting the complex expertise embedded in the codebooks into practical applications, we transcend the limits of conventional zero-shot modeling and showcase how PLMs like NLI and ChatGPT can be adapted to specialized domains.

7 Ethics Statement

The broad goal of producing accurate event data is to objectively measure and understand processes of political conflict and mediation around the world in order to prevent or mitigate their harm. We aim to produce a simple, flexible tool to serve this purpose. In particular, the zero-shot approach of this study largely reduces the costs, effort, and time to produce highly-quality event data on conflict, thus helping international and domestic government agencies, as well as researchers and practitioners to track, analyze, and mitigate the causes and effects of political violence. The project relied exclusively on news-story-like text as second-hand accounts of conflict events, but did not involve human research subjects.

Acknowledgments

The research reported herein was supported in part by NIST Award # 60NANB23D007, NSF awards DMS-1737978, DGE-2039542, OAC-1828467, OAC-1931541, OAC-2311142, and DGE-1906630, ONR awards N00014-17-1-2995 and N00014-20-1-2738, and the National Center for Transportation Cybersecurity and Resiliency (TraCR).

References

Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2023. Can we trust the evaluation on chatgpt? *arXiv preprint arXiv:2303.12767*.

Edward E Azar. 1980. The conflict and peace data bank (copdab) project. *Journal of Conflict Resolution*, 24(1):143–152.

Ken Barker, Parul Awasthy, Jian Ni, and Radu Florian. 2021. IBM MNLP IE at CASE 2021 task 2: NLI reranking for zero-shot text classification. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 193–202, Online. Association for Computational Linguistics.

Doug Bond, Joe Bond, Churl Oh, Craig J. Jenkins, and Charles L. Taylor. 2003. Integrated Data for Events Analysis (IDEA): An Event Typology for Automated Events Data Development. *Journal of Peace Research*, 40(6):733–745.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2016. ICEWS Coded Event Data.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Patrick T Brandt, John R Freeman, Tse-min Lin, and Phillip A Schrodt. 2013. Forecasting conflict in the cross-straits: long term and short term predictions. In *Annual Meeting of the American Political Science Association*.

Patrick T Brandt, John R Freeman, and Philip A Schrodt. 2011. Real time, time series forecasting of inter-and intra-state political conflict. *Conflict Management and Peace Science*, 28(1):41–64.

Patrick T Brandt, John R Freeman, and Philip A Schrodt. 2014. Evaluating forecasts of political conflict dynamics. *International Journal of Forecasting*, 30(4):944–962.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Berfu Büyüköz, Ali Hürriyetoğlu, and Arzucan Özgür. 2020. Analyzing ELMo and DistilBERT on socio-political news classification. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 9–18, Marseille, France. European Language Resources Association (ELRA).

Erica Cai and Brendan O'Connor. 2023. A monte carlo language model pipeline for zero-shot sociopolitical event extraction. *arXiv preprint arXiv:2305.15051*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Xinya Du and Claire Cardie. 2020. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online. Association for Computational Linguistics.

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

Steven Fincke, Shantanu Agarwal, Scott Miller, and Elizabeth Boschee. 2022. Language model priming for cross-lingual event extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10627–10635.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction. *arXiv preprint arXiv:2303.03836*.

Yuxia Geng, Jiaoyan Chen, Zhuo Chen, Jeff Z Pan, Zhiquan Ye, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2021. Ontozsl: Ontology-enhanced zero-shot learning. In *Proceedings of the Web Conference 2021*, pages 3325–3336.

Deborah J Gerner, Philip A Schrodt, Omür Yilmaz, and Rajaa Abu-Jabr. 2002. Conflict and mediation event observations (cameo): A new event data framework for the analysis of foreign policy interactions. *International Studies Association, New Orleans*.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2017. Cross-lingual classification of topics in political texts. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 42–46, Vancouver, Canada. Association for Computational Linguistics.

Sadaf MD Halim, Saquib Irtiza, Yibo Hu, Latifur Khan, and Bhavani Thuraisingham. 2023. Wokegpt: Improving counterspeech generation against online hate speech by intelligently augmenting datasets using a novel metric. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE.

Andrew Halterman and Benjamin J Radford. 2021. Few-shot upsampling for protest size detection. *Findings of ACL*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Jacek Haneczok, Guillaume Jacquet, Jakub Piskorski, and Nicolas Stefanovitch. 2021. Fine-grained event classification in news-like text snippets-shared task 2, case 2021. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 179–192.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv preprint arXiv:1911.10422*.

Yibo Hu, MohammadSaleh Hosseini, Erick Skorupa Parolin, Javier Osorio, Latifur Khan, Patrick Brandt, and Vito D'Orazio. 2022a. Conflibert: A pre-trained language model for political conflict and violence. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5469–5482.

Yibo Hu, Yu Lin, Erick Skorupa Parolin, Latifur Khan, and Kevin Hamlen. 2022b. Controllable fake document infilling for cyber deception. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6505–6519, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170.

Ali Hürriyetoğlu, Hristo Tanev, Vanni Zavarella, Jakub Piskorski, Reyyan Yeniterzi, Deniz Yuret, and Aline Villavicencio. 2021. Challenges and applications of automated extraction of socio-political events from text (case 2021): Workshop and shared task report. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 1–9.

Yiqiao Jin, Mohit Chandra, Gaurav Verma, Yibo Hu, Munmun De Choudhury, and Srijan Kumar. 2024. Better to ask in english: Cross-lingual evaluation of large language models for healthcare queries. In *Proceedings of the ACM on Web Conference 2024*, pages 2627–2638.

Daniel M Jones, Stuart A Bremer, and J David Singer. 1996. Militarized interstate disputes, 1816–1992: Rationale, coding rules, and empirical patterns. *Conflict Management and Peace Science*, 15(2):163–213.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. pages 7871–7880.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*.

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

J. Lu and Joydeep Roy. 2017. Universal petrarch: Language-agnostic political event coding using universal dependencies. Available at https://github.com/openeventdata/UniversalPetrarch (2020/05/22).

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Conference on Empirical Methods in Natural Language Processing*.

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

Charles McClelland. 1978. World event/interaction survey, 1966-1978. *WEIS Codebook ICPSR*, 5211.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017.

Teruko Mitamura and Eduard Hovy. 2015. Tac kbp event detection and coreference tasks for english.

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 72–78, Brussels, Belgium. Association for Computational Linguistics.

Fredrik Olsson, Magnus Sahlgren, Fehmi ben Abdesslem, Ariel Ekgren, and Kristine Eck. 2020. Text categorization for conflict event annotation. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 19–25, Marseille, France. European Language Resources Association (ELRA).

Open Event Data Alliance. 2018. Political language ontology for verifiable event records. https://github.com/openeventdata/PLOVER. Accessed: 2022-10-01.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Faik Kerem Örs, Süveyda Yeniterzi, and Reyyan Yeniterzi. 2020. Event clustering within news articles. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 63–68, Marseille, France. European Language Resources Association (ELRA).

Javier Osorio and Alejandro Beltrán. 2020. Enhancing the Detection of Criminal Organizations in Mexico using ML and NLP. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. Glasgow, Scottland.

Javier Osorio, Mohamed Mohamed, Viveca Pavon, and Brewer-Osorio Susan. 2019. Mapping Violent Presence of Armed Actors. *Advances in Cartography in GIScience of the International Cartographic Association*, pages 1–16.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Frank Robert Palmer. 2001. *Mood and modality*. Cambridge university press.

Erick Skorupa Parolin, MohammadSaleh Hosseini, Yibo Hu, Latifur Khan, Patrick T Brandt, Javier Osorio, and Vito D'Orazio. 2022a. Multi-coped: A multilingual multi-task approach for coding political event data on conflict and mediation domain. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 700–711.

Erick Skorupa Parolin, Yibo Hu, Latifur Khan, Patrick T Brandt, Javier Osorio, and Vito D'Orazio. 2022b. Confli-t5: An autoprompt pipeline for conflict related text augmentation. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1906–1913. IEEE.

Erick Skorupa Parolin, Latifur Khan, Javier Osorio, Patrick T Brandt, Vito D'Orazio, and Jennifer Holmes. 2021. 3M-Transformers for Event Coding on Organized Crime Domain. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE.

Erick Skorupa Parolin, Latifur Khan, Javier Osorio, Vito D'Orazio, Patrick T Brandt, and Jennifer Holmes. 2020. Hanke: Hierarchical attention networks for knowledge extraction in political science domain. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 410–419. IEEE.

Ellie Pavlick, Heng Ji, Xiaoman Pan, and Chris Callison-Burch. 2016. The gun violence database: A new task and data set for nlp. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1018–1024.

Valentina Pyatkin, Shoval Sadde, Aynat Rubinstein, Paul Portner, and Reut Tsarfaty. 2021. The possible, the plausible, and the desirable: Event-based modality detection for language processing. *arXiv preprint arXiv:2106.08037*.

Benjamin J. Radford. 2021. CASE 2021 task 2: Zero-shot classification of fine-grained sociopolitical events with transformer models. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 203–207, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Clionadh Raleigh, Andrew Linke, Håvard Hegre, and Joakim Karlsen. 2010. Introducing acled: an armed conflict location and event dataset: special data feature. *Journal of peace research*, 47(5):651–660.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 731–744.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero-and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227–268.

Philip A Schrodt. 1997. Early warning of conflict in southern lebanon using hidden markov models. In *American Political Science Association*.

Philip A Schrodt. 2006a. Forecasting conflict in the balkans using hidden markov models. In *Programming for peace*, pages 161–184. Springer.

Philip A. Schrodt. 2006b. Twenty Years of the Kansas Event Data System Project. *The Political Methodologist*, 14(1):2–6.

Philip A Schrodt. 2011. Forecasting political conflict in asia using latent dirichlet allocation models. In *Annual meeting of the European political science association, Dublin*.

Philip A Schrodt and Deborah J Gerner. 1996. *Using cluster analysis to derive early warning indicators for political change in the Middle East, 1979-1996*. University of Kansas.

Philip A Schrodt, Deborah J Gerner, and Omur Yilmaz. 2004. Using event data to monitor contemporary conflict in the israel-palestine dyad. *International Studies Association, Montreal, Quebec, Canada*, pages 1–31.

Philip A Schrodt, Ömür Yilmaz, and Deborah J Gerner. 2003. Evaluating "ripeness" and "hurting stalemate" in mediated international conflicts: An event data study of the middle east, balkans, and west africa. In *Annual Meeting of the International Studies Association, Portland, OR, February (eventdata. parusanalytics. com/papers. dir/Schrodt. etal. ISA03. pdf)*.

Robert Shearer. 2007. Forecasting israeli-palestinian conflict with hidden markov models. *Military Operations Research*, pages 5–15.

Stephen M Shellman and Brandon M Stewart. 2007. Predicting risk factors associated with forced migration: An early warning model of haitian flight. *Civil Wars*, 9(2):174–199.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. In *North American Chapter of the Association for Computational Linguistics*.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. pages 3914–3923.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot temporal relation extraction with chatgpt. *arXiv preprint arXiv:2304.05454*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*.

A Mode Design and Mapping

PLOVER suggests auxiliary modes to indicate whether a reported event is historical, future-oriented, hypothetical, or negated, as shown in Table 6. Some event types can theoretically combine with an auxiliary mode, such as AGREE becoming SUPPORT + future or THREATEN becoming ASSAULT + hypothetical. However, PLOVER's guidance lacks concrete implementation for annotators, merely assuming that "the coding engine will be able to resolve these and put that information in the context."

Moreover, while there are overlaps between PLOVER's auxiliary modes and the field of linguistic modality in NLP (Palmer, 2001; Saurí and Pustejovsky, 2009; Rudinger et al., 2018; Pyatkin et al., 2021), notable differences exist. For instance, Pyatkin et al. (2021) explore modes like event plausibility, which partially echoes aspects of political actors' intentions and event factuality in PLOVER. However, these explorations, though relevant, lack the precision and simplicity needed for direct application in PLOVER's context. Our focus, therefore, is on a simplified, practical, and task-specific mode framework for PLOVER.

Our proposed mode for PLOVER only consider four types: Past (**P**), Future (**F**), Contradict_Past (**CP**), Contradict_Future (**CF**). These modes were derived from our examination of the CAMEO/PLOVER ontology and PLOVER's auxiliary modes from the PLOVER codebook. Within this framework, we make a clear distinction between verbal, future or hypothetical events (Future) and historical or ongoing events (Past). And considering contradiction, we arrived at a simple 2x2 matrix with four modes outlined in Table 7. The table simplifies event coding and aids in accurately assigning Rootcode and Quadcode when an event's mode changes.

Specifically, Past covers historically significant or ongoing events, often presented in past tense but not restricted to it. Future includes verbal, hypothetical or future events. We consolidate hypothetical and future auxiliary modes in Table 6 because their similar nature in transitions between material and verbal events. For instance, THREATEN (Verbal Conflict, e.g., threatening to attack) can be considered either hypothetical or future ASSAULT (Material Conflict). Contradict_Past and Contradict_Future encompass events contradicting Past or Future occurrences, respectively. As illustrated in Table 3, CF and CP may include words with contradictory meanings, not necessarily containing negation words like "do not." Here, NLI's ability to identify negation allows us to focus on positive hypotheses with contradictory meanings, aligning with PLOVER's guideline to exclude negated events from datasets. Moreover, the codebook already provides mirrored hypotheses, eliminating the need for manual construction. For example, "YIELD: reduced protest against" is the CP of "PROTEST: protested against."

An additional observation in Table 7 is that verbal actions remain classified as verbal regardless of mode. In contrast, material actions are categorized differently based on their contradictory forms. For instance, the contradiction or negation of AGREE (e.g., "didn't agree to help") is always REJECT, Verbal Conflict. However, for material actions (e.g., "provided aid to"), its CP form (e.g., "stopped providing aid to") is SANCTION, Material Conflict, but its CF form (e.g., "would stop aid to") is REJECT, Verbal Conflict.

In sum, our task-specific mode concept aligns with PLOVER's auxiliary modes but enhances

Mode	Example
Historical	During the decolonization struggle, Angolan forces...
Future	Members of the G-7 will meet in Ottawa next month...
Hypothetical	If Russian forces were to cross the border, that would represent a major...
Negation	Thus far, fighting has not re-emerged in the tense region.

Table 6: Examples of PLOVER's auxiliary modes.

P	F	CP	CF
AGREE CONSULT SUPPORT 1	AGREE 1	REJECT 3	REJECT 3
COOPERATE AID YIELD 2	AGREE 1	SANCTION 4	REJECT 3
ACCUSE DEMAND REJECT THREATEN 3	ACCUSE DEMAND REJECT THREATEN 3	AGREE 1	AGREE 1
PROTEST MOBILIZE SANCTION COERCE ASSAULT 4	THREATEN 3	YIELD 2	AGREE 1

Table 7: PLOVER's labels (Rootcode text + Quadcode digits) w.r.t. our proposed modes: Past (**P**), Future (**F**), Contradict_Past (**CP**), Contradict_Future (**CF**).

Hypothesis	Label
<S> increased forces in <T>.	MOBILIZE 4
↑ override	
<S> increased peace forces in <T>.	AID 2
<S> retreated forces from <T>.	YIELD 2
↑ override	
<S> retreated peace forces from <T>.	SANCTION 4

Table 8: Examples of class disambiguation: We override forces if top predictions contain peace forces.

PLOVER's functionality, providing a practical, clear, and unambiguous approach to event coding.

B ZSP's Hypotheses and Class Disambiguation Rules

Table 16 shows the mode-aware hypotheses used in our experiments. We selected a subset of label descriptions in different Rootcode and Quadcode from the CAMEO codebook and converted these sentences to Past and Future modes. Some of them do not need Future variants as their labels from Past to Future remain the same, following Table 7.

Crafting disambiguation rules is a smaller part of our work compared to developing broad event modes. Event modes have resolved many cases, with custom rules providing fine-tuning. Our primary goal is to maintain a simple and generalizable model. Therefore, we only add commonly encountered rules like **Conflict Override**, which is prevalent in the CAMEO codebook and affects the coding for all conflict events.

Peace Override. As the second frequent case, classes related to "forces" vary according to actions and entities. For example, sending peacekeeping forces/workers/observers indicates cooperation, while sending forces to attack/occupy stands for conflict. Thus, we add hypotheses with "peace forces" distinct from normal "forces", as shown in Table 8. Predictions with "peace forces" have higher priority. I.e., we override "forces" if the top predictions contain "peace forces" because the latter one is more specified and infrequent. This simple rule ensures high recall for general forces and high precision for peace forces.

Consult Penalty. Another common issue found in CAMEO/PLOVER is the overly general CONSULT class (e.g., consult/talk/meet/visit). Many actions (e.g., sending forces, attacks, and investigations) entail that the source visited the target. Likewise, an accusation or threat indicates that the source talked or met with the target. One simple solution is to deduct the Consult Penalty, denoted as c (e.g., 2%), which penalizes the predicted entailment scores for the Rootcode "CONSULT".

Model		PLV Bin.	PLV Quad	PLV Root	A/W Bin.	Avg.
ZSP Flat	Tiny-c	89.7	68.9	49.5	81.0	72.3
	Tiny+c	90.5	69.5	50.8	83.6	73.6
	Full-c	89.4	70.8	53.1	75.9	72.3
	Full+c	91.0	73.4	55.7	82.4	75.6
ZSP Tree	l_1-c	95.6	85.1	77.3	85.4	85.9
	l_1+c	96.2	85.8	78.2	87.8	87.0
	$l_{1,2}$-c	96.0	87.0	78.7	85.5	86.8
	$l_{1,2}$+c	**96.5**	87.6	79.4	87.8	87.8
	$l_{1,2,3}$-c	95.9	89.0	81.8	85.5	88.1
	$l_{1,2,3}$+c	96.4	**89.6**	**82.4**	**88.0**	**89.1**

Table 9: Supplementary ablation study for Table 5. Macro F1 scores% of ZSP with $(+c)$ or without $(-c)$ Consult Penalty in different configurations.

We analyze the impact of c at every level in Table 9, with $(+c)$ indicating results with the penalty and $(-c)$ showing results without it. The effect of c is evident, with an average increase of 1.6% in macro F1 for all the tasks. For deeper levels, c ensures the accuracy of Level 1 predictions to avoid error propagation. These findings confirm the importance of preventing overly general and ambiguous hypotheses. Incorporating c provides a simple solution to alleviate manual efforts in curating alternative hypotheses.

Besides the three main disambiguation rules, users can easily add or tailor less-important rules for their specific study purposes, as discussed in Appendix C.

C Flexibility

The ZSP framework is notably flexible, easily accommodating changes in ontology or schema. Experts can swiftly update the ZSP method by modifying the hypothesis table or adjusting the class disambiguation rules to align with an evolving ontology. For example, if political scientists reclassify "arresting someone" from ACCUSE to COERCE (Table 7), they need only update the hypothesis label for "<S> arrested person of <T>". Similarly, introducing sub-categories within YIELD involves simple updates to the hypothesis labels.

Disambiguation rules, such as the Conflict Override rule, which prioritizes PROTEST over REQUEST in certain contexts, can also be refined. Transitioning to a multi-label approach is straightforward by eliminating the Conflict Override rule

Rootcode: 14-PROTEST **Code 144:** *Obstruct passage, block, not specified below* **Description:** Protest by blocking entry/exit into a building or area. **Usage Notes:** Use sub-categories if demands are known. Use this code for protests disrupting routine proceedings by blocking roads, buildings, etc. Use code 191 if the blockade involves military forces.
Rootcode: 19-COERCE **Code 191:** *Impose blockade, restrict movement* **Description:** Prevent entry/exit from a territory using armed forces. **Usage Notes:** Different from code 144, which refers to civilian protests.

Table 10: Examples of CAMEO classes with nuanced differences. Customized rules can differentiate these classes based on the actors involved.

to acknowledge both PROTEST and REQUEST as valid labels.

When should users write their own disambiguation rules? The need for custom rules depends on specific user requirements and the balance between manual effort and system precision and recall. Custom rules can be particularly beneficial for fine-grained analysis. For example, the CAMEO codebook includes similar classes "COERCE- Impose blockade" and "PROTEST- Obstruct passage/ blockade", as shown in Table 10. The key difference is whether the source is armed forces or protestors.

For researchers focusing on in-depth civil protest studies[6], distinguishing between codes 144 (civilian protests) and 191 (military blockades) is crucial for accurate classification. Thus, they can define a simple rule, **Blockade Override**, without additional cost: remove the hypothesis "COERCE-Impose blockade" if the top predictions contain PROTEST, indicating that the source is more likely protestors rather than armed forces. This adaptability showcases the model's flexibility and customizability in complex political scenarios.

While ChatGPT can be generalized to any task with their chat-style format, they may sacrifice precision compared with the ZSP model. Yet, both zero-shot models show promising applicability to surpass traditional dictionary-based methods or annotation-driven supervised learning methods.

D Building PLV and A/W Datasets

We extended existing resources to build our datasets, which is more efficient and effective than creating a new dataset from scratch. Table 11 summarizes the two datasets' detailed train and test split statistics. PLV is constructed from two resources. First, we outlined 335 examples (unique source-target pairs) with PLOVER Rootcode from the CAMEO **codebook**, and the PLOVER repository. Then we preprocessed a coarse-grained-labeled dataset from **CoPED** (Parolin et al., 2022a) and manually extended its Quadcode labels to 15 Rootcode in the new PLOVER schema. The major modification can be seen in Table 1. However, given that the current PLOVER codebook is in development, we leave YIELD without splitting it to CONCEDE and RETREAT. Finally, Figure 5 visualizes our final dataset's label distribution.

Dataset	Subset	# Docs	# S-T pairs	Tasks
PLV	CoPED	-	1043/698	Binary, Quadcode, Rootcode
	Codebook	-	0/335	
	Total	-	1050/1033	
A/W	ACE	337/338	432/451	Binary
	WikiEvents	91/92	370/434	
	Total	428/430	802/805	

Table 11: Statistics of the datasets: subsets, No. of documents and source-target pairs, and train/test splits.

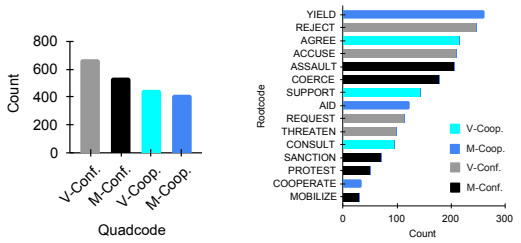
```
Conflict.attack: <arg1:attacker> attacked <arg2:target> using <arg3:instrument> at <arg4:place> place.
Justice.arrest: <arg1:jailer> arrested <arg2:detainee> for <arg3:crime> crime at <arg4:place> place.
```

Figure 4: Examples of templates in the A/W's original ontology (Li et al., 2021).

We built the A/W dataset from the ACE and WikiEvents datasets. First, the repository of (Li et al., 2021) provides templates for each event subtype of their ontology, enabling us to convert between different ontologies. For example, Figure 4 shows two frequent event types defined in the ontology. In both instances, argument 1 is equivalent to the source/actor, while argument 2 represents the target/recipient entities. Besides, the event type attack and arrest can be approximately mapped to ASSAULT and COERCE in PLOVER, respectively, as shown in Table 12.

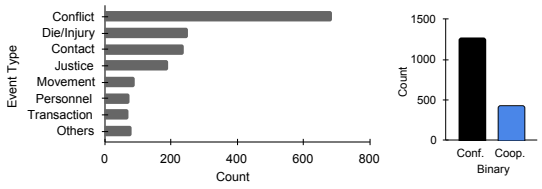
Therefore, we built labeled source-target pairs from ACE and WikiEvents. We extracted major sentences that contained the labeled entities from each long document in WikiEvents. We also removed entities that only consist of pronouns. Finally, we got 1258 valid sentences with 1687 labeled Source-Target pairs. To prevent label leaking, we split the dataset by document IDs, ensuring dis-

[6] https://github.com/emerging-welfare/glocongold.



(a) Quadcode (b) Rootcode

Figure 5: Extending PLV-Quadcode to Rootcode level.



(a) Event types statistics (b) Binary class

Figure 6: A/W dataset's original event types and its relabeled binary category.

tinct name entities for training and testing. Figure 6 shows the distribution of the original event types and the mapped binary class.

The nuanced differences between the two domains necessitate that event types be only "approximately" mapped to PLOVER Rootcodes. And extensive manual verification is needed to ensure accuracy. This complexity is rooted in the distinct focuses of NLP, which emphasizes predicates or topic-centric events, and Political Science, which concentrates on event status or mode. For instance, examples in Tables 2 (planned protest) and Table 3 (agreement to suspend protests) are both categorized as Conflict.Demonstrate in A/W, but in PLOVER, they are distinctly classified as THREATEN (verbal conflict) and AGREE (verbal cooperation), respectively. The binary labels even switch from conflict to cooperation in the second case. Thus, manual checking remains crucial even at the binary level.

The annotation process was carried out by two authors, achieving a Kappa score of 0.76, with discrepancies resolved through discussion.

E UP Experiment Setup

Universal Petrarch (UP) is a popular dictionary-based event coder (Lu and Roy, 2017). We adapted UP into our task of relation classification with gold source and target, i.e., source-target-action triplets. We found that UP is too strict and often results in

A/W Event Types	Approx. Root.	Binary
Life.Die/Injure	ASSAULT	Confli.
Conflict.Attack	ASSAULT	
Conflict.Demonstrate	PROTEST	
Justice	ACCUSE or COERCE	
Personnel.EndPosition	YIELD	Coop.
Contact	CONSULT	
Transaction	COOPERATE or AID	
Business.Merge-Org	COOPERATE	

Table 12: Mapping A/W's event types to PLOVER's approximate Rootcode and binary class.

incomplete or empty triplets. Thus, we reported the best possible result by the following methods. First, we used UP for each sentence to extract all possible events. Then we ranked the extracted triplets by the number of matched entities with gold sources and targets to decide the event code. We also counted the valid event code when there were no matched entities but only matched trigger action verbs. Even so, there are still 10% and 27% invalid event code results on PLV and A/W datasets, respectively. Finally, we mapped its output four-digit code to PLOVER Rootcode and Quadcode (similar to Figure 1).

F ZSP's Detailed Results Analysis

We examined the confusion matrix for ZSP on Binary (Figure 8), Quadcode (Figure 9), and Rootcode (Figure 7) classifications. The results show that ZSP perfectly classifies most contexts, with a slight degradation in differentiating mode (verbal vs. material).

In-depth class reports for PLV on Quadcode (Table 14) and Rootcode (Table 15) reveal that ZSP outperforms UP in nearly all metrics, except in the precision of the Verb-Conflict class (85.5%). However, UP's lower recall impacts its overall F1 score, showcasing the superiority of PLM's generalized knowledge over rigid pattern-matching approaches. Additionally, we noticed a performance trade-off when using overrides from Level 2 to Level 3. For instance, recall improves in Material-Conflict but decreases in Verbal-Conflict. Nevertheless, Level 3 significantly enhances overall F1 scores.

Further, we expand on the ablation study (Section 4.4), emphasizing why the tree-query approach, with fewer hypotheses, surpasses the "Full" model, which utilizes 222 flat hypotheses. Figure 10 illustrates the confusion matrix for the Full model's Rootcode classification. A comparison be-

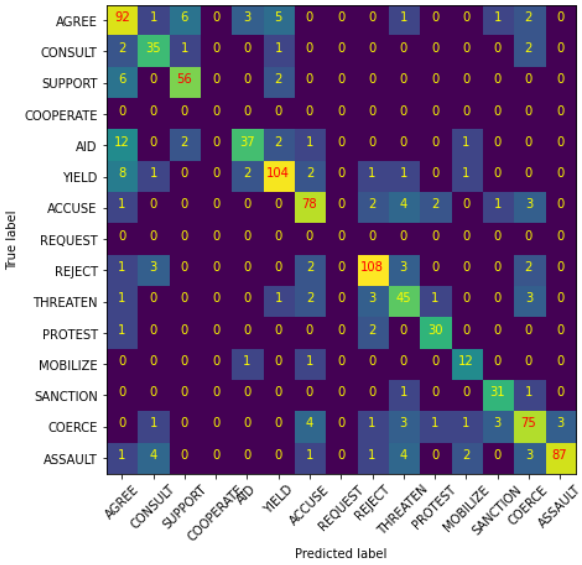


Figure 7: Confusion matrix for ZSP on PLV Rootcode.

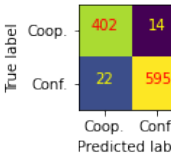


Figure 8: Confusion matrix for ZSP on PLV Binary code.

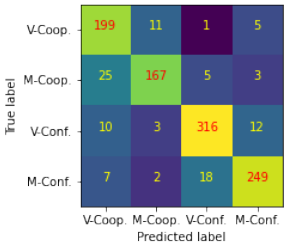


Figure 9: Confusion matrix for ZSP on PLV Quadcode.

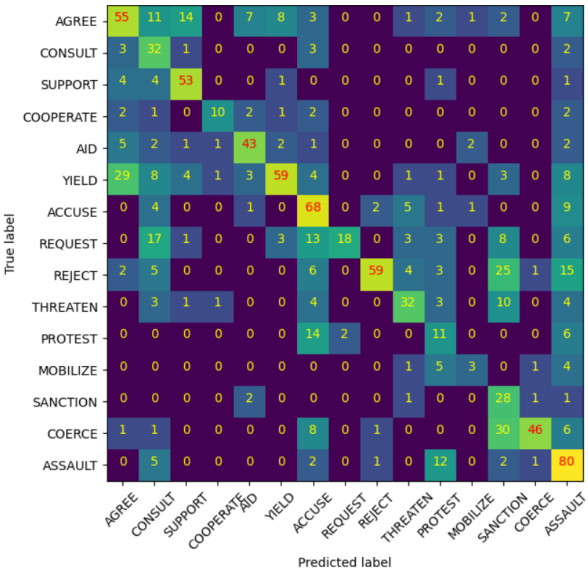


Figure 10: Confusion matrix on PLV Rootcode using the "Full" model in Section 4.4 Ablation Study.

Model	Size	PLV Bin.	PLV Quad	PLV Root	A/W Bin.	Avg.
base	125M	95.2	83.0	68.4	81.1	81.9
large	355M	96.4	89.6	82.4	88.0	89.1

Table 13: Macro F1 scores of ZSP models with different sized RoBERTa NLI models.

tween this matrix (Figure 10) and the default ZSP model using tree-query (Figure 7) reveals significant differences. The variable nature of NLI scores is a key factor in these differences. The tree-query model's focused approach on controlled hypothesis groups with consistent entities and predicates, but varying modes, leads to more accurate hypothesis identification. In contrast, the Full model's flat amalgamation of diverse hypotheses results in unpredictable outcomes and struggles with accurate mode classification, evident in frequent misclassifications between categories such as AGREE vs. SUPPORT, YIELD vs. AGREE, and REJECT vs. SANCTION or ASSAULT.

G NLI Model Selection

We selected RoBERTa-Large-MNLI[7] for its extensive usage in NLI research, with comparable alternatives like BART-Large-MNLI[8] also showing favorable results. Employing smaller-sized base models for zero-shot tasks is less common, primarily due to the significant drop in performance. Consequently, there are limited models specifically designed and widely accepted for zero-shot classification tasks.

From an efficiency perspective, employing large models for zero-shot tasks proves efficient as they are only required during the inference phase. Conversely, training supervise large models can be relatively expensive. Besides, one of our chosen baselines, CBERT (Hu et al., 2022a), only has a base version. Therefore, we conducted supervised experiments using base models while reserving large models exclusively for zero-shot tasks. This approach ensures a relatively fair and meaningful comparison between the two model types.

However, we also considered the possibility that a more rigorous comparison could have strengthened our hypotheses, particularly in demonstrating the effectiveness of smaller base models for handling fine-grained tasks in zero-shot scenarios. To explore this, we conducted experiments using an existing RoBERTa base model[9]. The results are

[7]https://huggingface.co/roberta-large-mnli
[8]https://huggingface.co/facebook/bart-large-mnli
[9]https://huggingface.co/cross-encoder/nli-roberta-base

Class	No.	Metrics	UP	ZSP		
				l_1	$l_{1,2}$	$l_{1,2,3}$
V-Coop.	216	Precison	63.1	**82.9**	**82.9**	82.6
		Recall	68.1	83.3	**92.1**	**92.1**
		Macro F1	65.5	83.1	**87.3**	87.1
M-Coop.	200	Precison	52.4	84.1	**91.7**	91.3
		Recall	60.5	**84.5**	83.0	83.5
		Macro F1	56.1	84.3	87.1	**87.2**
V-Conf.	341	Precison	85.5	85.9	85.9	**92.9**
		Recall	51.9	**94.4**	**94.4**	92.7
		Macro F1	64.6	89.9	89.9	**92.8**
M-Conf.	276	Precison	75.7	92.1	**93.2**	92.6
		Recall	69.9	80.1	80.1	**90.2**
		Macro F1	72.7	85.7	86.2	**91.4**
macro avg.	1033	Precison	55.3	86.2	88.4	**89.8**
		Recall	50.1	85.6	87.4	**89.6**
		Macro F1	51.8	85.8	87.6	**89.6**

Table 14: PLV Quadcode performance analysis.

Class	Precision	Recall	Macro F1	No.
AGREE	73.6	82.9	78.0	111
CONSULT	70.0	85.4	76.9	41
SUPPORT	84.8	87.5	86.2	64
COOPERATE	68.2	75.0	71.4	20
AID	82.2	62.7	71.2	59
YIELD	89.7	86.0	87.8	121
ACCUSE	82.1	85.7	83.9	91
REQUEST	96.8	84.7	90.4	72
REJECT	90.8	90.0	90.4	120
THREATEN	71.4	77.6	74.4	58
PROTEST	88.2	90.9	89.6	33
MOBILIZE	66.7	85.7	75.0	14
SANCTION	86.1	93.9	89.9	33
COERCE	82.4	80.6	81.5	93
ASSAULT	96.7	84.5	90.2	103
accuracy			83.8	1033
macro-avg.	82.0	83.5	82.4	1033

Table 15: PLV Rootcode performance analysis.

presented in Table 13, offering valuable additional insights alongside the findings presented in Table 4. While we observed that base models can effectively classify context or topics, they encountered challenges in distinguishing nuanced differences in mode. This distinction can lead to a drop in performance compared to larger models.

H ChatGPT Experiment Setup

Table 17 exemplifies inputs for relation classification tasks. Our task is characterized by challenging fine-grained classification that demands a substantial amount of input information. Due to token limitations and API costs, inputting one example at a time with a lengthy prompt is inefficient and costly. Instead, we used a long prompt followed by a list of input sentences to stay within the maximum token limits and obtain a list of predicted labels. More specifically, the inputs comprise the task and label description, a sentence list (usually limited to less than 50 sentences due to word constraints), and the task requirements. The anticipated output from the model is the predicted labels. Despite our repeated emphasis on ChatGPT generating only predefined labels, certain issues remain. To mitigate these, we use numerical codes (01-15) instead of text labels (AGREE - ASSAULT), reducing ChatGPT's generation of labels outside the predefined set. Additionally, we've noticed that ChatGPT tends to forget the task description and predefined label information, necessitating their input each time. Finally, refining the task and label description doesn't yield improved results. This underscores the complexity of the task, involving semantically non-mutually exclusive fine-grained labels, which proves challenging for ChatGPT.

Root.	Quad.	Past	Future
AGREE	V-Coop.	<S> agreed to do something for <T>	None
AGREE	V-Coop.	<S> promised to do something for <T>	None
CONSULT	V-Coop.	<S> held a talk with <T>	<S> agreed to hold a talk with <T>
CONSULT	V-Coop.	<S> met with <T>	<S> agreed to meet with <T>
CONSULT	V-Coop.	<S> undertook more negotiation with <T>	<S> agreed to undertake negotiation with <T>
SUPPORT	V-Coop.	<S> apologized to <T>	<S> agreed to apologize to <T>
SUPPORT	V-Coop.	<S> expressed support for <T>	<S> agreed to support <T>
SUPPORT	V-Coop.	<S> granted diplomatic recognition of <T>	<S> agreed to grant diplomatic recognition of <T>
SUPPORT	V-Coop.	<S> improved diplomatic cooperation with <T>	<S> agreed to improve diplomatic cooperation with <T>
SUPPORT	V-Coop.	<S> signed an agreement with <T>	<S> agreed to sign an agreement with <T>
AID	M-Coop.	<S> added aid to <T>	<S> agreed to provide aid to <T>
AID	M-Coop.	<S> added money to <T>	<S> agreed to add money to <T>
AID	M-Coop.	<S> granted asylum to <T>	<S> agreed to grant asylum to <T>
AID	M-Coop.	<S> increased peace forces in <T>	<S> agreed to increase peace forces in <T>
COOPERATE	M-Coop.	<S> cooperated with <T>	<S> agreed to cooperate with <T>
COOPERATE	M-Coop.	<S> extradited person to <T>	<S> agreed to extradite person to <T>
COOPERATE	M-Coop.	<S> shared information with <T>	<S> agreed to share information with <T>
YIELD	M-Coop.	<S> accepted demands of <T>	<S> promised to accept demands of <T>
YIELD	M-Coop.	<S> allowed entry of <T>	<S> promised to allow entry of <T>
YIELD	M-Coop.	<S> declared a ceasefire with <T>	<S> promised to a ceasefire with <T>
YIELD	M-Coop.	<S> eased restrictions on <T>	<S> promised to ease restrictions on <T>
YIELD	M-Coop.	<S> provided rights to <T>	<S> promised to provide rights to <T>
YIELD	M-Coop.	<S> reduced protest against <T>	<S> promised to reduce protest for <T>
YIELD	M-Coop.	<S> released person of <T>	<S> promised to release person of <T>
YIELD	M-Coop.	<S> resigned from the position in <T>	<S> promised to resign from the position in <T>
YIELD	M-Coop.	<S> retreated forces from <T>	<S> promised to retreat forces from <T>
YIELD	M-Coop.	<S> returned property of <T>	<S> promised to return property of <T>
YIELD	M-Coop.	<S> surrendered to <T>	<S> promised to surrender to <T>
YIELD	M-Coop.	<S> undertook reform in <T>	<S> promised to undertake reform in <T>
ACCUSE	V-Conf.	<S> accused <T> of something	None
ACCUSE	V-Conf.	<S> brought lawsuit against <T>	None
ACCUSE	V-Conf.	<S> expressed complaints of <T>	None
REQUEST	V-Conf.	<S> demanded something from <T>	None
INVESTIGATE	V-Conf.	<S> investigated something of <T>	<S> planned to investigate something of <T>
INVESTIGATE	V-Conf.	<S> sent people to investigate <T>	<S> planned to send people to investigate <T>
REJECT	V-Conf.	<S> defied laws of <T>	None
REJECT	V-Conf.	<S> rejected proposals of <T>	None
REJECT	V-Conf.	<S> rejected cooperation with <T>	None
REJECT	V-Conf.	<S> rejected to do something for <T>	None
REJECT	V-Conf.	<S> rejected to stop something against <T>	None
REJECT	V-Conf.	<S> rejected to consult with <T>	None
REJECT	V-Conf.	<S> rejected to yield to <T>	None
THREATEN	V-Conf.	<S> issued a ultimatum to <T>	None
THREATEN	V-Conf.	<S> threatened something against <T>	None
COERCE	M-Conf.	<S> arrested person of <T>	<S> threatened to arrest person of <T>
COERCE	M-Conf.	<S> attacked <T> cybernetically	<S> threatened to attack <T> cybernetically
COERCE	M-Conf.	<S> deported person of <T>	<S> threatened to deport person of <T>
COERCE	M-Conf.	<S> detained person of <T>	<S> threatened to detain person of <T>
COERCE	M-Conf.	<S> imposed blockades in <T>	<S> threatened to impose blockades in <T>
COERCE	M-Conf.	<S> imposed state of emergency in <T>	<S> threatened to impose state of emergency in <T>
COERCE	M-Conf.	<S> imposed more restrictions on <T>	<S> threatened to impose restrictions on <T>
COERCE	M-Conf.	<S> repressed person of <T>	<S> threatened to repress person of <T>
COERCE	M-Conf.	<S> seized property of <T>	<S> threatened to seize property of <T>
ASSAULT	M-Conf.	<S> seized territory of <T>	<S> threatened to seize territory of <T>
ASSAULT	M-Conf.	<S> assaulted person of <T>	<S> threatened to assault person of <T>
ASSAULT	M-Conf.	<S> destropyed property of <T>	<S> threatened to destropy property of <T>
ASSAULT	M-Conf.	<S> killed person of <T>	<S> threatened to kill person of <T>
ASSAULT	M-Conf.	<S> launched military strikes against <T>	<S> threatened to launch military strikes against <T>
ASSAULT	M-Conf.	<S> violated ceasefire with <T>	<S> threatened to violate ceasefire with <T>
FIGHT	M-Conf.	<S> attempted to assassinate <T>	None
FIGHT	M-Conf.	<S> used person of <T> as human shield	None
FIGHT	M-Conf.	Explosives in <S> attacked <T>	None
MOBILIZE	M-Conf.	<S> increased forces in <T>	<S> threatened to increase forces in <T>
MOBILIZE	M-Conf.	<S> kept alert in <T>	<S> threatened to keep alert in <T>
MOBILIZE	M-Conf.	<S> prepared forces against <T>	<S> threatened to prepare forces against <T>
PROTEST	M-Conf.	<S> launched protests against <T>	<S> threatened to launch protests against <T>
PROTEST	M-Conf.	<S> launched protests in <T>	<S> threatened to launch protests in <T>
PROTEST	M-Conf.	<S> protestors obstructed roads against <T>	<S> protestors threatened to obstruct roads against <T>
PROTEST	M-Conf.	<S> undertook boycotts against <T>	<S> threatened to undertake boycott against <T>
SANCTION	M-Conf.	<S> discontinued cooperation with <T>	<S> threatened to discontinue cooperation with <T>
SANCTION	M-Conf.	<S> expelled diplomatic people of <T>	<S> threatened to expel diplomatic people of <T>
SANCTION	M-Conf.	<S> expelled organizations of <T>	<S> threatened to expel organizations of <T>
SANCTION	M-Conf.	<S> expelled peacekeepers of <T>	<S> threatened to expel peacekeepers of <T>
SANCTION	M-Conf.	<S> halted negotiations with <T>	<S> threatened to halt negotiate with <T>
SANCTION	M-Conf.	<S> reduced aid to <T>	<S> threatened to reduce aid to <T>
SANCTION	M-Conf.	<S> retreated peace forces from <T>	<S> threatened to retreat peace forces from <T>

Table 16: The mode-aware hypothesis table considering Past and Future modes. <S> and <T> represent the source and the target entities in practical examples. Some hypotheses do not require Future variants as their labels (Rootcode and Quadcode) remain unchanged from Past to Future, as indicated in Table 7.

Relation Extraction (RE) Task is to classify the political relations between a source (indicated by <S></S>) and a target (indicated by <T></T>) within a given input sentence. The goal is to assign these relations into a predefined set of labels. The predefined set of relation labels 1-15 is as follows. The relations can be categorized into four quadrants: Q1(Verbal Cooperation), Q2 (Material Cooperation), Q3 (Verbal Conflict), and Q4 (Material Conflict).

1. AGREE, Q1: Agree to, offer, promise, or otherwise indicate willingness or commitment to cooperate, including promises to sign or ratify agreements. Cooperative actions (CONSULT, SUPPORT, COOPERATE, AID, YIELD) reported in future tense are also taken to imply intentions and should be coded as AGREE.
2. CONSULT, Q1: All consultations and meetings, including visiting and hosting visits, meeting at neutral location, and consultation by phone or other media.
3. SUPPORT, Q1: Initiate, resume, improve, or expand diplomatic, non-material cooperation; express support for, commend, approve policy, action, or actor, or ratify, sign, or finalize an agreement or treaty.
4. COOPERATE, Q2: Initiate, resume, improve, or expand mutual material cooperation or exchange, including economics, military, judicial matters, and sharing of intelligence.
5. AID, Q2: All provisions of providing material aid whose material benefits primarily accrue to the recipient, including monetary, military,humanitarian, asylum etc.
6. YIELD, Q2: yieldings or concessions, such as resignations of government officials, easing of legal restrictions, the release of prisoners, repatriation of refugees or property, allowing third party access, disarming militarily, implementing a ceasefire, and a military retreat.
7. REQUEST, Q3: All verbal requests, demands, and orders, which are less forceful than threats and potentially carry less serious repercussions. Demands that take the form of demonstrations, protests, etc. are coded as PROTEST.
8. ACCUSE, Q3: Express disapprovals, objections, and complaints; condemn, decry a policy or an action; criticize, defame, denigrate responsible parties. Accuse, allege, or charge, both judicially and informally. Sue or bring to court. Investigations.
9. REJECT, Q3: All rejections and refusals, such as assistance, changes in policy, yielding, or meetings.
10. THREATEN, Q3: All threats, coercive or forceful warnings with serious potential repercussions. Threats are generally verbal acts except for purely symbolic material actions such as having an unarmed group place a flag on some territory.
11. PROTEST, Q4: All civilian demonstrations and other collective actions carried out as protests against the recipient: Dissent collectively, publicly show negative feelings or opinions; rally, gather to protest a policy, action, or actor(s).
12. SANCTION, Q4: All reductions in existing, routine, or cooperative relations. For example, withdrawing or discontinuing diplomatic, commercial, or material exchanges.
13. MOBILIZE, Q4: All military or police moves that fall short of the actual use of force. This category is different from ASSAULT, which refers to actual uses of force, while military posturing falls short of actual use of force and is typically a demonstration of military capabilities and readiness. MOBILIZE is also distinct from THREAT in that the latter is typically verbal, and does not involve any activity that is undertaken to demonstrate military power.
14. COERCE, Q4: Repression, restrictions on rights, or coercive uses of power falling short of violence, such as arresting, deporting, banning individuals, imposing curfew, imposing restrictions on political freedoms or movement, conducting cyber attacks, etc.
15. ASSAULT, Q4: Deliberate actions which can potentially result in substantial physical harm.

Note that we give priority to labels in Material Conflict over Verbal Conflict. For example, we label "protest to request" as material PROTEST other than verbal REQUEST. Similarly, we label "convict and arrest" as material COERCE other than verbal ACCUSE, considering the more severe actions involved.

Input and Task Requirement:
Perform the RE task for the given input list and print the output with columns (No., Label, Quadrants) split by the tab delimiter. Use 1-15 to denote the predefined labels above (1. AGREE, 2. CONSULT, 3. SUPPORT, 4. COOPERATE, 5. AID, 6. YIELD, 7. REQUEST, 8. ACCUSE, 9. REJECT, 10. THREATEN, 11. PROTEST, 12. SANCTION, 13. MOBILIZE, 14. COERCE, and 15. ASSAULT).

No.	Sentence
1	<S>A Brazilian federal court</S> has rejected a request from <T>jailed former President Luiz Inacio Lula da Silva</T> to be present at the first debate of presidential candidates for October's election.
2	<S>Afghan rebels</S> have kidnapped up to 16 <T>Soviet civilian advisers</T> from a town bazaar and exploded a series of bombs in the capital Kabul, western diplomatic sources in neighboring Pakistan said today.
3	<S>A local Taliban leader and his five associates</S> have given up fighting and surrendered in <T>Afghanistan's northern Faryab province</T>, an army source said Tuesday.
4	<S>French National Assembly president Laurent Fabius</S> and a group of deputies held talks with leaders of<T>Romania's</T> new government on Tuesday, the first high level Western delegation to visit Bucharest since last month's revolution.

Output:

No.	Label	Quadrants	Correct?
1	9 (REJECT)	Q3: Verbal Conflict	✓
2	15 (ASSAULT)	Q4: Material Conflict	✓
3	6 (YIELD)	Q2: Material Cooperation	✓
4	2 (CONSULT)	Q1: Verbal Cooperation	✓

Table 17: Input and Output of ChatGPT.