FAMuS: Frames Across Multiple Sources

Siddharth Vashishtha[1] **Alexander Martin**[1] **William Gantt**[1]
Benjamin Van Durme[2] **Aaron Steven White**[1]
[1] University of Rochester [2] Johns Hopkins University
{svashis3@cs.|wgantt@cs.|amart50@u.|aaron.white@}rochester.edu

Abstract

Understanding event descriptions is a central aspect of language processing, but current approaches focus overwhelmingly on single sentences or documents. Aggregating information about an event *across documents* can offer a much richer understanding. To this end, we present FAMuS, a new corpus of Wikipedia passages that *report* on some event, paired with underlying, genre-diverse (non-Wikipedia) *source* articles for the same event. Events and (cross-sentence) arguments in both report and source are annotated against FrameNet, providing broad coverage of different event types. We present results on two key event understanding tasks enabled by FAMuS: *source validation*—determining whether a document is a valid source for a target report event—and *cross-document argument extraction*—full-document argument extraction for a target event from both its report and the correct source article.

1 Introduction

Recent years have witnessed a resurgence of interest in document-level event and argument extraction tasks, such as *template filling* (Du et al., 2021b; Chen et al., 2023b; Gantt et al., 2022), *role-filler entity extraction* (Du et al., 2021a; Huang et al., 2021), and *event argument extraction* (Ebner et al., 2020; Li et al., 2021; Tong et al., 2022). Indeed, the earliest goals of information extraction (IE), as advanced by the Message Understanding Conferences (MUCs), were to develop systems capable of extracting document-level event structures (Grishman and Sundheim, 1996; Grishman, 2019). While the renewed interest in these goals represents clear progress beyond the longstanding and dominant focus on *sentence-level* event extraction, recent work in this area suffers from two key shortcomings.

For one, major benchmarks on these tasks, including MUC-4 (muc, 1992), RAMS (Ebner et al., 2020), WikiEvents (Li et al., 2021), and DocEE (Tong et al., 2022) feature highly domain-specific event ontologies. Even when the absolute number of types is relatively large (e.g. the 139 event types covered by RAMS), they tend to be tightly clustered within a small handful of categories.

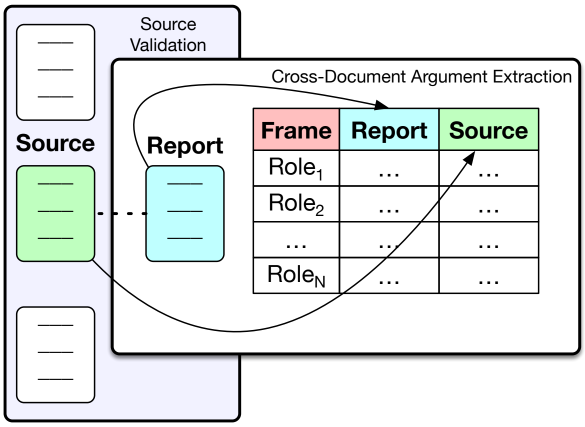

Figure 1: Schematic of the two FAMuS tasks: source validation and cross-document argument extraction.

For another, although whole-document extraction enables a richer understanding of an event than its sentence-level analogue, it is still constrained by the input document's description of that event, which may lack key details. The task of *event linking* partly remedies this by linking event mentions to a canonical entry in a knowledge base, but stops there, providing no actual extractions from those entries (Nothman et al., 2012; Yu et al., 2023).

This work introduces **FAMuS** (**F**rames **A**cross **Mu**ltiple **S**ources), a dataset and benchmark aimed at addressing both of these shortcomings. FAMuS provides event and cross-sentence argument annotations on over 1,255 Wikipedia passages (or *reports*), each paired with cross-sentence argument annotations for the *same* event as described in the document cited as the passage's *source*. Events and arguments are annotated against FrameNet (Baker et al., 1998), providing genuinely broad coverage with 253 diverse event types and five supporting documents per type. Beyond the dataset itself, we

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8250–8273
June 16-21, 2024

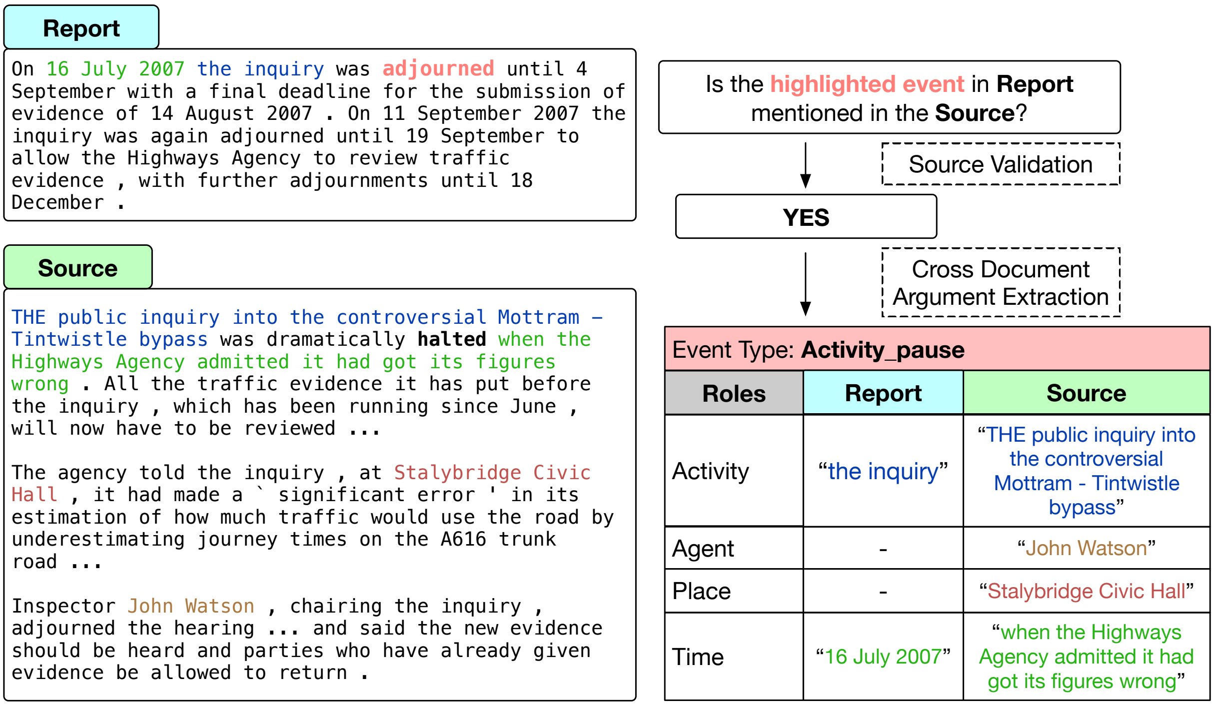

Figure 2: An example from FAMuS. The **Source Validation** task asks whether the event denoted by the trigger highlighted in the report text (*adjourned*) is also described in the source text. If so, the system must then identify and extract all arguments of that event in *both* the report and the source in the **Cross-Document Argument Extraction** task. FAMuS contains genre-diverse (report, source) pairs selected from the MegaWika dataset (Barham et al., 2023) and annotates a single target event trigger in the report, along with all arguments in both report and source, against FrameNet (Baker et al., 1998), enabling broad coverage of different event types.

make the following further contributions:

- We introduce a novel cross-document objective (Figure 1), supported by FAMuS, comprising two challenging tasks: (i) **Source Validation**, which requires determining whether an input document is a valid *source* for a tagged event in a given report; and (ii) **Cross-Document Argument Extraction**, which requires extracting arguments for a tagged report event from *both* the report and its source.
- We present results from a diverse suite of models on both tasks, including heuristic baselines, fine-tuned models using off-the-shelf encoders, and few-shot LLMs.
- We propose a new evaluation metric for argument extraction that computes an edit distance-based soft match between predicted and reference arguments to provide a richer picture of systems' argument extraction performance than traditional exact match.

The FAMuS dataset and baselines are available at https://github.com/FACTSlab/FAMuS.

2 Task Definitions

To situate FAMuS in the context of prior work, we first give a formal statement of the tasks it presents:

1. **Source Validation (SV).** Given a *report text* R, a target event trigger (mention) e occurring in R, and a candidate *source text* S, determine whether S contains a description of the same event as the one denoted by e.
2. **Cross-Document Argument Extraction (CDAE).** Given a report text R, a target event trigger e in R, and a *correct* source text S, extract all arguments of e in both R and S. We assume e is assigned an event type from some underlying ontology of event types $E_1, \ldots E_N$, where each E_i has roles $R_1^{(i)}, \ldots, R_{M_i}^{(i)}$, and where e's arguments must each be assigned one of these roles.[1]

Both tasks are schematically depicted in Figure 1 and detailed in Figure 2. Collectively, these tasks formalize *informal* reading habits common to researchers and internet users: during reading, we

[1]Note that we do *not* require S to contain an explicit event trigger e' coreferent with e. We require only that S refers somehow to the event denoted by e, even if this reference is made more obliquely than with a single lexical item.

discover intriguing events and then we seek further details about them in other *relevant* sources.

3 Background

We are aware of no prior work that combines identification of a report event's source document (SV) with argument extraction from both the report and the source (CDAE). However, both closely relate to a number of established tasks in the literature, which we survey briefly below.

Event Linking (EL) or *event grounding* is the task of associating an event description (typically, a single mention) with a canonical entry for that event in some knowledge base. It resembles SV in attempting to ground a target event mention in a text to a more comprehensive description of the same event in a source text. But whereas SV takes a candidate source text as input (along with the report), EL aims to produce (a link to) one as output.

Introduced by Nothman et al. (2012) as an event-centric analogue to the more popular *entity linking* objective (Bunescu and Paşca, 2006; Ji and Grishman, 2011), EL has received comparatively little attention. While Nothman et al. used Australian news articles for both report and source, more recent efforts have focused on Wikipedia and Wikidata. Yu et al. (2023) use Wikipedia articles as source documents and present evaluations with both Wikipedia and New York Times report articles. Ou et al. (2023), extending work by Pratapa et al. (2022), propose an interesting hierarchical variant of the task, in which mentions must be linked to a *set* of hierarchically related events in WikiData.

Cross-Document Event Coreference (CDEC) involves identifying all coreferring event mentions across a collection of documents (Bagga and Baldwin, 1999). Various benchmarks exist for the task, including ECB+ (Cybulska and Vossen, 2014), MEANTIME (Minard et al., 2016), the Gun Violence Corpus (GVC; Vossen et al., 2018b), and WEC (Eirew et al., 2021, 2022). From one angle, CDEC can be viewed as a kind of generalization of EL, insofar as the latter is concerned only with matching up *pairs* of documents that describe the same event, and the former with matching up (potentially) multiple. However, CDEC usually expressly clusters event *mentions*, whereas EL and SV often do not.

Claim Verification SV is also structurally similar to *fact* or *claim verification*, in which the goal is to determine whether some target statement (the *claim*) is supported, unverified, or refuted by a source text.[2] Notable benchmarks here include Emergent (Ferreira and Vlachos, 2016), the Fake News Challenge (Pomerleau and Rao, 2017), LIAR (Wang, 2017), and FEVER (Thorne et al., 2018). Although they are *structurally* similar, the underlying relations governing each task (event coreference and evidentiary support) are clearly distinct.

Event Argument Extraction (EAE) is a generalization of semantic role labeling (SRL; Gildea and Jurafsky, 2002) that additionally assigns roles to a predicate's extra-sentential arguments.[3] Our CDAE subtask is just EAE applied to both the report and the source texts. SemEval 2010 Task 10 (Ruppenhofer et al., 2010) and Beyond NomBank (Gerber and Chai, 2010) represent the first true benchmarks for EAE, with the former consisting of a set of Sherlock Holmes stories annotated against FrameNet, and the latter annotating the arguments of a set of 10 nominal predicates from NomBank (Meyers et al., 2004) on the Penn Tree Bank corpus (Marcus et al., 1993). Other resources include ONV5 (Moor et al., 2013) and MS-AMR (O'Gorman et al., 2018). Unfortunately, these datasets are all quite small: the largest, MS-AMR, still contains only about 2,400 implicit arguments. EAE has lately seen renewed interest due mainly to the much larger RAMS (Ebner et al., 2020) and WikiEvents (Li et al., 2021) benchmarks (20-30k arguments each). The more recent DocEE (Tong et al., 2022) benchmark is an order of magnitude larger still (180k arguments). One disadvantage of these three datasets relative to their predecessors, however, is their use of domain-specific ontologies. FAMuS aims to address both of the above issues by providing a relatively *large* dataset annotated against a *broad-coverage* ontology.

Predicate-Argument Alignment Related to CDAE (and CDEC), some prior work has studied cross-document alignment of predicate-argument structures. Roth and Frank (2012b), for instance, annotate gold predicate alignments in 70 pairs of topically related documents from GigaPairs (Roth and Frank, 2012a) and introduce a graph-based

[2] In some cases, the relevant evidentiary sentences from the source must also be provided.

[3] EAE is synonymous with *multi-sentence argument linking* and arguably also with *implicit semantic role labeling*, though exact task definitions differ. See O'Gorman (2019) and Gantt (2021) for surveys.

	Train	Dev
Event Types	253	253
Role Types (R)	712	580
Role Types (S)	749	643
SV Examples (+)	759	253
SV Examples (−)	759	253
Avg. Tokens (R)	59	60
Avg. Tokens (S)	1,084	1,511
Avg. Filled Roles (R)	2.97	3.45
Avg. Filled Roles (S)	3.45	3.89
Avg. Args (R)	3.07	3.55
Avg. Args (S)	3.70	4.28

Table 1: Summary statistics for the FAMuS train and dev splits (test deliberately omitted). "(R)" and "(S)" denote *report* and *source*, respectively. Note that CDAE examples (not shown) are the same as "SV Examples (+)," as these consist of the same documents (see §4).

clustering model for the task. Wolfe et al. (2013) present PARMA, a feature-rich, regularized logistic regression model for the same task that makes independent alignment decisions for each predicate and argument. While CDAE demands neither identification of a predicate in the source document nor an explicit argument-to-argument alignment, it is similar to this work in identifying aligned *sets* of arguments of the same event across documents.

4 Data Collection

The FAMuS documents represent a subset of English documents from MegaWika (Barham et al., 2023), a dataset comprising millions of (report, source) pairs across 50 languages. Below, we discuss data collection for our SV and CDAE tasks.

4.1 Source Validation

Overview Verifying the quality of a web page as the source for a given report text is imperative. Barham et al. (2023) introduced a source validation task where annotators determine the correct FrameNet frame for a tagged event in a report and assess if the corresponding source describes the same event as is tagged in the report. Barham et al. observe relatively low inter-annotator agreement on this task (Krippendorff's α of 0.41 (Krippendorff, 2018)), and just under half of their source documents were deemed valid.

Refining their approach, we only accept *positive* source validation (SV) examples when (i) at least two-thirds of annotators agree on the correct FrameNet frame, and (ii) all three annotators, or a two-thirds majority *plus one of the authors*, agree on the source's validity. Negative examples are identified through a combinatnion of manual and automated techniques, which are detailed below.

Positive Examples We prioritize *broad coverage* in event *types* and examples per type while balancing the trade-offs within a constrained annotation budget. Our methodology seeks an optimal compromise to meet these dual objectives. At a high level, we rely on the FrameNet inheritance hierarchy to identify a subset of 328 frames that denote a *situation*—i.e. an EVENT, STATE, or PROCESS in FrameNet.[4] We then iterate Barham et al.'s annotation protocol until we obtain at least *five* (report, source) pairs per frame that satisfy our two criteria—(i) and (ii) above—for positive examples, for at least 75% (250) of the 328 situation-denoting frames. We used Barham et al. (2023)'s oversampling technique with the LOME FrameNet parser (Xia et al., 2021) and a Longformer-based SV model (see §5) to estimate the number of annotations needed to secure five positive examples per frame. This estimation used a negative binomial model, considering the parser's precision and the SV model's accuracy, with adjustments for frames with limited test support. Through seven iterations, we ensured diversity in our annotations using stratified sampling and k-means clustering on SpanBERT embeddings of the source text, selecting varied report-source pairs within each frame category (refer to Appendix A for further details).

Negative Examples To build a balanced dataset for the SV task, we include five negative examples per frame. Most of these are taken from the annotated documents described above, provided all annotators unanimously agree they do not match the report event.

Some frames were short of five negative examples after the main annotation process. To address this, we supplemented the dataset with additional *silver* negative examples as needed. We also ensured that each example in the test set is either a gold standard example or has undergone manual platinum annotation by one of this paper's authors for the generated examples.[5]

Our method involves matching unannotated reports with a *new* source text that is semantically close but does not describe the same event, ensur-

[4] Details on the frame selection process are in Appendix D.

[5] The test set includes 11 platinum-annotated examples.

ing a challenging task. For each frame f_i, we take the same candidate example set c_i described above, remove annotated examples to form $c'_i \subset c_i$, and randomly choose a pair $(r_i^{(j)}, s_i^{(j)})$ from c'_i. We then find a pair $(r_i^{(k)}, s_i^{(k)})$ from the remaining set $c'_i - (r_i^{(j)}, s_i^{(j)})$ where $s_i^{(k)}$ is most similar to $r_i^{(j)}$ based on SimLM scores (Wang et al., 2023), creating a new negative example $(r_i^{(j)}, s_i^{(k)})$ with $j \neq k$. The chance that $s_i^{(k)}$ describes the same event as $r_i^{(j)}$ is low due to the vast number of sources. Additionally, distributions of report-source similarity scores for the positive examples and for these "silver" negative examples shown in Appendix E are quite divergent, underscoring this point.

Annotation Quality In our two-stage qualification for Amazon Mechanical Turk workers for the CDAE task (Section 4.2), successful candidates from the second phase joined the source validation task. Additionally, 11 new workers who matched the majority on ten gold-standard validations were added, totaling 26 workers for source validation, with each task triple-checked. Each Human Intelligence Task (HIT) consisted of one report-source pair and offered \$0.20.

Krippendorff's α for frame identification was 0.62, demonstrating reliable agreement, comparable or superior to other crowd-sourced tasks (Hong and Baker, 2011; Fossati et al., 2013; Vossen et al., 2018a, 2020; Dumitrache et al., 2018, and others). For source validation, all examples had either unanimous agreement or majority agreement with additional author approval for positive cases.

4.2 Cross-Document Argument Extraction

Overview In each round, after SV annotation, we collect *full-document*[6] role annotations on both the report and the (valid) source for the annotated report event. We annotate only the core roles of each frame, plus TIME and PLACE.[7] Here, annotators select roles from the role set for the report trigger's frame and then select a contiguous span from the report or source text as an argument for that role. The interface also supports annotating multiple arguments for the same role. When a role is selected, its FrameNet definition and an example are displayed. Annotators are strongly encouraged to annotate based on the highlighted frame (chosen during SV annotation) but are permitted to change the frame in the rare case they deem it incorrect. Of the 1,255 CDAE examples annotated, only 4.6% actually had their frame types changed—a testament to the high quality of the SV frame annotations.

While we do *not* annotate for coreference, we do provide model-predicted (*silver*) coreference clusters for all annotated arguments, which are used in one evaluation setting (see §5). We use F-COREF (Otmazgin et al., 2022) as the coreference model.

Annotation Quality We conducted 2 selection stages on Amazon Mechanical Turk for CDAE task annotators. Initially, candidates provided annotations for a short ($\sim$ 250-token) document, followed by a longer ($\sim$ 4k-token) document in the second phase. Two paper authors assessed their work, advancing only those who passed the first phase to the second. This resulted in 15 annotators for the main task, each paid \$1 per task with a bonus opportunity of up to \$4 for exceptional work.

To maintain high-quality annotations, we combined automatic checks with manual reviews. Post-initial annotation iteration, authors corrected all entries, comparing unedited with edited annotations using the metric in Appendix B, yielding a 0.94 F1 mean score for report annotations and a 0.92 F1 for source annotations, with many showing perfect agreement.

As manually reviewing *all* annotations was impractical, subsequent rounds used a hybrid verification approach. We compared ChatGPT predictions with Turker annotations, manually correcting only those in the lowest agreement quartile.[8] We removed some examples for poor document quality, like excessive non-English text.

5 Experiments

We now describe the models and setup for experiments on SV and CDAE. Model hyperparameters, prompts and details can be found in Appendix C.

5.1 Source Validation

Per §2, SV is a binary classification task that takes as input a report R, a (typed) event trigger $e \in R$, and a candidate source text S, and outputs a binary judgment indicating whether S contains a description of the same event as is denoted by e. We consider three models for this task, in addition to a majority-class baseline.

[6]In contrast to much prior work on EAE, we do not impose fixed-size context windows during annotation, allowing arguments to be annotated *anywhere* in the document.

[7]Annotation interface shown in Appendix A.

[8]See Appendix B for details.

Lemma Baseline This model simply predicts `YES` if the lemma of the report's event trigger exists in the (lemmatized) source, and `NO` otherwise. We use NLTK's `WordNetLemmatizer` to obtain lemmas (Bird et al., 2009).

Longformer We use Longformer (Beltagy et al., 2020) with a classification head and fine-tune it on FAMuS. The input sequence to the Longformer model is a `</s>`-delimited concatenation of the report and source text, with the report event's trigger marked by `<event>` tags.[9]

Few-Shot LLMs Inspired by the successes of recent large language models (LLMs) on many IE tasks (Wei et al., 2023), we also evaluate ChatGPT (`gpt-3.5-turbo-0301`) and Llama 2 (`llama-2-13b`; Touvron et al., 2023) on FAMuS in the few-shot setting. The prompt (which is the same for both models) describes the task and includes two positive and two negative examples handwritten by one of the authors. We set model temperature to 0 to ensure consistent generations.

5.2 Cross Document Argument Extraction

In CDAE, the input is a valid (report, source) pair, along with a (typed) event trigger in the report. The output is a set of arguments for the trigger, extracted from both report and source. Below, we present results on three CDAE models, training and evaluating each separately on report and source.

IterX The IterX model by (Chen et al., 2023b) sets a new benchmark in template filling, excelling on MUC-4 (Sundheim, 1992; muc, 1992) and SciREX (Jain et al., 2020) by approaching template prediction as autoregressive span assignment. IterX methodically assigns input spans roles within a template, updating candidate embeddings based on those assignments and repeating the process until all candidates are labeled null. Designed for multiple templates per document, we tailored IterX for CDAE to output one template each for sources and reports.

IterX operates with predefined candidate spans. Depending on the setting (see below), these spans are drawn from different subsets of the following three sources: (i) gold-standard CDAE annotations; (ii) LOME FrameNet parser arguments; (iii) Stanza's NER identified entities (Qi et al., 2020). Training and evaluation settings vary: **gold spans** uses only (i), **predicted spans** trains on all but tests on (ii) and (iii), and **gold and predicted spans** involves all three during both training and testing, reflecting the value of gold-span access.

For template type input, IterX uniquely integrates both frame type and lexical triggers from the CDAE input, using `<event>` tags to include triggers as input spans but assigns them a null role (ϵ). This incorporation leverages the Transformer encoder's self-attention (Vaswani et al., 2017) to condition each span's role on the trigger. Example inputs are shown in Appendix subsection C.2 (Figure 13). Following Chen et al., we use IterX with a T5-large encoder (Raffel et al., 2020).

Longformer QA Our second model recasts CDAE as extractive question answering (QA) in the style of SQuAD 2.0 (Rajpurkar et al., 2018), following much recent work in IE that takes a QA-based approach (Du and Cardie, 2020; Liu et al., 2020; Holzenberger et al., 2022, *i.a.*).

We map each possible role of each report trigger's FrameNet frame, together with that role's gold argument(s), to a single QA pair. Separate QA datasets are created for the source and report annotations. For the report dataset, the context passage for each QA pair is the report text, the "question"[10] is the concatenated names of the event and role, and the answer is the gold report argument(s) for that event and role. For the source dataset, the context passage is the source text; the question is the same as in the report model, but with the full report text (with marked event trigger) concatenated at the end; and the answer is again the gold source argument(s) for the given event and role. For the QA recast setting, if a role had multiple gold arguments, we only create a single instance for that role choosing the first appearing argument in the text. Both datasets' examples are in Appendix subsection C.2 (Figure 13).

Few-Shot LLMs As with SV, we present few-shot evaluations on CDAE using Llama and ChatGPT in the few-shot setting.[11] The prompt (again, the same for both models) describes the task and includes two examples from the FAMuS training split, each consisting of a document (report or source) and its CDAE annotations.

[9]The SV model we use for oversampling (see §4) is the same, except that we fine-tune it on Barham et al.'s SV data.

[10]The "questions" in QA-recasted IE datasets are often not syntactically interrogative (Du and Cardie, 2020); we follow this looser notion of a question here.

[11]We use `llama-2-13b-chat` (not `llama-2-13b`, as in SV). The ChatGPT version is unchanged.

Report Baseline For the source document, we present a baseline score by predicting gold arguments from the *report* document. This baseline, focusing solely on report-derived arguments, offers a relative measure—not a direct comparison—of the additional context sources provide compared to reports. A low score from this baseline would indicate sources furnish significant extra information.

Additionally, we mention ensembled model variations with the report baseline (+rb) in Tables 3 and 5. These variants supplement a model's predictions with report arguments for any role r lacking arguments, without altering other roles $r' \neq r$.

Evaluation We assess CDAE using the CEAF-RME metric from Chen et al. (2023b), which adapts argument P/R/F1 for models predicting argument *mentions* against references with complete coreference data.[12] We present two metric versions. The CEAF-RMEϕ_3 gives full credit for an exact mention match p_r with any mention g_r in reference entity C_{g_r} for role r.

We also use a modified version to reflect span boundary variability,[13] employing normalized edit distance ($\hat{A}$) for leniency:

$$\hat{A}(p_r, g_r) = 1 - \frac{\mathrm{E}(p_r, g_r)}{(\mathrm{S}-1)\min(L_{p_r}, L_{g_r}) + \max(L_{p_r}, L_{g_r})} \quad (1)$$

Here, E is Levenshtein distance with substitution cost S=2, and L_{p_r} and L_{g_r} are the token counts in p_r and g_r. We define a as the highest $\hat{A}$ across all $g_r \in C_{g_r}$:

$$a = \max_{g_r \in C_{g_r}} \hat{A}(g_r, p_r) \quad (2)$$

This second metric is CEAF-RME$_a$. We report both metrics using single gold-annotated mentions (Table 3) and full predicted coreference clusters from F-COREF (Table 5).

6 Results

6.1 Source Validation

Performance metrics for SV models and a majority baseline are outlined in Table 2, considering our balanced SV dataset.[14] The lemma baseline leads in precision and accuracy, indicating the trigger's lemma in a source document is a reliable validity signal and an effective overall heuristic.

Model	Accuracy	P	R	F1
Majority	50.00	100.00	50.00	66.66
Lemma	**75.89**	89.70	58.50	70.81
Longformer	71.94	66.67	**87.75**	**75.77**
ChatGPT	67.98	84.21	44.27	58.03
Llama-2-13b	58.50	65.93	35.18	45.88

Table 2: FAMuS Source Validation (SV) results.

The Longformer model, however, records the best recall (87.75%) and F_1 score (75.77%). The near 30-point recall advantage over the lemma baseline suggests it better identifies valid sources with paraphrased event descriptions. Users might favor the Longformer's recall over the lemma baseline's precision since CDAE can subsequently filter out falsely validated sources by contrasting source and report arguments.

Conversely, ChatGPT lags behind in accuracy and F_1, even falling short of the majority baseline in F_1 due to its lower recall. Its precision (84.21%) hints at a possible overemphasis on simple lexical signals. Llama 2 displays a comparable trend but with reduced metrics compared to ChatGPT.

6.2 Cross-Document Argument Extraction

Full CDAE results on the FAMuS test set are shown in Table 3.[15] A key (if unsurprising) theme that emerges is the value of high-quality candidate spans. The IterX$_{\text{gold}}$ results ablate span extraction and reflect argument labeling performance on gold spans for the target document. Unsurprisingly, these are the best absolute numbers, with CEAF-RME F_1 scores in the high 60s and low 70s. Setting aside the report baseline (rb) ensembles, Longformer-QA shows the best performance among models that do not have access to gold arguments, but even these results consistently trail F_1 scores of IterX$_{\text{gold}}$ by huge margins.

A second, related theme is the difficulty of CDAE on source documents relative to report documents. All models without access to gold spans (both few-shot and fine-tuned) see a significant drop in performance when moving from report extraction to source extraction: even the smallest such drop (CEAF-RME$_{\phi_3}$ for Llama) is still almost 7 F_1. This is likely a result of models having to consider

[12]Predicted mentions are treated as singletons that can align with reference entities, as detailed by Chen et al. (2023a) and Chen et al. (2023b).

[13]Differences in annotator practices regarding determiners and relative clauses affect span marking.

[14]The dataset has an equal class distribution.

[15]As noted in §5, results in Table 3 use only the human-annotated argument mentions in the reference. Results with full reference argument coreference clusters (generated by F-COREF) are in Table 5.

	Model	Report						Source					
		CEAF-RME$_{\phi 3}$			CEAF-RME$_a$			CEAF-RME$_{\phi 3}$			CEAF-RME$_a$		
		P	**R**	**F$_1$**	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**
−rb	IterX$_{\text{gold}}$	73.11	72.00	72.55	73.56	72.44	73.00	70.46	69.16	69.80	70.58	69.28	69.92
	IterX$_{\text{gold+pred}}$	40.57	29.38	34.08	42.24	30.59	35.48	25.07	10.82	15.11	29.85	12.88	18.00
	IterX$_{\text{pred}}$	37.63	24.14	29.41	42.16	27.04	32.94	20.83	8.63	12.21	27.63	11.45	16.19
	Longformer-QA	**43.56**	**40.14**	**41.78**	**56.01**	**51.61**	**53.72**	**25.53**	**22.21**	**23.75**	**38.85**	**33.80**	**36.15**
	ChatGPT	33.67	32.00	32.81	51.28	48.73	49.97	14.00	12.77	13.36	33.31	30.39	31.78
	Llama-2-13b-chat	12.97	22.76	16.52	23.65	41.49	30.13	11.14	8.52	9.65	20.11	15.36	17.42
+rb	Report Baseline (rb)	-	-	-	-	-	-	23.59	19.68	21.46	**47.80**	39.88	**43.48**
	IterX$_{\text{gold}}$	-	-	-	-	-	-	60.38	75.95	67.28	64.12	80.65	71.45
	IterX$_{\text{gold+pred}}$	-	-	-	-	-	-	24.43	19.56	21.73	38.47	30.82	34.22
	IterX$_{\text{pred}}$	-	-	-	-	-	-	22.24	17.38	19.51	37.42	29.24	32.83
	Longformer-QA	-	-	-	-	-	-	**24.12**	**25.89**	**24.97**	38.41	**41.24**	39.77
	ChatGPT	-	-	-	-	-	-	15.93	17.95	16.88	34.99	39.42	37.07
	Llama-2-13b-chat	-	-	-	-	-	-	11.11	8.52	9.64	20.24	15.51	17.56

Table 3: CEAF-RME scores for CDAE on FAMuS test set. The Report Baseline (rb) predicts the gold *report* arguments as the arguments for the source. IterX and Longformer-QA are fine-tuned on FAMuS. ChatGPT and Llama results are evaluated in the few-shot setting. "+/−rb" indicates whether the model is ensembled with the report baseline (see §5). **Bolded** results are best across models within the same +/−rb setting that do not have access to gold spans for the target document.

a much larger set of candidate arguments in the source to identify a set of *correct* ones that is generally comparable in size to the set of gold report arguments (see Table 1).

We also note that few-shot results with ChatGPT are notably close to those of fine-tuned models, surpassing IterX$_{\text{pred}}$ and IterX$_{\text{gold+pred}}$ on CEAF-RME$_a$ for both report and source tasks, and only slightly trailing Longformer-QA.[16]

While the report baseline (predicting gold arguments from the report) isn't directly comparable to models in the −rb group, it outperforms all non-gold models. Ensembling models with the report baseline usually boosts recall (and sometimes precision), but only the ensembled Longformer-QA beats the report baseline on CEAF-RME$_{\phi 3}$, yet it still lags on CEAF-RME$_a$. These outcomes hint at the models' struggle to extract new information beyond what is present in the report.

Finally, we note that the generally large absolute differences between CEAF-RME$_{\phi 3}$ and CEAF-RME$_a$ results for the same model and settings suggest that many predicted arguments are at least partially correct, but do not receive credit under exact match. These results point to the additional information about model performance that incorporating partial span matching into existing metrics can provide for argument extraction. Caution is warranted here though: weakening the requirement for exact matches increases the possibility that models get credit for mentions of incorrect referents—e.g. getting credit for responding *New York* when the correct mention is *New York Times*. Future work on incorporating partial matching into these metrics might investigate using coreference information to penalize models in these cases.

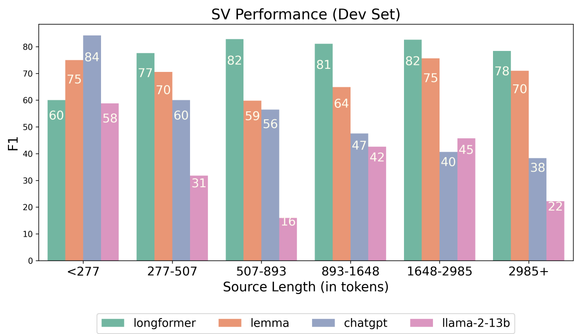

Figure 3: Source Validation F$_1$ on the FAMuS dev set, broken down by source document length percentile (0-10%, 10-25%, 25-50%, 50-75%, 75-90%, 90-100%).

6.3 Model Performance & Document Length

Next, we consider how model performance changes on both tasks as a function of the length of the source document. Figure 3 shows dev set source validation performance of models reported in Table 2, broken down by source length percentile. Several observations stand out. For one, ChatGPT performs exceptionally well on the shortest documents, achieving 84 F$_1$ and actually outperforming both the lemma baseline (75 F$_1$) and Longformer

[16]ChatGPT has larger gaps under CEAF-RME$_{\phi 3}$ due to challenges in exactly matching annotated mentions.

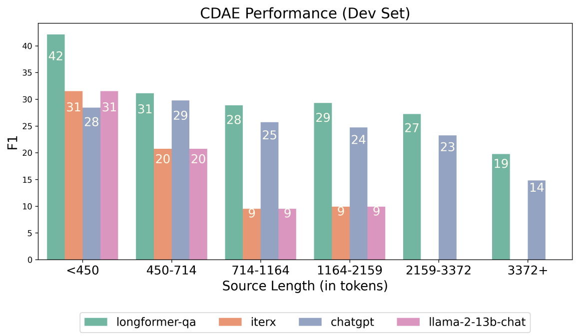

Figure 4: CEAF-RME$_a$ F$_1$ on Cross-Document Argument Extraction on source documents, broken down by document length. Percentile bins are the same as in Figure 3. IterX=IterX$_{\text{pred}}$ (see Table 3).

(60 F$_1$) by wide margins. Across the remaining bins, however, ChatGPT's performance decreases monotonically, faring worse than either of these models, suggesting its strong few-shot capabilities on this task (see above) may be limited to shorter texts. By contrast, Longformer exhibits remarkable consistency across source documents of different lengths: while its performance trails ChatGPT and the lemma baseline on the shortest documents, it outperforms them on all bins of greater length, sustaining F$_1$ scores between 77 and 82. Llama 2 exhibits the most *in*consistent performance, showing wide variation across bins.

CDAE results in Figure 4 contrast with SV findings, showing a consistent trend of performance decline from shorter to longer documents across all models. Notably, IterX and Llama 2 exhibit a pronounced drop, with CEAF-RME$_a$ F$_1$ scores plummeting below 10 for documents at or beyond the 25$^{\text{th}}$ percentile and reaching zero for those in the top quartile. ChatGPT and Longformer-QA perform slightly better, yet their F$_1$ scores remain below 20 for the longest 10% of documents. This highlights the significant need for argument extraction models that are more robust on long texts.

7 Conclusion

We have presented FAMuS, a new dataset comprising *reports* (Wikipedia passages) that describe an event, along with *source* documents for those events—featuring high-quality, full-document FrameNet frame and role annotations on both. We have also introduced two event understanding tasks enabled by FAMuS: *source validation*—determining whether a candidate document is a valid source for a given report event—and *cross-document argument extraction*—extracting the arguments of an identified report event in both the report and its source. We have provided baselines for both tasks, along with detailed analysis of their performance, and release both these models and our data to facilitate future research.

Limitations

One limitation of FAMuS is that its annotations are *non-exhaustive*: only the arguments of the (single) target event are annotated in the report and source. This makes it unsuited to training models for full (document-level) event extraction, in which systems typically may have to extract multiple events. Remedying this shortcoming is one of our primary goals for follow-up work.

Additionally, while FAMuS provides annotations for argument coreference, these are model-predicted, and thus will contain some noise. (Granted, this is irrelevant for evaluation against only the gold annotated spans, as in Table 3.)

Finally, because the valid source documents in FAMuS are cited by their corresponding reports, this may result in artificially high agreement between the arguments in the report and those in the source. Different internet sources routinely give somewhat differing, and even conflicting, accounts of the same event, and insofar as Wikipedia articles overwhelmingly cite documents *in support* of the claims they make, FAMuS likely overestimates the level of inter-document consensus present on the internet more broadly.

Ethics Statement

We do not believe this work raises significant ethical issues.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback. This work was supported by DARPA AIDA, IARPA BETTER, and NSF-BCS (2040831). The views and conclusions contained in this work are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, or endorsements of DARPA, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

1992. *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Amit Bagga and Breck Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Coreference and Its Applications*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Samuel Barham, Orion Weller, Michelle Yuan, Kenton Murray, Mahsa Yarmohammadi, Zhengping Jiang, Siddharth Vashishtha, Alexander Martin, Anqi Liu, Aaron Steven White, et al. 2023. Megawika: Millions of reports and their sources across 50 diverse languages. *arXiv preprint arXiv:2307.07049*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16, Trento, Italy. Association for Computational Linguistics.

Yunmo Chen, William Gantt, Tongfei Chen, Aaron Steven White, and Benjamin Van Durme. 2023a. A unified view of evaluation metrics for structured prediction. *arXiv preprint arXiv:2310.13793*.

Yunmo Chen, William Gantt, Weiwei Gu, Tongfei Chen, Aaron White, and Benjamin Van Durme. 2023b. Iterative document-level information extraction via imitation learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1858–1874, Dubrovnik, Croatia. Association for Computational Linguistics.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552, Reykjavik, Iceland. European Language Resources Association (ELRA).

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Xinya Du, Alexander Rush, and Claire Cardie. 2021a. GRIT: Generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 634–644, Online. Association for Computational Linguistics.

Xinya Du, Alexander Rush, and Claire Cardie. 2021b. Template filling with generative transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 909–914, Online. Association for Computational Linguistics.

Anca Dumitrache, Lora Aroyo, and Chris Welty. 2018. Capturing ambiguity in crowdsourcing frame disambiguation. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 6, pages 12–20.

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.

Alon Eirew, Avi Caciularu, and Ido Dagan. 2022. Cross-document event coreference search: Task, dataset and modeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 900–913, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alon Eirew, Arie Cattan, and Ido Dagan. 2021. WEC: Deriving a large-scale cross-document event coreference dataset from Wikipedia. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2498–2510, Online. Association for Computational Linguistics.

William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California. Association for Computational Linguistics.

Marco Fossati, Claudio Giuliano, and Sara Tonelli. 2013. Outsourcing FrameNet to the crowd. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 742–747, Sofia, Bulgaria. Association for Computational Linguistics.

William Gantt. 2021. Argument linking: A survey and forecast. *arXiv preprint arXiv:2107.08523*.

William Gantt, Reno Kriz, Yunmo Chen, Siddharth Vashishtha, and Aaron Steven White. 2022. On event individuation for document-level information extraction. *arXiv preprint arXiv:2212.09702*.

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

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Ralph Grishman. 2019. Twenty-five years of information extraction. *Natural Language Engineering*, 25(6):677–692.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Nils Holzenberger, Yunmo Chen, and Benjamin Van Durme. 2022. Asking the right questions in low resource template extraction. *arXiv preprint arXiv:2205.12643*.

Jisup Hong and Collin F. Baker. 2011. How good is the crowd at "real" WSD? In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 30–37, Portland, Oregon, USA. Association for Computational Linguistics.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA. Association for Computational Linguistics.

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

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

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.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Anne-Lyse Minard, Manuela Speranza, Ruben Urizar, Begoña Altuna, Marieke van Erp, Anneleen Schoen, and Chantal van Son. 2016. MEANTIME, the NewsReader multilingual event and time corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4417–4422, Portorož, Slovenia. European Language Resources Association (ELRA).

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28.

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

Joel Nothman, Matthew Honnibal, Ben Hachey, and James R. Curran. 2012. Event linking: Grounding event reference in a news archive. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–232, Jeju Island, Korea. Association for Computational Linguistics.

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

Timothy J O'Gorman. 2019. *Bringing together computational and linguistic models of implicit role interpretation*. University of Colorado at Boulder.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022. F-coref: Fast, accurate and easy to use coreference resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56, Taipei, Taiwan. Association for Computational Linguistics.

Jiefu Ou, Adithya Pratapa, Rishubh Gupta, and Teruko Mitamura. 2023. Hierarchical event grounding. *arXiv preprint arXiv:2302.04197*.

Dean Pomerleau and Delip Rao. 2017. Fake news challenge.

Adithya Pratapa, Rishubh Gupta, and Teruko Mitamura. 2022. Multilingual event linking to Wikidata. In *Proceedings of the Workshop on Multilingual Information Access (MIA)*, pages 37–58, Seattle, USA. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Michael Roth and Anette Frank. 2012a. Aligning predicate argument structures in monolingual comparable texts: A new corpus for a new task. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 218–227, Montréal, Canada. Association for Computational Linguistics.

Michael Roth and Anette Frank. 2012b. Aligning predicates across monolingual comparable texts using graph-based clustering. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 171–182, Jeju Island, Korea. Association for Computational Linguistics.

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

Beth M. Sundheim. 1992. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

MeiHan Tong, Bin Xu, Shuai Wang, Meihuan Han, Yixin Cao, Jiangqi Zhu, Siyu Chen, Lei Hou, and Juanzi Li. 2022. DocEE: A large-scale and fine-grained benchmark for document-level event extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3970–3982, Seattle, United States. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Piek Vossen, Antske Fokkens, Isa Maks, and Chantal van Son. 2018a. Towards an open dutch framenet lexicon and corpus. In *Proceedings of the LREC 2018 Workshop International FrameNet Workshop*, pages 75–80.

Piek Vossen, Filip Ilievski, Marten Postma, Antske Fokkens, Gosse Minnema, and Levi Remijnse. 2020. Large-scale cross-lingual language resources for referencing and framing. In *Proceedings of the Twelfth*

Language Resources and Evaluation Conference, pages 3162–3171, Marseille, France. European Language Resources Association.

Piek Vossen, Filip Ilievski, Marten Postma, and Roxane Segers. 2018b. Don't annotate, but validate: a data-to-text method for capturing event data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258, Toronto, Canada. Association for Computational Linguistics.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

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*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Travis Wolfe, Benjamin Van Durme, Mark Dredze, Nicholas Andrews, Charley Beller, Chris Callison-Burch, Jay DeYoung, Justin Snyder, Jonathan Weese, Tan Xu, and Xuchen Yao. 2013. PARMA: A predicate argument aligner. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 63–68, Sofia, Bulgaria. Association for Computational Linguistics.

Patrick Xia, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme. 2021. LOME: Large ontology multilingual extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 149–159, Online. Association for Computational Linguistics.

Xiaodong Yu, Wenpeng Yin, Nitish Gupta, and Dan Roth. 2023. Event linking: Grounding event mentions to Wikipedia. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2679–2688, Dubrovnik, Croatia. Association for Computational Linguistics.

A Annotation Details

A.1 Source Validation

Barham et al. (2023) devise their own source validation task for (report, source) pairs in MegaWika, in which annotators on Amazon Mechanical Turk are presented with a highlighted event trigger in the report text and are asked two questions:

1. What is the most likely FrameNet frame for the highlighted text in the report?[17]
2. Does the source describe the same event as is denoted by the highlighted report trigger?

In this work, we refine Barham et al.'s methodology for our own SV annotation. To ensure high-quality annotations, we restrict *positive* SV examples to the set of (report, source) pairs where (i) a majority (2/3) agrees on the report event's correct frame, and (ii) *either* all three annotators *unanimously* agree that the source is a valid one for the report event *or* 2/3 agree and an expert (one of the authors) agrees with the majority.[18]

Positive Examples

To select (report, source) pairs for annotation, we rely on the oversampling technique from Barham et al. (2023), which leverages the LOME FrameNet parser of Xia et al. (2021) and a simple Longformer-based model for the SV task (see §5). Broadly, for each frame, we want to estimate how many *total* examples we need to annotate in order obtain five *positive* ones. This count (X) is modeled as a negative binomial random variable $X \sim \text{NB}(r, p)$ with $r = 5$ denoting the desired number of positive examples and p denoting the probability of an example being positive. Given our two criteria for positive examples, p can be expressed as $p = P_i \cdot v$, where P_i is the precision of the parser on frames of type f_i and where v is the test set accuracy of our SV model.[19] For frames for which the FrameNet test set has poor support (< 10 examples), we use the *average* precision across all frames, P_{avg}, in lieu of P_i, for a more robust estimate. Thus, the *expected* number of examples needed to obtain five positive ones, $D_i = \mathbb{E}[X]$, is:

$$D_i = \begin{cases} \lceil \frac{5}{P_i * v} \rceil, & \text{if } count(f_i) \geq 10 \\ \lceil \frac{5}{P_{avg} * v} \rceil, & \text{otherwise} \end{cases} \quad (3)$$

[17] Annotators are shown the top five candidate frames from a FrameNet parser along with a "none" option.

[18] A subset of the authors inspected all 2/3 majority cases.

[19] P_i and v correspond to our models' ability to correctly answer questions (1) and (2) above, respectively.

While annotating D_i examples for frame f_i will yield five positive examples *in expectation* for f_i, multiple rounds of annotation are needed to actually obtain five positive examples for all frames. In total, we conducted seven rounds. In each round, for each frame f_i, we use stratified sampling to ensure diversity among the D_i (report, source) pairs selected for annotation. We first identify a candidate set c_i of 250 pairs from MegaWika for which the FrameNet parser has identified at least one instance of frame f_i in the report.[20] We then perform k-means clustering on all pairs, clustering on the SpanBERT (Joshi et al., 2020) `CLS` token embedding of the first five sentences of the source text for each pair, fixing $k = D_i$. We then sample one pair from each cluster, aiming to select pairs for which the report trigger's lemma differs from those in all other pairs chosen for f_i.

Figure 5 shows an example of the source validation annotation interface with the report text displayed. Figure 6 shows the same example, but with the *source* text displayed, highlighting that the document *is* a valid source for the report event.

A.2 CDAE

Figure 7 shows an example of an annotated role instance from our role annotation task interface.

B IAA and Annotation Correction

This appendix offers additional details on annotator agreement and annotation correction for CDAE.

B.1 Agreement

Here, we describe the agreement metric used for computing inter-annotator agreement (IAA) for CDAE annotation. We compute a F_1 score based on the maximum normalized edit distance (a) between annotated and reference argument *mentions* given in Eq. (2). If r is a role in the role set R_f for a frame f; p_r is a predicted mention; g_r is a reference mention; C_{g_r} is the reference entity containing mention g_r; and ϵ is the "null" span (indicating the absence of an argument), we compute this F_1 score based on the following counts of true positive (TP), false positive (FP), and false negative (FN) arguments:

$$\text{TP} = \sum_{\substack{C_{g_r} \neq \phi \,\cap\, p_r \neq \epsilon \\ r \in R_f}} a$$

[20] If fewer than 250 pairs are available for f_i, we include all $N_i < 250$ pairs.

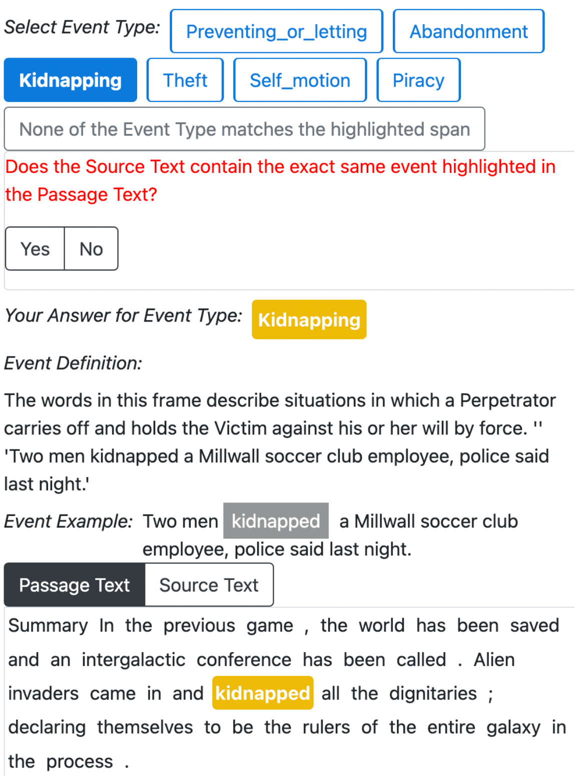

Figure 5: The source validation annotation interface, with the report ("passage") text displayed. Annotators are shown the report with a highlighted event trigger and are asked to select the correct frame for the trigger from among the top five predictions of a FrameNet parser (or none, if all candidates are wrong). When a candidate frame is selected, its definition and an example from FrameNet are displayed.

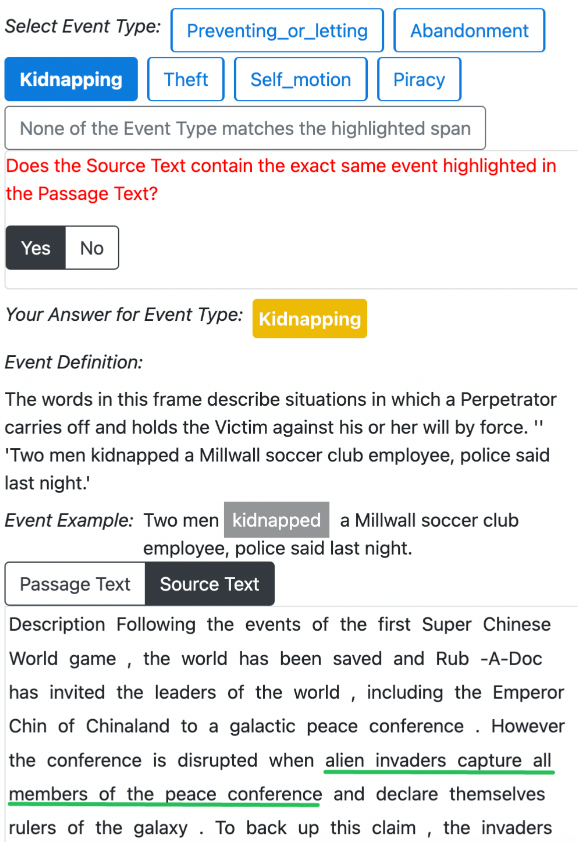

Figure 6: The same example as in Figure 5, but with (a portion of) the source text displayed. Here, the source document *does* describe the same report event (relevant text underlined in green) as shown in Figure 5, and so is a valid source. Role annotation (§4.2) is done only on examples with valid source texts.

$$\mathrm{FP} = \sum_{\substack{C_{gr} \neq \phi \,\cap\, p_r \neq \epsilon \\ r \in R_f}} \frac{1-a}{2} + \sum_{\substack{C_{gr} = \phi \,\cap\, p_r \neq \epsilon \\ r \in R_f}} 1$$

$$\mathrm{FN} = \sum_{\substack{C_{gr} \neq \phi \,\cap\, p_r \neq \epsilon \\ r \in R_f}} \frac{1-a}{2} + \sum_{\substack{C_{gr} \neq \phi \,\cap\, p_r = \epsilon \\ r \in R_f}} 1$$

B.2 Annotation Correction

As discussed in §4, we use the agreement metric above to evaluate the similarity between annotators' CDAE annotations on the source text and those produced by ChatGPT (`gpt-3.5-turbo-0301`) in order to identify potentially lower quality human annotations. At the end of each round of CDAE annotation, (report, source) pairs for which the source agreement score with ChatGPT falls in the bottom quartile are manually verified and corrected by the authors. The prompt template we use to obtain source document CDAE annotations with ChatGPT is shown in Figure 9. The prompt includes two examples in the chat history, where the first is the same across report documents, while the second uses the gold annotation from the report associated with the target source document. We set `max_tokens` to 128, `top_p` to 1.0, and `temperature` to 0, with no presence or frequency penalties. Figure 8 shows boxplots of the agreement F_1 scores between the report and source annotations before and after manual correction by the authors, aggregated over all rounds of annotation. Note that the majority of corrected annotations actually exhibit perfect agreement with their uncorrected counterparts, resulting in high mean scores of 0.90 and 0.85 for reports and sources, respectively, and offering compelling evidence for the quality of the annotations overall.

C Model Details

This appendix presents model implementation details, hyperparameters, and prompts. The training of all Longformer models was conducted on a single NVIDIA GeForce GTX 1080 Ti graphics

Figure 7: The role annotation interface for the same example as in Figure 5. Here, annotators identify arguments of the highlighted report ("passage") event in the full texts of *both* the report and source.

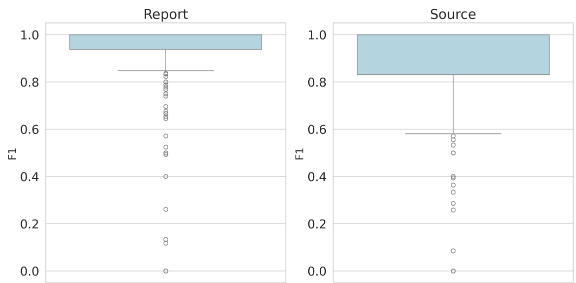

Figure 8: Boxplots for agreement F_1 between bottom quartile report and source CDAE annotations before and after correction by the authors.

processing unit (GPU), equipped with 12 GB of RAM. The training of all Iter-X models was performed using an NVIDIA Quadro RTX 6000/8000 GPU, with 26 GB of RAM. For all experiments conducted within this study, we employed Stanza version 1.2.3 and the Hugging Face 'transformers' library version 4.12.5. Additionally, the Natural Language Toolkit (NLTK) version 3.6.3 was utilized for text-processing tasks.

C.1 Source Validation

Longformer We use the LongformerForSequenceClassification class from the HuggingFace Transformers library (Wolf et al., 2020) to fine-tune Longformer for source validation. We fine-tune for 15 epochs with a batch size of 2, and an initial warmup phase of 400 steps. We conduct limited hyperparameter search using Optuna (Akiba et al., 2019), targeting the learning rate and weight decay, and varying them logarithmically from 1e-6 to 1e-4 and from 1e-6 to 1e-1 respectively. This process is conducted over 5 trials, with the optimal setting selected based on the highest validation accuracy. We then fine-tune a final Longformer model for 30 epochs using the best hyperparameter configuration, using the checkpoint with highest dev accuracy across all 30 epochs for the final evaluation.

ChatGPT We use `gpt-3.5-turbo-0301` and do not perform any fine-tuning or hyperparameter search, evaluating only in the few-shot setting. Figure 10 presents a sketch of the prompt we use. We set `max_tokens` to 128, `top_p` to 1.0, and `temperature` to 0, with no presence or frequency penalties.

Llama 2 We use `llama-2-13b` for source validation and do not perform any fine-tuning or hyperparameter search (just as with ChatGPT). We use the default hyperparameters, except for `max_seq_len` (5,000), `max_gen_len` (128), `top_p` (0.9), and `temperature` (0.0). The prompt is the same as the one used for ChatGPT on SV (Figure 10).

C.2 Cross-Document Argument Extraction

Example Inputs Figure 13 shows model inputs to the Iter-X model and to the Longformer-QA model.

Longformer We use the LongformerForQuestionAnswering class from the HuggingFace transformers library to fine-tune the Longformer-QA model on the recasted CDAE datasets for both report and source. We fine-tune for a maximum of 10 epochs with a batch size of 1. As with Longformer for SV, we use the Optuna library for hyperparameter tuning to optimize the learning rate and weight decay, varying them logarithmically from 1e-6 to 1e-4 and 1e-6 to 1e-1 respectively. This process is conducted over 5 trials, with the optimal setting selected based on the lowest validation loss.

IterX We base our IterX hyperparameters on the best ones reported for the MUC-4 task in Table 6 of Chen et al. (2023b), though with two important differences. First, as noted in §5, the CDAE task requires extraction of only a single template per document. As such, we set the maximum number of templates to decode ("#Max Iterations") to 1. Second, Chen et al. train their model for MUC-4 on *predicted* spans only, whereas we use different

System Prompt

```
You are a system that generates high quality data for document-level role annotations based on
Framenet.
The following inputs are given to you:
1. Event Type: A Frame name from the FrameNet ontology (eg: Hiring, Arrest, etc.)
2. Event Definition: Definition of the event type along with an optional example.
3. Roles: All roles (or participants) of the event type (or frame) followed with an optional
example.
4. Document: A document from which the roles are to be extracted.
You should output the extracted spans from the document for each role in the order they are listed
in the roles section.
Note that you can leave a N/A if no span is found for that role.
```

First prompt example (Fixed template)

```
User
Event Type: Hiring
Event Definition: definition + example
Roles:
1.Employee: definition + example
2.Employer: definition + example
…
Document:
John Smith is a recent graduate of the University of Washington. He interned at Microsoft Research
in Seattle, Washington. His research includes machine learning, computer vision, and natural
language processing. After 6 rounds of interviewing, he was hired as a Research Scientist by
Microsoft to work on their new chatbot.

Assistant
1. Employee: John Smith
2. Employer: Microsoft
3. Task: to work on their new chatbot
4. Position: Research Scientist
5. Field: N/A
```

Second prompt example (Gold report annotation)

```
User
Event Type: <gold_frame>
Event Definition: definition + example
Roles:
<definitions + examples for the gold_frame's roles>

Document:
<report_text>

Assistant
<report_gold_annotation>
```

Target Example (Source)

```
User
Event Type: <gold_frame>
Event Definition: definition + example
Roles:
<definitions + examples for the gold_frame's roles>
…
Document:
<source_text>
```

Figure 9: Prompt template used to generate CDAE annotations on the source for annotation correction. Note the use of gold report annotation as the second prompt example.

Prompt Prefix

```
In this task, you are given a report document marked up an XML tag 'report'.
The report describes an event denoted with an XML tag 'event'. You are also given a source
document marked up an XML tag 'source'. Your task is to determine whether the 'source' document
contains the 'event' described in the 'report' or not. This is equivalent to determining whether
the source is a valid reference for the tagged event in the report.
Steps to follow to arrive at the answer:
    1. Summarize the 'event' described in the 'report' in one line.
    2. Check if the 'source' document describes the summarized 'event' or not.
    If the 'source' document describes the summarized 'event', then
    in one line explain how the 'source' document describes the 'event' and answer 'yes'.
    If the 'source' document does not describe the summarized 'event', then
    in one line explain how the 'source' document does not describe the 'event' and answer 'no'.

The answer 'Yes' or 'No' should be in a separate line at the end inside the <valid_source> tag.
Below are some examples.

<report> Jon <event> picked </event> up the gun. </report>
<source> Jon enjoyed hunting. One day, he grabbed his gun and went to the forest. </source>
<answer>
   The report focuses on the event of Jon picking up the gun.
   The source describes Jon grabbing his gun which is the same event tagged in the report.
  <valid_source> Yes <valid_source>
</answer>

<report> Jon  <event> picked </event> up Janice. </report>
<source> Jon enjoyed driving a lot. One day, he picked up Daniel from a store. </source>
<answer>
   The report focuses on the event of Jon picking up Janice.
   The source describes Jon picking up Daniel which is not the same event tagged in the report.
    <valid_source> No <valid_source>
</answer>

<report> <event> Riots </event>  erupted in various parts of the city after the violent speech.
</report>
<source> Various violent acts were seen in the city after the minister's controversial hate
speech. </source>
<answer>
   The report focuses on the event of riots erupting in various parts of the city.
   The source describes various violent acts in the city which is the same event tagged in the
report.
   <valid_source> Yes <valid_source>
</answer>

<report>  Osama Bin Laden was <event> killed </event> in Abbottabad, Pakistan on May 2, 2011 </
report>
<source> Osama bin Mohammed bin Awad bin Laden was a Saudi Arabian-born militant and founder of
the pan-Islamic militant organization Al-Qaeda. </source>
<answer>
   The report focuses on the killing of Osama Bin Laden.
   The source does not mention anything about the killing of Osama Bin Laden.
   <valid_source> No <valid_source>
</answer>
```

Prompt Suffix

```
<report> {report} </report>
<source> {source} </source>
<answer>
```

Figure 10: Prompt template used for ChatGPT and Llama 2 on SV.

System Prompt

```
You are a system that generates high quality document role annotations based on Framenet ontology.
The following inputs are given to you:
1. Event Type <event_type>: A Frame name from the FrameNet ontology (eg: Hiring, Arrest, etc.)
2. Event Definition <event_definition>: Definition of the event type along with an optional
   example.
3. Roles <event_roles>: All roles (or participants) of the event type (or frame) followed with an
   optional example.
4. Report Document <report_document>: A report document with a tagged event '<event>' of the given
   event type.

Your job is to extract all the roles of the tagged Report <event> from the <report_document>. The
ouput should be in a JSON string
where each key represents the role name as provided in the <event_roles> and its corresponding
value should be a
list of contiguous text spans from the <report_document> that are valid for that role.
Note that if no text span is found for a role, the value should be an empty list.
Your text spans should strictly come from the <report_document>. DO NOT use spans from Event
Definition or Roles sections.
```

First prompt example (Fixed)

User

```
<event_type> Hiring </event_type>
<event_definition> {event_definition_from_framenet} </event_definition>
<event_roles>
1.Employee: definition + example
2.Employer: definition + example
…
</event_roles>
<report_document> He was <event> hired </event> as a Research Scientist by Microsoft. </
report_document>
```

Assistant

```
{
  "Employee": ["He"],
  "Employer": ["Microsoft"],
  "Task": [],
  "Position": ["as a Research Scientist"],
  "Field": []
}
```

Second Prompt Example (Fixed)

User

```
<event_type> Clemency </event_type>
<event_definition> {event_definition_from_framenet} </event_definition>
<event_roles>
{role definition + examples from FrameNet}
…
</event_roles>
<report_document> In 2023, the governor granted <event> clemency </event> to the prisoner who had
been wrongly convicted. </report_document>
```

Assistant

```
{
  "Offender": ["the prisoner"],
  "Crime": [],
  "Executive_authority": ["The governor"],
  "Time": ["2023"],
  "Place": []
}
```

Target Example

User

```
<event_type> {gold_frame} </event_type>
<event_definition> {event_definition_from_framenet} </event_definition>
<event_roles>
{role definition + examples from FrameNet}
…
</event_roles>
<report_document> {report} </report_document>
```

Figure 11: Prompt template used for ChatGPT and Llama 2 on CDAE for report documents.

System Prompt

```
You are a system that generates high quality cross-document role annotations based on Framenet
ontology.
The following inputs are given to you:
1. Event Type <event_type>: A Frame name from the FrameNet ontology (eg: Hiring, Arrest, etc.)
2. Event Definition <event_definition>: Definition of the event type along with an optional
   example.
3. Roles <event_roles>: All roles (or participants) of the event type (or frame) followed with an
   optional example.
4. Report Document <report_document>: A report document with a tagged event '<event>' of the
   given event type.
5. Source Document <source_document>: A document from which the roles are to be extracted.

Your job is to extract all the roles of the tagged Report <event> from the <source_document>.
The ouput should be in a JSON string where each key represents the role name as provided in
the <event_roles> and its corresponding value should be a contiguous text span from the
<source_document>. Note that if no text span is found for a role, the value should be an empty
string. Your text spans should strictly come from the <source_document>. DO NOT use spans from
Event Definition, Roles, or Report Document sections.
```

First Prompt Example (Fixed)

User

```
<event_type> Hiring </event_type>
<event_definition> {event_definition_from_framenet} </event_definition>
<event_roles>
1.Employee: definition + example
2.Employer: definition + example
…
</event_roles>
<report_document> He was <event> hired </event> as a Research Scientist by Microsoft. </
report_document>
<source_document> John Smith is a recent graduate of the University of Washington. He interned at
Microsoft Research in Seattle, Washington. After 6 rounds of interviewing, he was hired as a
Research Scientist by Microsoft to work on their new chatbot. </source_document>
```

Assistant

```
{
  "Employee": ["John Smith"],
  "Employer": ["Microsoft"],
  "Task": ["to work on their new chatbot"],
  "Position": ["as a Research Scientist"],
  "Field": []
}
```

Second Prompt Example (Fixed)

User

```
<event_type> Clemency </event_type>
<event_definition> {event_definition_from_framenet} </event_definition>
<event_roles>
{role definition + examples from FrameNet}
…
</event_roles>
<report_document> In 2023, the governor granted <event> clemency </event> to the prisoner who had
been wrongly convicted. </report_document>
<source_document> In 2023, Governor Rick granted clemency to John Doe, who had been wrongly
convicted of murder in 1993. Doe was exonerated in 2012 after DNA testing proved that he was
innocent. He spent 19 years in prison before being released. </source_document>
```

Assistant

```
{
  "Offender": ["John Doe"],
  "Crime": ["murder"],
  "Executive_authority": ["Governor Rick"],
  "Time": ["2023"],
  "Place": []
}
```

Target Example

User

```
<event_type> {gold_frame} </event_type>
<event_definition> {event_definition_from_framenet} </event_definition>
<event_roles>
{role definition + examples from FrameNet}
…
</event_roles>
<report_document> {report} </report_document>
<source_document> {source }</source_document>
```


Figure 12: Prompt template used for ChatGPT and Llama 2 on CDAE for source documents.

IterX	Report Model	Source Model
Candidate Spans	1. <event> adjourned </event> 2. 16 July 2007 3. the inquiry 4. until 4 September …	1. <event> adjourned </event> 2. THE public inquiry into the controversial Mottram – Tintwistle bypass 3. public 4. the controversial Mottram – Tintwistle bypass …
Context	<event> adjourned </event> On 16 July 2007 the inquiry was adjourned until …	On 16 July 2007 the inquiry was <event> adjourned </event> until 4 September … THE public inquiry into the controversial Mottram – Tintwistle bypass was …
Longformer		
Question	**Event:** Activity_pause, **Role:** Activity, **Trigger:** adjourned	**Event:** Activity_pause, **Role:** Activity, **Report:** On 16 July 2007 the inquiry was **<event> adjourned </event>** until 4 September with a …
Context	On 16 July 2007 the inquiry was adjourned until 4 September with a …	THE public inquiry into the controversial Mottram – Tintwistle bypass was dramatically halted when the Highways Agency admitted it had got its figures wrong …
Answer	the inquiry	THE public inquiry into the controversial Mottram – Tintwistle bypass

Figure 13: **Top**: IterX inputs for the example in Figure 2, including the set of candidate spans (first row) and the document text (second row) for the report (left) and source (right) models. **Bottom**: Longformer QA report (left) and source (right) model inputs for the `Activity` role for the example in Figure 2. Note that the question for the source model has the report text prepended, with the event trigger highlighted. This is done to condition extraction specifically on that trigger. See §5 for details.

sets of spans for training depending on the setting (**gold**, **predicted**, or **gold and predicted**). All models are trained for a maximum of 150 epochs with a patience of 30, using CEAF-RME$_{\phi_3}$ on the dev set as the validation metric.

ChatGPT As with SV, we do not perform any fine-tuning or hyperparameter search on ChatGPT for CDAE. The model version (`gpt-3.5-turbo-0301`) and hyperparameters used here are also identical to those used for ChatGPT on SV. We use separate prompts for extraction on report and source documents. The prompt used for source documents is sketched in Figure 12. It consists of a system prompt, two example extractions (included in the chat history), followed by the target example on which extraction is to be performed. Note that this is different from the prompt used to generate CDAE annotations for purposes of annotation correction (Figure 9).

Llama 2 We use `llama-2-13b-chat` in lieu of `llama-2-13b` (used in SV), though all hyperparameters are the same as those used for Llama on SV. The prompt is the same as that used for ChatGPT for CDAE.

D Frame Selection

We follow the frame selection methodology of Barham et al. (2023) for selecting situation-denoting frames. Drawing inspiration from Moens and Steedman (1988), we focus on the top-level EVENT, STATE, and PROCESS FrameNet frames. We initially take these three frames and all those related to them via the INHERITANCE, SUBFRAME, or PRECEDES relations, on the assumption that the set of situation-denoting frames is closed under these relations, yielding 387 frames.[21]

However, some of these frames also inherit from other top-level frames that are *not* situation-denoting (i.e. RELATION, ENTITY, and LOCALE). We remove all such frames from the set above, which leaves 369 frames remaining.

Finally, because we are reliant on an existing FrameNet parser for data collection (Xia et al., 2021), we must further subset to those frames for which there is FrameNet training data and that therefore exist in the model's vocabulary. This yields the final set of 328 frames reported in §4.

E Additional Statistics

E.1 Similarity Between Report and Source

As discussed in §4, we use SimLM to compute the similarity between (report, source) pairs when automatically generating negative examples for the SV task. Figure 14 presents the distributions of the SimLM scores for all positive SV examples (top left), all negative SV examples (top right), human-annotated (gold) negative examples (bottom left), and automatically curated (silver) negative examples (bottom right). As may be expected, the modal similarity in the negative example plots is less than

[21] See the FrameNet Lattice List: `https://framenet.icsi.berkeley.edu/FrameLatticeList`

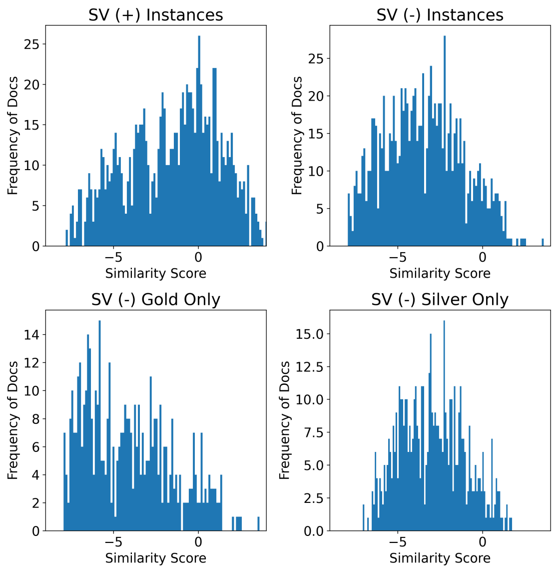

Figure 14: Histogram of SimLM similarity scores between the Report and the Source Text across SV train and dev examples.

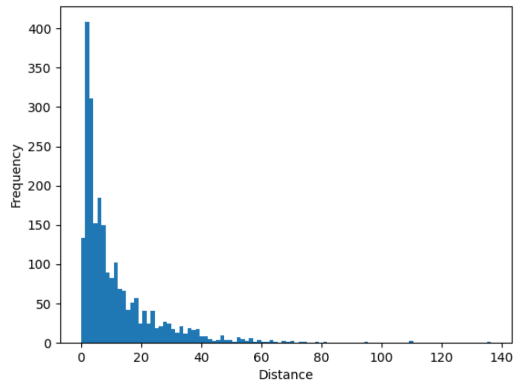

Figure 15: Histogram of word distances between triggers and their arguments in report documents from train.

		Train	Dev
Report	Mean	20.5	25.6
	Median	16.5	22.0
	Std Dev	18.2	18.9
Source	Mean	193.7	310.4
	Median	67.3	122.0
	Std Dev	353.6	529.0

Table 4: Statistics for word distances between the first and last arguments in report and source documents.

that of the positive examples. This is true even for the silver negative examples, for which we deliberately selected sources based on similarity to a target report, which offers further evidence that we are unlikely to be accidentally including positive source documents in the automatically generated negative examples.

E.2 Argument Distances

Here, we report distributions and statistics for word distances between (1) event triggers and their arguments in training split report documents (Figure 15); and (2) the first and last arguments annotated in each report and source document (Table 4, Figure 16) in the training split. Recall that in contrast to a number of resources for event argument extraction (EAE; Ebner et al., 2020; Li et al., 2021), FAMuS permits arguments to be annotated *anywhere* in the report and source documents.

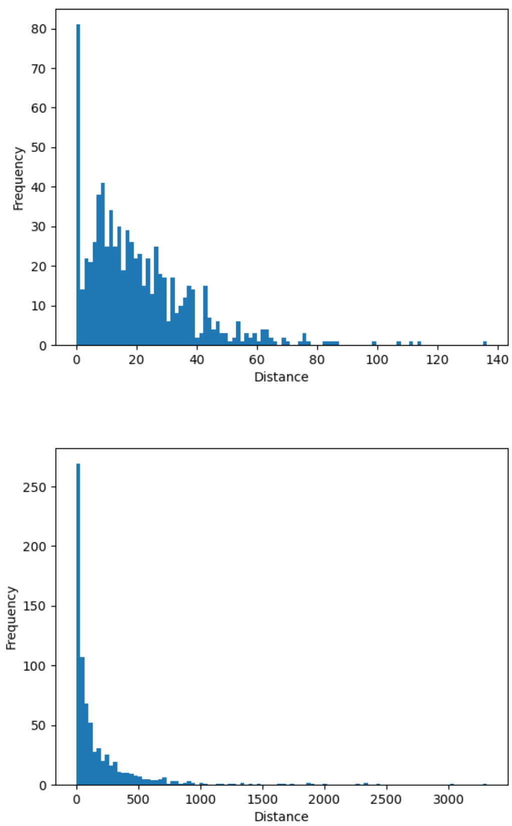

Figure 16: Histogram of word distances between the first and last arguments in *report* documents (top) and source documents (bottom) from the train split.

F Additional Results

Table 5 presents CEAF-RME scores on the same models as in Table 3, but using the full coreference cluster for each gold argument (as predicted by F-COREF) in the metric computation. The results are qualitatively similar (Longformer-QA remains dominant in the $-$rb setting and the Report Baseline still generally outperforms models in the $+$rb setting), though absolute F_1 scores are noticeably reduced. This reduction in F_1 scores when using coreference information may seem counterintuitive, as one might expect higher scores due to increased leniency in what counts as a correctly identified argument. However, the models are still predicting singleton entities while being evaluated with the ϕ_3 and soft-match ϕ_3 scoring functions, which only award full credit if all and only the mentions in the reference entity are predicted. As this is impossible for models predicting singleton entities, the coreference-based evaluation results in lower scores compared to the non-coreference evaluation.

		Report						Source					
		CEAF-RME$_{\phi_3}$			CEAF-RME$_a$			CEAF-RME$_{\phi_3}$			CEAF-RME$_a$		
	Model	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**	**P**	**R**	**F$_1$**
$-$rb	IterX$_{\text{gold}}$	73.11	57.17	64.17	73.56	57.53	64.56	70.46	30.26	42.34	70.61	30.33	42.43
	IterX$_{\text{gold+pred}}$	40.95	23.55	29.90	42.24	24.29	30.84	27.47	5.19	8.73	31.63	5.97	10.05
	IterX$_{\text{pred}}$	38.06	19.39	25.69	42.33	21.56	28.57	23.61	4.28	7.25	29.80	5.40	9.15
	Longformer-QA	**44.31**	**32.42**	**37.44**	**56.92**	**41.65**	**48.10**	**29.10**	**11.08**	**16.05**	**41.86**	**15.93**	**23.08**
	ChatGPT	35.85	27.05	30.84	53.27	40.20	45.82	17.28	6.90	9.86	36.48	14.56	20.82
	Llama-2-13b-chat	13.68	19.06	15.93	24.20	33.72	28.18	12.35	4.13	6.19	21.41	7.16	10.73
$+$rb	Report Baseline (rb)	-	-	-	-	-	-	**28.69**	10.47	15.34	**51.08**	18.65	**27.32**
	IterX$_{\text{gold}}$	-	-	-	-	-	-	61.21	33.69	43.46	64.91	35.72	46.09
	IterX$_{\text{gold+pred}}$	-	-	-	-	-	-	27.16	9.52	14.09	40.55	14.21	21.05
	IterX$_{\text{pred}}$	-	-	-	-	-	-	25.04	8.56	12.76	39.93	13.65	20.35
	Longformer-QA	-	-	-	-	-	-	26.90	**12.64**	**17.20**	41.30	**19.40**	26.40
	ChatGPT	-	-	-	-	-	-	19.10	9.42	12.61	38.24	18.85	25.26
	Llama-2-13b-chat	-	-	-	-	-	-	12.31	4.13	6.18	21.44	7.19	10.77

Table 5: Results for the same models as reported in Table 3, but using full (F-COREF-predicted) coreference clusters for the reference arguments when computing CEAF-RME scores.