BEMEAE: Moving Beyond Exact Span Match for Event Argument Extraction

Enfa Fane* **Md Nayem Uddin†** **Oghenevovwe Ikumariegbe*** **Daniyal Kashif***
Eduardo Blanco* **Steven R. Corman†**
*University of Arizona †Arizona State University
*{enfageorge, oaikumariegbe, dkashif, eduardoblanco}@arizona.edu
†{muddin11, steve.corman}@asu.edu

Abstract

Event Argument Extraction (EAE) is a key task in natural language processing, focusing on identifying and classifying event arguments in text. However, the widely adopted exact span match (ESM) evaluation metric has notable limitations due to its rigid span constraints, often misidentifying valid predictions as errors and underestimating system performance. In this paper, we evaluate nine state-of-the-art EAE models on the RAMS and GENEVA datasets, highlighting ESM's limitations. To address these issues, we introduce BEMEAE (Beyond Exact span Match for Event Argument Extraction), a novel evaluation metric that recognizes predictions that are semantically equivalent to or improve upon the reference. BEMEAE integrates deterministic components with a semantic matching component for more accurate assessment. Our experiments demonstrate that BEMEAE aligns more closely with human judgments. We show that BEMEAE not only leads to higher F1 scores compared to ESM but also results in significant changes in model rankings, underscoring ESM's inadequacy for comprehensive evaluation of EAE.

1 Introduction

Event Argument Extraction (EAE) is a complex task in natural language understanding that requires deep comprehension of text to accurately identify and classify event arguments (Huang et al., 2024). As a core component of event extraction—the process of transforming unstructured text into structured event representations—EAE plays a critical role in enabling various downstream applications such as narrative event prediction (Wang et al., 2021) and news summarization (Li et al., 2016). Given its broad applicability and significance, the precise evaluation and meaningful comparison of EAE systems is crucial for advancing research.

The most commonly used evaluation metric for EAE, Exact Span Match (ESM), assesses model predictions by requiring an exact match between the predicted argument spans (candidates) and human-annotated argument spans (references). While this approach offers a clear and objective measure of performance, it suffers from several notable limitations. Zhang et al. (2020b) and Uddin et al. (2024) analyzed candidate arguments flagged as errors and found that many were not genuine mistakes; humans would often consider these arguments valid for the assigned role, even if they did not exactly align with the reference span. For example, as illustrated in Figure 1, although *"convoy"* is a valid argument for the role *"target"* of event *"airstrike,"* ESM would incorrectly classify it as an error because it does not exactly match the reference argument, *"the vehicles."*

There are no craters, while [the vehicles]$^{\text{reference}}$ have their chassis intact and they have not been severely damaged, which would have been the case from an [airstrike]$^{\text{event\_trigger}}$, Konashenkov said. On Tuesday, the UN also retreated from its claims that the [convoy]$^{\text{candidate}}$ was hit by military planes.

Event type: conflict.attack.selfdirectedbattle
Role: target

Figure 1: An example from RAMS where *airstrike* is the event trigger and *the vehicles* is the reference argument for the role *target* . The standard metric, exact span match (ESM) is suboptimal. Seven models we work with correctly predict *convoy* as the *target* of the airstrike, yet ESM classifies it as an error. Furthermore, *vehicles*, *chassis*, *their chassis*, or *the convoy* would also receive no credit with ESM despite being correct, as none are an exact span match to *the vehicles*.

Motivated by these observations and the need for a more comprehensive evaluation framework, we introduce BEMEAE (Beyond Exact span Match for Event Argument Extraction).[1] BEMEAE addresses key shortcomings of ESM by combining deterministic components to handle textual variations with a

[1]BEMEAE is available at https://github.com/beingenfa/bemeae.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5734–5749

semantic matcher that evaluates the meaning of candidate arguments relative to reference arguments. Our experiments, conducted on nine models across two datasets, show that BEMEAE provides a more accurate assessment of system performance and leads to significant changes in system rankings.

The main contributions of this paper are:

1. **BEMEAE:** We introduce a novel evaluation metric that addresses the limitations of ESM by incorporating deterministic components and a semantic matcher.
2. **Comprehensive Evaluation:** Through extensive experiments on nine EAE models across RAMS and GENEVA datasets, we demonstrate that (a) BEMEAE aligns more closely with human judgments than ESM, and (b) F1 scores under BEMEAE reveal uneven improvements across models, providing new insights into model performance and leading to significant changes in system rankings.
3. **Automatic Semantic Matching:** We evaluate various semantic matching methods, including GPT-4. They obtain a fair agreement with human judgments, enabling scalable evaluations without the need for costly manual annotations.

A key advantage of our approach is its adaptability to any event argument extraction framework, without the need for additional costly human assessments or extensive retraining. This flexibility allows it to be seamlessly integrated into existing systems, providing a more accurate evaluation of model performance with minimal overhead.

2 Background

Event extraction (EE) is a key task in information extraction that focuses on converting unstructured text into structured representations of events, which are defined by an event ontology. An event ontology provides a shared framework to categorize events and their argument roles, ensuring consistency in how events, triggers, and arguments are identified across systems. To ensure robustness, we use datasets with different ontologies.

EE typically consists of two sub-tasks: *event detection (ED)* and *event argument extraction (EAE)*. Event detection identifies event triggers which are keywords or phrases that indicate an event (e.g., *"airstrike"* in Figure 1) and classifies them into event types (e.g., *"conflict.attack.selfdirectedbattle"* in Figure 1). Event argument extraction (EAE) focuses on identifying arguments associated with the event and assigning specific roles to them. For instance, EAE would identify *"convoy"* as an argument for the *"target"* role (in Figure 1).

In this paper, we focus specifically on *evaluating* EAE. Evaluation involves comparing the human-annotated argument with the model's prediction. We adopt the terminology of Bulian et al. (2022) and refer to human-annotated arguments as *reference arguments* and the model-predicted arguments as *candidate arguments*.

3 Related Work

Evaluating models in tasks like text generation, question answering, and event argument extraction remains a significant challenge. Traditional methods, often based on exact span or text match metrics, tend to overlook the variability of possible correct candidates. In this work, we focus specifically on evaluating event argument extraction. We begin by reviewing prior research on event extraction evaluation, followed by a discussion of the challenges in directly comparing models for EE and EAE. Finally, we conclude by examining proposed metrics that address the limitations of exact span matching in other NLP tasks.

Evaluation of End-to-End Event Extraction End-to-end event extraction (EE) typically uses role-averaged evaluation. However, Zheng et al. (2021) argues this can mislead downstream tasks, as a single incorrect argument can change the meaning of the event instance. They propose metrics that treat the event as a whole, explicitly penalizing wrongly identified event arguments. In contrast, our focus is not on how event argument evaluation should integrate into overall event evaluation. Instead, we focus on accurately determining what constitutes a correct event argument prediction, ensuring precise assessment at the role level.

Direct Comparison is Challenging Recent studies highlight several challenges in evaluating event extraction models and propose solutions. Peng et al. (2023b) show that models are not directly comparable due to inconsistencies in data processing, differing output spaces, and the absence of an evaluation pipeline. Omnievent (Peng et al., 2023a) released a toolkit addressing these issues. Echoing this work, Huang et al. (2024) identified additional challenges, such as divergent data assumptions and

split bias. They addressed these challenges by releasing the TextEE framework, which incorporates more diverse datasets as well as recent methods. In our study, while we adopt TextEE to compare models, our primary focus is on defining correctness at the event argument level. By evaluating several models on two different datasets, we demonstrate that using the exact span match metric to assess model performance differs substantially from human judgments of correctness.

Alternatives to Exact Span Match To overcome the limitations of exact span match evaluation, several alternative metrics have been proposed, particularly in text generation tasks.

First, *word-level evaluation metrics* such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and NIST (Doddington, 2002) assess similarity between generated and reference texts using n-gram matching. Second, *string distance-based methods*, as detailed in Cohen et al. (2003), calculate string similarity measures like edit distance to evaluate the closeness of texts. Third, *entity-based evaluation methods*, such as CEAF-REE (Du et al., 2021) in the role-filler entity extraction task, as well as Coref F1 (Li et al., 2021) in EAE task assess candidates at the entity level, considering all coreferent mentions of an entity as equivalent. Fourth, *learned metrics* such as BLEURT (Sellam et al., 2020), COMET (Rei et al., 2020), and BERTScore (Zhang et al., 2020a) have moved beyond surface-level overlap by utilizing word embeddings, contextual embeddings, and learned models to evaluate semantic similarity. Fifth, approaches that incorporate *additional human annotations* and require training models for evaluation have been proposed (Risch et al., 2021; Chen et al., 2020; Bulian et al., 2022).

In contrast to these alternatives to ESM, BEMEAE does not require additional human annotations. Our deterministic components employ simple rules to handle textual variations that do not impact correctness. The semantic component uses either human judgments or off-the-shelf methods, enabling effective evaluation without the need for extensive additional resources.

4 Datasets, Framework, and Models

Datasets We use two datasets in this study: RAMS (Ebner et al., 2020) and GENEVA (Parekh et al., 2023). RAMS, based on the AIDA-1 ontology[2], consists of documents from news articles and includes 139 event types and 65 argument types. It annotates event-argument structures within a 5-sentence window, allowing arguments to appear either in the same sentence as the trigger or in neighboring sentences. GENEVA, in contrast, is a general-domain dataset with its own ontology. Although it includes fewer event types (115) than RAMS, it captures a broader range of argument types (220). GENEVA annotates event-argument structures within a sentence, focusing on more localized contexts. Refer to Table 1 for comparison.

	RAMS	GENEVA
Domain	News	General
Event Ontology	AIDA	Geneva
# Event Types	139	115
# Roles	65	220
# Documents	9,107	3,684
# Events	9,107	7,505
# Arguments	21,206	12,269
Avg # per document		
Sentences	4.70	1.04
Events	1.00	2.04
Arguments	2.33	3.33

Table 1: Basic statistics of RAMS and GENEVA as seen in TextEE standardized data. RAMS uses news as source texts and annotates one event per document (length: 4.7 sentences). In contrast, GENEVA uses text from several domains and annotates two events per document (length: one sentence). GENEVA considers less event types but over three times the amount of argument types (65 vs. 220); documents in GENEVA have on average one more argument.

Framework and Models In our study, we adopt TextEE (Huang et al., 2024), a standardized and reproducible framework for event extraction. It offers uniform data preprocessing scripts, five standardized splits across multiple datasets, and implementations of recent event extraction methods. This uniformity ensures fair comparisons and removes biases related to datasets or splits, leading to a more accurate assessment of model performance.

By using TextEE, we evaluate the top nine models based on their average F1 scores with the RAMS and GENEVA datasets. This includes four classification-based models and five generation-based models. The classification models, which frame EAE as token classification, sequential labeling, or question answering, include

[2]https://www.darpa.mil/program/active-interpretation-of-disparate-alternatives

System	RAMS (F1, rank)	
	5 Splits	Split 1
PAIE (Ma et al., 2022)	50.5 (1)	52.7 (1)
TagPrime-CR (Hsu et al., 2023a)	49.7 (2)	51.4 (2)
TagPrime-C (Hsu et al., 2023a)	48.3 (3)	48.3 (3)
X-Gear (Huang et al., 2022)	46.2 (5)	47.0 (4)
AMPERE (Hsu et al., 2023b)	46.8 (4)	46.5 (5)
DEGREE (Hsu et al., 2022)	45.5 (6)	46.5 (6)
BART-Gen (Li et al., 2021)	45.4 (7)	46.0 (7)
CRF Tagging (Huang et al., 2024)	43.5 (9)	43.8 (8)
EEQA (Du and Cardie, 2020)	44.7 (8)	43.4 (9)

Table 2: Results (F1 and rank) obtained with RAMS, (a) average score of the 5 data splits and (b) training and testing with Split 1. There are only minimal differences in F1 and only two changes in rank (4 and 5; 8 and 9). Therefore, we use the latter setting in our analyses as it decreases the computational demands by 80%. We observe the same trend with GENEVA (Appendix A).

two variants of TagPrime (Hsu et al., 2023a) (TagPrime-C and TagPrime-CR referred to as T-C and T-CR) , EEQA (Du and Cardie, 2020) and CRF-Tagging (CRF-T). The generation-based models frame the task as conditional generation and include PAIE (Ma et al., 2022), X-Gear (Huang et al., 2022), AMPERE (Hsu et al., 2023b), DEGREE (Hsu et al., 2022), and BART-Gen (Li et al., 2021). All classification models use RoBERTa-large (Liu et al., 2019), while all generation-based models use BART-large (Lewis et al., 2020).

In Table 2, we compare the performance of these models in two settings: (a) the average F1 score across five data splits and (b) training and testing using only Split 1. Since we observe minimal differences in F1 scores between the two settings, we choose Split 1 for our analyses. This choice significantly reduces computational demands without sacrificing accuracy. We observe the same trend with the GENEVA dataset, as shown in Table 9.

5 Limitations of Exact Span Match

The most commonly used definition of correctness in event argument extraction is Exact Span Match (ESM), where a candidate argument is considered correct only if both the offsets and role exactly match the reference argument. Unless specified otherwise, in this paper, "F1 score" refers to the ESM F1 Score. Despite being a de facto standard, as shown in Table 3 and discussed below, we observe several limitations with ESM.

One limitation arises when identical tokens appear multiple times in a document, as in the "Marseille" example (Limitation 1), where both mentions refer to the same entity and should be treated as correct. However, exact span matching fails to account for this. Similarly, candidate arguments with slight token variations — such as articles, determiners, or punctuation (Limitation 2a) or non-informative tokens (Limitation 2b)—are unfairly penalized for these minor differences.

More significant limitations arise with informative variations (Limitation 3), where additional details, such as "South Korean" in "South Korean vehicles," provide correct information without changing the core meaning, yet are penalized for including additional information. In both datasets, we observed that coreferent mentions (e.g., pronouns) were sometimes annotated as reference arguments, even when more informative alternatives, such as names, were available. Yet, exact span match fails to account for equivalence between mentions like "Clinton" and "she" (Limitation 4a), which should be considered interchangeable in context.

Similarly, non-coreferent mentions, such as the metonymic use of "U.S." (Limitation 4b), may also be valid, but exact span match lacks the flexibility to capture these nuances. Finally, exact span match struggles with lists (Limitation 5), where combining elements or listing them individually conveys the same meaning even though ESM fails to indicate so.

6 BEMEAE: Beyond Exact Match for Event-Argument Extraction

To address some of the limitations outlined in Section 5 and assess their impact on model performance and rankings, we propose a novel evaluation metric for event argument extraction: BEMEAE. This metric introduces (1) a series of deterministic components to account for textual variations that do not affect the correctness of an argument and (2) a semantic matching component to identify additional correct candidate arguments. While BEMEAE does not resolve all the limitations observed with ESM, we demonstrate that by effectively addressing some of the significant limitations, it provides a more accurate assessment of system performance and leads to significant changes in system rankings.

6.1 Deterministic Components

Same text, not just spans As shown in Limitation 1 of Table 3, exact span match incorrectly

Limitation	Example (and role of the argument)
1. Identical texts ↪ The same tokens may occur twice in the document; both should be counted correct as they are indistinguishable.	[...] [Marseille]$^{\text{candidate}}$ anti - semitic [stabbing]$^{\text{event\_trigger}}$ Last week , a Jewish teacher in the French city of [Marseille]$^{\text{reference}}$ was the target of an anti - Semitic machete attack. (PLACE)
2. Uninformative variations	
2a. Articles, determiners, punctuation. ↪ These tokens barely change meaning; including or missing them is harmless.	[...] Iran eventually did come to the [negotiating]$^{\text{event\_trigger}}$ table and [the [Obama administration]$^{\text{reference}}$]$^{\text{candidate}}$ was able to work out its historic Iran Deal in a case where diplomacy proved to be the correct course of action. (PARTICIPANT)
2b. Other ↪ Additional tokens are sometimes entailed; including them is harmless.	It was [[Bush's administration]$^{\text{reference}}$, not Obama's]$^{\text{candidate}}$, that negotiated event trigger the 2009 agreement that called for the withdrawal of all U.S. forces from Iraq by Dec. 31, 2011. (PARTICIPANT)
3. Informative variations ↪Additional tokens providing details are valid (e.g., ownership, origin).	[...] [South Korean [vehicles]$^{\text{reference}}$]$^{\text{candidate}}$ [transporting]$^{\text{event\_trigger}}$ employees working at the Kaesong Industrial Complex (KIC) arrive at the South's CIQ (Customs, Immigration and Quarantine) [...] (TRANSPORTER)
4. Alternative mentions	
4a. Coreferent ↪ Coreferent mentions refer to the same entity; all of them should get credit as they are interchangeable in context.	[...] Dr. Lisa Bardack adds that [Clinton]$^{\text{reference}}$ is "recovering well with antibiotics and rest" after she became overheated, [dehydrated]$^{\text{event\_trigger}}$ and felt dizzy at a 9/11 memorial ceremony on Sunday. Clinton's aides say [she]$^{\text{candidate}}$'ll return to the campaign trail on Thursday. (VICTIM)
4b. Non-coreferent ↪ Mentions to other entities may be correct, especially when using metonymy.	Assange has characterized the investigation as part of a broader conspiracy on the part of the [[U.S.]$^{\text{candidate}}$ government]$^{\text{reference}}$ to [incarcerate]$^{\text{event\_trigger}}$ him, extradite him, and have him killed in prison—"Jack Ruby style," as he put it [...] (JAILER)
5. Aggregated Lists ↪ One argument with all the elements of a list and several arguments with the individual elements are equivalent.	The statement, citing GCC Secretary General Rashid al-Zayani, accused Hezbollah of recruitment to carry out terrorist attacks, and of [smuggling]$^{\text{event\_trigger}}$ [[weapons]$^{\text{candidate}}$ and [explosives]$^{\text{candidate}}$]$^{\text{reference}}$, "in flagrant violation of GCC members"' sovereignty, security and stability." (ARTIFACT)

Table 3: Limitations and examples of Exact Span Match F1 as an evaluation metric for event argument extraction (with respect to the reference annotation). All the candidates are real predictions by the systems we analyze in this paper. Note that all differences between the candidate and reference arguments are harmless (i.e., all candidates are correct), even though some candidates include additional details (e.g., informative variations). Exact Span Match F1, the standard metric, considers all the candidates wrong despite being correct according to human assessments.

classifies identical text that refer to the same entity appearing in different positions as incorrect. To address this issue, we compare candidate and reference arguments (excluding pronouns) based on their uncased textual content rather than their offsets.

Uninformative Variations In Limitation 2a of Table 3, we highlight how exact span match fails to account for uninformative variations. To partially address this limitation, we identify and remove uninformative tokens that do not alter the meaning of the span. Specifically, we remove articles, determiners, punctuation, and the Saxon genitive ('s). We then check for a textual match between the cleaned reference and candidate arguments.

Breaking Down (and Aggregating) Arguments Exact span match struggles with granularity, as shown in Limitation 5 of Table 3. In the example, the reference argument is *"weapons and explosives"* but the model predicts *"weapons"* and *"explosives"* separately. ESM penalizes the model despite the functional equivalence. To address this, we process candidate arguments for a given role by handling conjunctions such as "and". Specifically, we either break them into separate arguments or aggregate separate arguments using "and" as needed to check for a match with the reference.

Informative Variations In Limitation 3 of Table 3, exact span match overlooks candidate arguments that contain informative modifiers that add context without altering correctness. To partially address this limitation, we incorporate modifiers when evaluating candidate arguments.

Specifically, we generate modified versions of single-token reference arguments[3] by adding relevant modifiers identified via dependency parsing using the spaCy toolkit (Honnibal et al., 2020). To prevent spurious content, we restrict modifiers to the following dependency labels: adjectival modi-

[3]We restrict this process to single-token references to maintain precision, acknowledging that modifiers may be crucial to correctness in multi-token references.

(1) Also , he has argued unsuccessfully in British courts that the investigation has been a ruse to extradite him to Sweden so he could then be extradited to the [United States]$^{\text{reference}}$ to face reported but still-secret [U.S.]$^{\text{candidate}}$ [charges]$^{\text{event\_trigger}}$.

(2) [. . .] anthrax-tainted letters have made policymakers and the general public concerned that terrorists could [use]$^{\text{event\_trigger}}$ [[chemical or biological weapons]$^{\text{candidate}}$ (CBW)]$^{\text{reference}}$ to inflict mass casualties.

(3) [. . .] with experts saying [extradition]$^{\text{event\_trigger}}$ of a [Romanian hacker]$^{\text{reference}}$ who revealed her unsecured email server could spell deep trouble for Clinton's presidential campaign. Last week , U.S. officials moved to extradite the hacker known as ['Guccifer']$^{\text{candidate}}$, who is accused of [. . .]

(4) [. . .] Christina Grimmie, 22, who was an American singer and songwriter, known for her participation in the NBC singing competition The Voice, was signing autographs at a [concert venue in [Orlando]$^{\text{reference}}$]$^{\text{candidate}}$ on 10 June when an assailant [shot]$^{\text{event\_trigger}}$ her.

Table 4: Examples of event triggers, reference arguments, and candidate arguments for the roles of PROSECUTOR, INSTRUMENT, DEFENDANT, and PLACE. Neither Exact Span Match nor the four deterministic components of BEMEAE consider these candidates correct, but human annotators do (i.e., via the semantic matching component). These examples underscore the critical role of the semantic matcher in BEMEAE.

fiers (amod), appositional modifiers (appos), nominal modifiers (nmod), noun modifiers (nounmod), numeric modifiers (nummod), possessives (poss), and possessive markers (possessive). We then compare these more informative version of reference arguments to the candidate arguments. To validate this deterministic component, we manually analyzed 100 candidate arguments initially flagged as errors by exact span matching but deemed correct by this step. Our analysis showed 100% accuracy.

6.2 Semantic Matching Component

Human Assessments We sampled 500 candidate arguments from RAMS and GENEVA that were still identified as errors after applying our deterministic steps. To select these samples, we chose the top 500 erroneous predictions made by models. Each candidate argument was reviewed by two human annotators, who were provided with the relevant document and asked to answer three specific questions. Details about the annotators and the annotation interface are provided in Appendix B.

Based on their responses to the questions, each candidate argument was assigned one of the following three labels: *Correct*: The candidate argument is the correct argument role of the event trigger. *Partial*: The candidate argument is the correct argument role of the event trigger but is missing important information. *Incorrect*: The candidate argument is not the argument role of the event trigger or adds spurious information. The inter-annotator reliability, measured by Cohen's kappa (κ), was 0.84 for RAMS and 0.95 for GENEVA, indicating almost perfect agreement.

Although our deterministic components improve upon ESM, they cannot capture all semantically equivalent reference-candidate pairs that ESM misidentified as errors. Table 4 presents examples of candidate arguments that human annotators determined to be correct but were classified as errors by both ESM and our deterministic components.

In the first example, the candidate argument is an *abbreviation* of the reference. In the second, the candidate omits the *acronym* included in the reference. The third example presents the candidate as an *alternate name* for the reference, and in the final example, the candidate provides *additional detail* compared to the reference. These examples highlight the importance of incorporating a semantic matching component to ensure accurate evaluation.

Automatic Assessments While human judgment is the most reliable, it is impractical to apply it to every prediction due to time and cost constraints. Therefore, we assessed the following methods to approximate human assessments and analyzed their agreement with human annotators.

(a) Cosine Similarity: First, we calculate the cosine similarity between the average word embeddings of the reference and candidate arguments. Then, we train a logistic regression classifier[4] on this cosine similarity as the only feature (labels: Correct, Partial, or Incorrect). We evaluated this approach using word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014)[5] and sentence-BERT embeddings (Reimers and Gurevych, 2019)[6].

(b) BERTScore: Following the same setup as above, we replace cosine similarity with BERTScore (Zhang et al., 2020a) with DeBERTa embeddings (He et al., 2021).[7]

[4]Using Scikit-learn(Buitinck et al., 2013)

[5]Using Gensim (Řehůřek and Sojka, 2010)

[6]Using HuggingFace (Wolf, 2019)

[7]DeBERTa was chosen for its high Pearson correlation with human annotations in WMT16.

	κ_3	κ_2
Human Semantic Matcher	0.84	0.88
Automatic Semantic Matchers		
Cosine similarity		
word2vec	0.05	0.02
GloVe	0.05	0.03
SentenceBERT	0.20	0.20
BERTScore	0.02	0.01
Prompting GPT-4o-mini		
wording: intuitive, reference: yes	0.10	0.03
wording: intuitive, reference: no	0.19	0.16
wording: expert, reference: yes	0.30	0.16
wording: expert, reference: no	0.26	0.24
Prompting GPT-4o		
wording: intuitive, reference: yes	0.20	0.15
wording: intuitive, reference: no	0.17	0.09
wording: expert, reference: yes	0.43	0.43
wording: expert, reference: no	0.34	0.38

Table 5: Cohen's κ scores for 500 candidate arguments from the RAMS dataset. First row shows agreement between human annotators. Other rows show agreements between automatic methods and humans. κ_3 uses three labels (Correct, Partial, Incorrect), and κ_2 merges Partial and Incorrect. GPT-4o with expert instructions and explicit references achieves the highest agreement among automatic methods. However, it remains far from reliable human annotation ($\kappa = 0.43$) and obtains much lower agreement than humans ($\kappa = 0.84, 0.88$). A similar trend is observed with the GENEVA dataset (Appendix C).

(c) GPT Models: We utilized GPT-4o and GPT-4o mini (OpenAI, 2023), experimenting with four different configurations: (1) with expert or intuitive instructions, and (2) with and without the reference in the prompt. Further details on these configurations, including the prompts are in Appendix D.

We evaluated Cohen's kappa (κ) between the automated methods and human judgments under two settings: first, using the original three labels (Correct, Partial, and Incorrect); and second, under a stricter definition where Partial is merged with Incorrect. The results for RAMS are presented in Table 5. We observed the highest agreement with GPT-4o when expert instructions were provided and the reference was shared.

7 Insights from BEMEAE

In this section, we present the results from evaluating nine EAE models using BEMEAE and compare them to the baseline ESM. The incremental F1 improvements from each BEMEAE component are presented in Table 6 for RAMS, and Table 7 for GENEVA. We also examine two versions of the semantic matcher: one based on human judgments (BEMEAE $_{\text{Human}}$) and the other on the best automated method (BEMEAE $_{\text{Best Automated}}$). Key insights from the evaluation are summarized below.

BEMEAE indicates higher F1, but gains are uneven. A consistent trend in our evaluation is that for most BEMEAE components, the F1 score increases with each successive step across all models. As BEMEAE moves from addressing harmless tokens to accounting for modifiers and semantic variations, it consistently captures more valid candidate predictions than ESM. However, the magnitude of these improvements varies significantly across models.

One of the starkest differences appears in the "Breaking & Aggregating" component, which leads to no gains for most models. However, for AMPERE, DEGREE, and BART-Gen, it results in an average 2-point F1 increase with RAMS (Table 6), and an average 6.6-point F1 increase with GENEVA (Table 7). This suggests that exact span match disproportionately penalizes certain models for failing to conform to strict span boundaries.

BEMEAE substantially improves overall F1, but uneven gains lead to rank changes. The uneven gains across BEMEAE components accumulate, leading to significant shifts in model rankings, as seen in Table 8. While all models experience F1 score increases under BEMEAE, the magnitude of these gains vary considerably.

For example, when comparing model ranking based on ESM and BEMEAE $_{\text{Human}}$, BART-Gen's F1 score increases by 15 percentage points with RAMS (46% to 61%) and 12 percentage points with GENEVA (64.5% to 76.5%), boosting its rank from 7th to 2nd in RAMS and from 9th to 6th in GENEVA. In contrast, EEQA shows a smaller improvement, gaining 6.1 points with RAMS and 6.2 points with GENEVA, maintaining its low position in RAMS while dropping two ranks in GENEVA.

Exact Span Match Does Not Correlate Well with Human Judgments. If F1 improvements had been consistent across models, ESM might still serve as a reasonable evaluation metric for ranking models. However, a comparison of model rankings from ESM and BEMEAE $_{\text{Human}}$ show only moderate correlation (Kendall's τ = 0.44 for RAMS, 0.67 for GENEVA; Table 8). In contrast, BEMEAE $_{\text{Best Automated}}$ closely aligns with

Metric	PAIE	T-CR	T-C	X-Gear	AMPERE	DEGREE	BART-Gen	CRF-T	EEQA
Exact Span Match	52.7	51.3	48.3	47.0	46.5	46.5	46.0	43.8	43.4
BEMEAE (Ours)									
Deterministic comp									
same text	53.8 (1.1)	52.2 (0.9)	49.4 (1.1)	48.4 (1.4)	47.7 (1.2)	47.9 (1.4)	47.8 (1.8)	44.7 (0.9)	44.1 (0.7)
+ harmless tokens	56.6 (2.8)	55.0 (2.8)	52.7 (3.3)	51.7 (3.3)	50.6 (2.9)	51.0 (3.1)	50.9 (3.1)	47.8 (3.1)	45.3 (1.2)
+ breaking & aggreg.	56.6 (0.0)	55.0 (0.0)	52.8 (0.1)	51.7 (0.0)	52.5 (1.9)	52.9 (1.9)	53.1 (2.2)	47.8 (0.0)	45.3 (0.0)
+ modifiers	58.1 (1.5)	56.6 (1.6)	54.1 (1.3)	53.0 (1.3)	54.0 (1.5)	54.0 (1.1)	54.5 (1.4)	49.6 (1.8)	46.0 (0.7)
+ Semantic Matcher									
human	62.9 (4.8)	60.0 (3.4)	58.4 (4.3)	57.6 (4.6)	58.7 (4.7)	59.4 (5.4)	61.0 (6.5)	53.5 (3.9)	49.5 (3.5)
best automatic	61.5 (3.4)	59.1 (2.5)	57.0 (2.9)	56.4 (3.4)	57.4 (3.4)	57.6 (3.6)	58.5 (4)	52.3 (2.7)	48.4 (2.4)

Table 6: Evaluation of several systems on RAMS using Exact Span Match (F1) and several versions of our metric (F1 and Δ with respect to the previous step). Results are presented incrementally, adding the deterministic components one at a time (Section 6). Two versions of the semantic matcher are also compared: human judgments (tedious but reliable) and GPT-4o with expert instructions and reference shared (automatic but less reliable). Notably, BEMEAE, our metric, reveals significant differences in system improvements, which ultimately lead to changes in system rankings (Table 8). We observe a similar trend with GENEVA (Table 7).

Metric	T-CR	T-C	X-Gear	CRF-T	PAIE	DEGREE	EEQA	AMPERE	BART-Gen
Exact Span Match	80.3	78.3	73.9	70.7	70.5	66.2	66.0	65.7	64.5
BEMEAE (Ours)									
Deterministic comp									
same text	80.6 (0.3)	78.6 (0.3)	74.8 (0.9)	71.2 (0.5)	70.7 (0.2)	67.2 (1.0)	66.4 (0.4)	66.6 (0.9)	65.3 (0.8)
+ harmless tokens	82.4 (1.8)	80.5 (1.9)	77.0 (2.2)	73.1 (1.9)	72.3 (1.6)	69.0 (1.8)	67.9 (1.5)	68.2 (1.6)	66.9 (1.6)
+ breaking & aggreg.	82.4 (0.0)	80.5 (0.0)	77.0 (0.0)	73.1 (0.0)	72.3 (0.0)	75.7 (6.7)	67.9 (0.0)	74.9 (6.7)	73.3 (6.4)
+ modifiers	82.4 (0.0)	80.6 (0.2)	77.2 (0.2)	73.2 (0.1)	72.4 (0.1)	75.8 (0.1)	68.0 (0.1)	75.0 (0.1)	73.4 (0.1)
+ Semantic Matcher									
human	85.7 (3.3)	84.5 (3.9)	80.7 (3.5)	77.8 (4.6)	76.0 (3.6)	78.1 (2.3)	72.2 (4.2)	77.0 (2.0)	76.5 (3.1)
best automatic	83.6 (1.2)	81.8 (1.2)	78.1 (0.9)	74.5 (1.3)	73.1 (0.7)	76.4 (0.6)	69.8 (1.8)	75.6 (0.6)	74.3 (0)

Table 7: Evaluation of several systems on GENEVA using Exact Span Match (F1) and several versions of our metric (F1 and Δ with respect to the previous row). The deterministic components are always correct (Section 6); we present results incrementally adding components. We also present results with two versions of the semantic matcher: humans (tedious but reliable) and the best model (automatic but less reliable). Crucially, BEMEAE, our metric, reveals substantially different improvements across systems, resulting in ranking changes (Table 8). This table complements Table 6 by providing detailed F1 scores with GENEVA instead of RAMS.

BEMEAE $_{\text{Human}}$, achieving a high Kendall's τ of 0.94 for both datasets.

This highlights that ESM is limited in its effectiveness to approximate human judgment. By recognizing more candidate arguments semantically equivalent to the reference, BEMEAE offers a more accurate evaluation that is more reflective of human judgment, which ultimately changes model rankings and provides a better understanding of how models perform against each other.

Automated Methods for Semantic Matcher are reliable for ranking, but not for estimating absolute F1. The best automated semantic matcher in our study demonstrated only fair agreement with human judgment, achieving a Cohen's kappa (κ) of 0.43 with RAMS (Table 5) and 0.31 for GENEVA (Table 10). Furthermore, model rankings produced by BEMEAE $_{\text{Best Automated}}$ showed close alignment with those from BEMEAE $_{\text{Human}}$, as indicated by a high Kendall's τ score of 0.94 for both RAMS and GENEVA (Table 8), demonstrating that automated semantic matchers are reliable for ranking.

However, automated matchers fall short when estimating absolute F1 scores. The gap between BEMEAE $_{\text{Human}}$ and BEMEAE $_{\text{Best Automated}}$ is significant, with models like BART-Gen showing F1 differences of 2.5 points in RAMS and 2.2 points in GENEVA. This highlights that even the best automated matchers in our study, despite their ranking reliability, fail to capture all valid candidate arguments that a human would recognize as correct. As a result, our best automated matcher cannot provide absolute F1 scores, as a "true" F1 would require either a perfect semantic matcher—which does not yet exist—or human evaluation, such as BEMEAE $_{\text{Human}}$.

System	Exact Span Match F1	BEMEAE (Our metric)			
		Human		automatic	
		F1 (Δ)	Rank	F1 (Δ)	Rank
PAIE	52.7	62.9 (10.2)	1	61.5 (8.8)	1
T-CR	51.3	60.0 (8.7)	3 ↓1	59.1 (7.8)	2
T-C	48.3	58.4 (10.1)	6 ↓3	57.0 (8.7)	6 ↓3
X-Gear	47.0	57.6 (10.6)	7 ↓3	56.4 (9.4)	7 ↓3
AMPERE	46.5	58.7 (12.2)	5	57.4 (10.9)	5
DEGREE	46.5	59.4 (12.9)	4 ↑2	57.6 (11.1)	4 ↑2
BART-Gen	46.0	61.0 (15.0)	2 ↑5	58.5 (12.5)	3 ↑4
CRF-T	43.8	53.5 (9.7)	8	52.3 (8.5)	8
EEQA	43.4	49.5 (6.1)	9	48.4 (5.0)	9

System	Exact Span Match F1	BEMEAE (Our metric)			
		Human		Automatic	
		F1 (Δ)	Rank	F1 (Δ)	Rank
T-CR	80.3	85.7 (5.4)	1	83.6 (3.3)	1
T-C	78.3	84.5 (6.2)	2	81.8 (3.5)	2
X-Gear	73.9	80.7 (6.8)	3	78.1 (4.2)	3
CRF-T	70.7	77.8 (7.1)	5 ↓1	74.5 (3.8)	6 ↓2
PAIE	70.5	76.0 (5.5)	8 ↓3	73.1 (2.6)	8 ↓3
DEGREE	66.2	78.1 (11.9)	4 ↑2	76.4 (10.2)	4 ↑2
EEQA	66.0	72.2 (6.2)	9 ↓2	69.8 (3.8)	9 ↓2
AMPERE	65.7	77.0 (11.3)	6 ↑2	75.6 (9.9)	5 ↑3
BART-Gen	64.5	76.5 (12.0)	7 ↑2	74.3 (9.8)	7 ↑2

Table 8: Evaluation of systems on RAMS (left) and GENEVA (right) using ESM and BEMEAE. We present results with the deterministic components and two semantic matchers: humans (BEMEAE $_{\text{Human}}$) and the best automatic matcher (BEMEAE $_{\text{Best Automated}}$). ESM rankings do not correlate well with BEMEAE $_{\text{Human}}$ (Kendall's $\tau = 0.44$ with RAMS and 0.67 with GENEVA), indicating that the rank changes between ESM and BEMEAE are substantial. However, the similarity of the rankings obtained with BEMEAE $_{\text{Human}}$ and BEMEAE $_{\text{Best Automated}}$ (Kendall's $\tau = 0.94$) suggests a scalable and reliable alternative.

As demonstrated in our analysis, BEMEAE is a significant improvement over ESM, providing an evaluation that aligns much more closely with human judgments. By moving beyond exact span matching and recognizing a broader range of candidate arguments that humans would consider semantically equivalent to the reference and thus correct, BEMEAE offers a more realistic and faithful assessment of model performance.

8 Conclusion

We evaluated nine EAE models on the RAMS and GENEVA datasets, highlighting the limitations of the traditional exact span match (ESM) evaluation metric. To address these shortcomings, we introduced BEMEAE, a novel metric that combines deterministic components with both human and automated semantic matching. Our results show that BEMEAE provides a far more accurate and fair evaluation since it accounts for a broader range of argument representations resulting in significantly shifted model ranks. When compared, ranks based on ESM show poor correlation with BEMEAE $_{\text{Human}}$, whereas BEMEAE $_{\text{Best Automated}}$ achieves strong correlation, presenting a scalable and reliable alternative.

Limitations

While BEMEAE improves evaluation over ESM, it cannot capture all correct candidate arguments, as even the best automated semantic matcher showed only fair agreement with human judgments, highlighting a gap in modeling semantic equivalence. We report results using GPT-4o, as tested open-source alternatives struggled with prompt adherence and produced verbose responses. Additionally, larger models like GPT demand substantial computational resources. Our evaluation remains limited to general and news-based English datasets, leaving its effectiveness in specialized domains like cybersecurity untested.

Acknowledgments

This research was supported by a grant from the U.S. Office of Naval Research (N00014-22-1-2596). We would like to thank the anonymous reviewers for their insightful comments and suggestions.

References

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 291–305, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. MOCHA: A dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6521–6532, Online. Association for Computational Linguistics.

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the 2003 International Conference on Information Integration on the Web*, IIWEB'03, page 73–78. AAAI Press.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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. 2021. 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.

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.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

I-Hung Hsu, Kuan-Hao Huang, Shuning Zhang, Wenxin Cheng, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2023a. TAGPRIME: A unified framework for relational structure extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12917–12932, Toronto, Canada. Association for Computational Linguistics.

I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. 2023b. AMPERE: AMR-aware prefix for generation-based event argument extraction model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10976–10993, Toronto, Canada. Association for Computational Linguistics.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4633–4646, Dublin, Ireland. Association for Computational Linguistics.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. TextEE: Benchmark, reevaluation, reflections, and future challenges in event extraction. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 12804–12825, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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.

Wei Li, Lei He, and Hai Zhuge. 2016. Abstractive news summarization based on event semantic link network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 236–246, Osaka, Japan. The COLING 2016 Organizing Committee.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

OpenAI. 2023. Gpt-4 technical report. Accessed: 2024-10-11.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Tanmay Parekh, I-Hung Hsu, Kuan-Hao Huang, Kai-Wei Chang, and Nanyun Peng. 2023. GENEVA: Benchmarking generalizability for event argument extraction with hundreds of event types and argument roles. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3664–3686, Toronto, Canada. Association for Computational Linguistics.

Hao Peng, Xiaozhi Wang, Feng Yao, Zimu Wang, Chuzhao Zhu, Kaisheng Zeng, Lei Hou, and Juanzi Li. 2023a. OmniEvent: A comprehensive, fair, and easy-to-use toolkit for event understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 508–517, Singapore. Association for Computational Linguistics.

Hao Peng, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023b. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9206–9227, Toronto, Canada. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic answer similarity for evaluating question answering models. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 149–157, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2024. Label Studio: Data labeling software. Open source software available from https://github.com/HumanSignal/label-studio.

Md Nayem Uddin, Enfa Rose George, Eduardo Blanco, and Steven R. Corman. 2024. Asking and answering questions to extract event-argument structures. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1609–1626, Torino, Italia. ELRA and ICCL.

Shichao Wang, Xiangrui Cai, HongBin Wang, and Xiaojie Yuan. 2021. Incorporating circumstances into narrative event prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*,

pages 4840–4849, Punta Cana, Dominican Republic. Association for Computational Linguistics.

T Wolf. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020b. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. Association for Computational Linguistics.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2021. Revisiting the evaluation of end-to-end event extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4609–4617, Online. Association for Computational Linguistics.

A Train and Test Splits with GENEVA

System	GENEVA (F1, rank)	
	5 Splits	Split 1
TagPrime-CR (Hsu et al., 2023a)	80.1 (1)	80.3 (1)
TagPrime-C (Hsu et al., 2023a)	79.0 (2)	78.3 (2)
X-Gear (Huang et al., 2022)	74.9 (3)	73.9 (3)
CRF Tagging (Huang et al., 2024)	74.3 (4)	70.7 (4)
PAIE (Ma et al., 2022)	70.3 (5)	70.5 (5)
DEGREE (Hsu et al., 2022)	63.9 (9)	66.2 (6)
EEQA (Du and Cardie, 2020)	67.0 (6)	66.0 (7)
AMPERE (Hsu et al., 2023b)	64.8 (7)	65.7 (8)
BART-Gen (Li et al., 2021)	64.3 (8)	64.5 (9)

Table 9: Results (F1 and rank) obtained with GENEVA (a) average score of the 5 data splits and (b) training and testing with Split 1. There are only minimal differences in F1. Therefore, we use the latter setting in our analyses as it decreases the compute demands by 80%. This complements the RAMS analysis presented in Table 2, where similarly minimal changes were observed.

As discussed in the main paper (Section 4), the analyses presented here leverage the TextEE framework with two datasets: RAMS and GENEVA. While TextEE recommends running experiments using five different train-test splits and reporting the average performance, we found that using a single split yields only minor changes in results, as shown in Table 2 for RAMS and Table 9 for GENEVA. This approach reduces computational requirements by 80% with minimal impact on outcomes.
We acknowledge that using five splits provides more robust results; however, it is important to note that (a) our primary focus is on evaluating metrics rather than optimizing model performance, and (b) using a single test split allows for more in-depth and manageable analyses across all models.

B Annotation Task Details

The annotation process involved evaluating the candidate arguments produced by the models with respect to human-annotated reference arguments for both RAMS and GENEVA datasets.

The Annotation Task

Each candidate argument was evaluated with a series of three questions.

1. **Validity Check:** Annotators first determined whether any part of the candidate argument was valid for the specific role within the event. This step addressed whether the candidate argument was, or included, a correct answer for the role, including cases where there may be spurious content or coreferent mentions. If the candidate argument was entirely incorrect, the annotators proceeded to the next task, otherwise they are asked the next question.

2. **Similarity Check:** If the candidate argument was deemed valid, the annotators then assessed whether it conveyed the same meaning as any of the reference arguments. The task required choosing whether the candidate argument was semantically equivalent to any reference argument. To account for possible annotation errors, we offered a none option as well.

3. **Relationship Determination:** Finally, annotators identified the relationship between the candidate argument and the chosen reference argument. Several options were provided to describe this relationship:

    - *Conveys at least the needed information*: The candidate argument includes all relevant information present in the reference.
    - *Entirely different from the reference, but correct*: The candidate argument text is not the same as any part of the reference argument but still correct for the assigned role. This accounts for annotation errors, metonyms etc.
    - *Removes important details*: The candidate argument is a more general version, missing important specifics compared to the reference such as predicting the name of the city instead of the specific venue of an event.
    - *Adds spurious content*: The candidate argument contains unnecessary information not relevant to the role.

We used Label Studio (Tkachenko et al., 2020-2024) as our annotation platform. A screenshot with a real example is shared in Figure 2.

Processing Annotations

We processed the annotator's choices as follows: If the annotator selected "No" to Question 1 or marked the relationship as "Adds Spurious Content," the annotation was labeled as 'Incorrect'. If the annotator selected "Yes" to Question 1 and marked either "Conveys at least the needed information" or "Entirely different from reference, but

Figure 2: Our annotation interface for evaluating event argument extraction. In this example the annotator assesses whether the candidate argument "Lazar" is a valid answer to the role "defendant" in the event "extradited". The annotator follows three steps: (1) validity check to determine if "Lazar" is a valid argument for the event role, (2) similarity check to compare "Lazar" with reference argument "he," and (3) relationship determination to assess if the candidate conveys the needed information, removes important details, or adds spurious content.

correct," the annotation was labeled as 'Correct'. Lastly, if the annotator selected "Yes" to Question 1 and marked "Removes important details" in response to Question 3, the annotation was labeled as 'Partial'. The response to Question 2 was used to determine which reference argument was correctly matched. This step was particularly valuable when the candidate was an aggregation of two or more reference arguments, ensuring that multiple references were appropriately evaluated in the final assessment.

Annotators and Agreement

The annotators for this study were a mix of graduate and undergraduate computer science students. To ensure consistency, all annotators received detailed instructions and were trained using a set of 50 tasks to develop a shared understanding of the annotation process. For each task, two annotators independently labeled the candidate arguments. In cases where disagreements occurred, the labeling provided by the more experienced annotator was used in the final analysis. The inter-annotator reliability, calculated using Cohen's kappa (κ), was 0.84 for the RAMS dataset and 0.95 for GENEVA, demonstrating substantial agreement between annotators.

C Cohen's Kappa for Semantic Matchers for GENEVA

We evaluated Cohen's kappa (κ) between the automatic methods and human labels under two settings: first, using the original three labels (Correct, Partial, and Incorrect); and second, under a stricter definition where Partial is merged with Incorrect. The results for GENEVA are presented in Table 10. We observed the highest agreement with GPT-4o-mini when expert instructions were provided and the reference was explicitly shared. A similar trend is observed with the RAMS dataset in Table 5.

	κ_3	κ_2
Human Semantic Matcher	0.95	0.95
Automatic Semantic Matchers		
Cosine similarity		
word2vec	0.25	0.21
GloVe	0.16	0.12
SentenceBERT	0.28	0.24
BERTScore	0.23	0.20
Prompting GPT-4o-mini		
wording: intuitive, reference: yes	0.13	0.03
wording: intuitive, reference: no	0.16	0.17
wording: expert, reference: yes	0.31	0.30
wording: expert, reference: no	0.28	0.27
Prompting GPT-4o		
wording: intuitive, reference: yes	0.14	0.10
wording: intuitive, reference: no	0.18	0.08
wording: expert, reference: yes	0.29	0.32
wording: expert, reference: no	0.28	0.34

Table 10: Cohen's κ scores for 500 candidate arguments from the GENEVA dataset. First row shows agreement between human annotators. Other rows show agreements between automatic methods and humans. κ_3 uses three labels (Correct, Partial, Incorrect), and κ_2 merges Partial and Incorrect. For GENEVA GPT-4o-mini with expert instructions and references shared achieves the highest agreement among automatic methods. However, it remains far from reliable human annotation ($\kappa = 0.31, 0.30$); and obtains much lower agreement than humans ($\kappa = 0.94, 0.95$). Observations with the RAMS dataset are shared in Table 5.

D GPT Prompts for Semantic Matcher

We used GPT-4o and GPT-4o mini (OpenAI, 2023), experimenting with four combinations of two conditions: (1) with or without expert instructions, and (2) with or without the explicit inclusion of the reference argument in the prompt.

System instructions refer to a broader set of guidelines provided to the model before it processes the specific user prompt which is the specific request task request. System instructions offer general context to the model, influencing how it interprets and responds to the user prompt. We give expert instructions in the system prompt.

In our study, we experimented with expert and intuitive instructions, and with and without reference in the prompt. Including the reference provides additional context for the model to compare candidate arguments against, while excluding it forces the model to make a judgment without explicit reference to the expected argument.

Table 11 presents the detailed prompts used in each setting. All other hyperparameters were kept at their default settings.

Prompts with experiment setting
Intutive instructions, Reference shared *User Instructions* Read the following paragraph: <document>. We know that <reference> is the <role> of <event trigger>. Is it true that <candidate> and <reference> have the same meaning? Answer only one of these options: 'Yes', 'Yes, but it is not exactly the same meaning; the first text is missing important information', or 'No'.
Intutive instructions, Reference not shared *User Instructions* Read the following paragraph:<document>. Based on the paragraph, is it true that <candidate> is the <role> of <event trigger>? Answer only one of these options: 'Yes', 'Yes, but it is missing important information', or 'No'
Expert instructions, Reference shared *System Instruction* You are provided with a document, event type, event trigger, argument role, reference argument (i.e., the correct argument for the argument role of the event trigger), and a candidate argument (i.e., a candidate argument for the argument type of the event trigger that needs to be verified). Task: Classify whether the candidate argument is correct based on the reference argument. "Correct": The candidate argument matches the reference argument fully or is the correct argument for the argument role of the event trigger. "Partial": The candidate argument partially matches the reference argument or is a partially correct argument for the argument role of the event trigger. "Incorrect": The candidate argument does not match the reference argument fully and it not the correct argument for the argument role of the event trigger. Output: Return a JSON object with the following structure: "label": "Correct", "Partial" or "Incorrect". "explanation": 1-2 sentences explaining the classification decision. *User Instructions* Document : <document> Event Type: <event type> Event Trigger: <event trigger> Argument Role : <role> Reference argument: <reference> Candidate argument: <candidate>
Expert instructions, Reference not shared *System Instruction* You are provided with a document, event type, event trigger, argument role, and candidate argument (i.e., a candidate argument for the argument type of the event trigger that needs to be verified). Task: Classify whether the candidate argument is correct "Correct": The candidate argument is the argument role of the event trigger. "Partial": The candidate argument is the argument role of the event trigger, but it is missing important information. "Incorrect: The candidate argument is not the argument role of the event trigger or is adding spurious information. Output: Return a JSON object with the following structure: "label": "Correct","Partial" or "Incorrect". "explanation": 1-2 sentences explaining the classification decision. *User Instructions* Document : <document> Event Type: <event type> Event Trigger: <event trigger> Argument Role : <role> Candidate argument: <candidate>

Table 11: Prompts used for evaluating event argument extraction. We explore variations with and without expert instructions, as well as with and without the explicit inclusion of the reference argument. The <item> placeholders indicate where the actual content is inserted (e.g., document, event type, or argument role).