Argument-Aware Approach To Event Linking

I-Hung Hsu[*‡] **Zihan Xue**[*†] **Nilay Pochhi**[†] **Sahil Bansal**[†]
Premkumar Natarajan[‡] **Jayanth Srinivasa**[§] **Nanyun Peng**[†]
[‡]Information Science Institute, University of Southern California
[†]Computer Science Department, University of California, Los Angeles
[§]Cisco Research
{ihunghsu, pnataraj}@isi.edu, {zihanxue}@ucla.edu
{jasriniv}@cisco.com, {violetpeng}@cs.ucla.edu

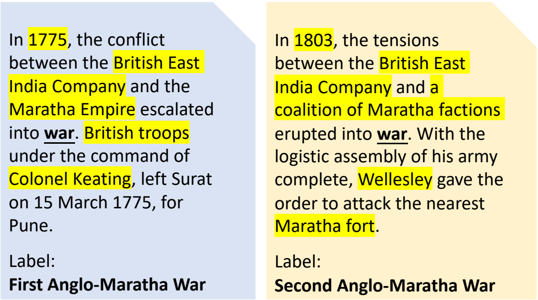
Abstract

Event linking connects event mentions in text with relevant nodes in a knowledge base (KB). Prior research in event linking has mainly borrowed methods from entity linking, overlooking the distinct features of events. Compared to the extensively explored entity linking task, events have more complex structures and can be more effectively distinguished by examining their associated arguments. Moreover, the information-rich nature of events leads to the scarcity of event KBs. This emphasizes the need for event linking models to identify and classify event mentions not in the KB as "out-of-KB," an area that has received limited attention. In this work, we tackle these challenges by introducing an argument-aware approach. First, we improve event linking models by augmenting input text with tagged event argument information, which facilitates the recognition of key information about event mentions. Subsequently, to help the model handle "out-of-KB" scenarios, we synthesize out-of-KB training examples from in-KB instances through controlled manipulation of event arguments. Our experiments across two test datasets showed significant enhancements in both in-KB and out-of-KB scenarios, with a notable 22% improvement in out-of-KB evaluations.

1 Introduction

Event Linking (Nothman et al., 2012; Ou et al., 2023) involves associating mentions of events in text with corresponding nodes in a knowledge base (KB). This process of linking nodes can enhance text comprehension to facilitate various downstream applications, such as question-answering and recommendation systems (Yu et al., 2023b; Li et al., 2020; Jacucci et al., 2021).

Despite its significance, event linking remains a challenging and relatively under-explored task (Yu et al., 2023a; Pratapa et al., 2022). In contrast to entities that generally maintain consistent attributes over time, events can vary based on nuances in event arguments, such as time, location, and their participants, leading to increased complexity and ambiguity. For instance, Fig. 1 illustrates that the two event mentions of *"war"* should be differentiated and linked to distinct Wikipedia entries by recognizing their unique occurrence times and involved leaders, despite the similarity in event names and combatants.



In 1775, the conflict between the British East India Company and the Maratha Empire escalated into **war**. British troops under the command of Colonel Keating, left Surat on 15 March 1775, for Pune.

Label:
First Anglo-Maratha War

In 1803, the tensions between the British East India Company and a coalition of Maratha factions erupted into **war**. With the logistic assembly of his army complete, Wellesley gave the order to attack the nearest Maratha fort.

Label:
Second Anglo-Maratha War

Figure 1: An example of two distinct events with the same event mention but different event arguments.

Previous studies on event linking predominantly employed methods for entity linking. Yu et al. (2023a) and Pratapa et al. (2022) both reuse the framework from Wu et al. (2020), which first uses a bi-encoder to perform efficient retrieval, followed by precise re-ranking of top candidates using a cross-encoder. Yu et al. (2023a) further enhanced this approach by incorporating adjacent named entities into the query text, emphasizing the role of entities in event linking. However, not all adjacent entities are relevant to the event, and even the relevant ones can play different roles in an event.

Furthermore, given the limited entries KBs usually possess, it is always a practical challenge for linking models to deal with out-of-KB queries (Dong et al., 2023). This challenge is more acute in event linking due to the vast number of newly occurring events, of which only a fraction

[*] The authors contribute equally.

Findings of the Association for Computational Linguistics: ACL 2024, pages 12769–12781
August 11-16, 2024 ©2024 Association for Computational Linguistics

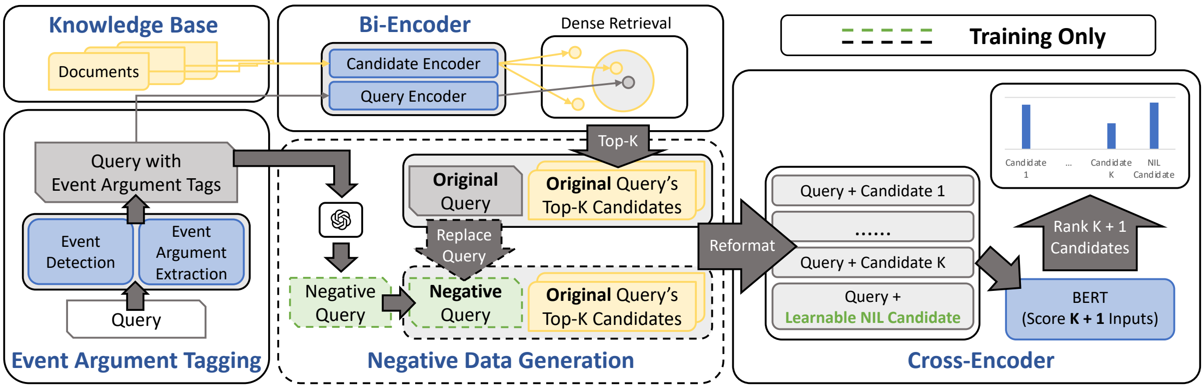


Figure 2: Given a text with an event to ground, our method extracts the event's attributes through event detection and argument extraction modules. The text, enriched with event argument tags, is then input into a Bi-Encoder to identify the top-k potential nodes. These candidates are further ranked by a Cross-Encoder, which also considers an additional *"NIL"* candidate in case of out-of-KB instances. To equip the Cross-Encoder to robustly predict *"NIL"* and real KB entries, we train it with additional synthetic data generated through our negative data creation process.

are recorded in KBs. However, this issue has often been overlooked in existing event linking research.

In this work, we enhance event linking systems by capitalizing on the role of event arguments in distinguishing events. We use established event extraction models (Huang et al., 2022; Hsu et al., 2023a) to capture the participants, time, and locations of the query event. As illustrated in Fig. 1, this argument data is crucial for differentiating events. To address out-of-KB challenges, we train models to predict *"out-of-KB"* labels using synthetic out-of-KB query data, which is created by manipulating the event arguments of existing queries. For example, our system will replace *British East India Company* and the *Maratha Empire* in Fig. 1 with alternative fictional combatant pairs to form the training data for *"out-of-KB"* prediction.

We apply our design to a model architecture akin to Yu et al. (2023a); Pratapa et al. (2022) and conduct experiments on the two event linking datasets introduced by Yu et al. (2023a). Our approach yields a 22% accuracy improvement over previous baselines for out-of-KB testing and an over 1% increase for in-KB testing. Additionally, by comparing various methods for generating synthetic out-of-KB examples, we demonstrate that our data synthesis approach successfully balances in-KB and out-of-KB usage for event linking. Our code can be found in https://github.com/PlusLabNLP/Argu_Event_Linking.

2 Related Work

Entity Linking, which associates entity mentions with KB entries, has been studied for years (Bunescu and Pasca, 2006; Mihalcea and Csomai, 2007; Gupta et al., 2017; Botha et al., 2020). Common approaches include using neural networks to represent queries and KB entries for discriminative prediction (Francis-Landau et al., 2016; Wu et al., 2020; Zhang et al., 2022) or using generation-based methods (Cao et al., 2021; Mrini et al., 2022; Xiao et al., 2023). While these techniques can be adapted for event linking, they are not tailored to incorporate the structured information within events, which, as we will demonstrate in §5, is vital for disambiguating events for grounding.

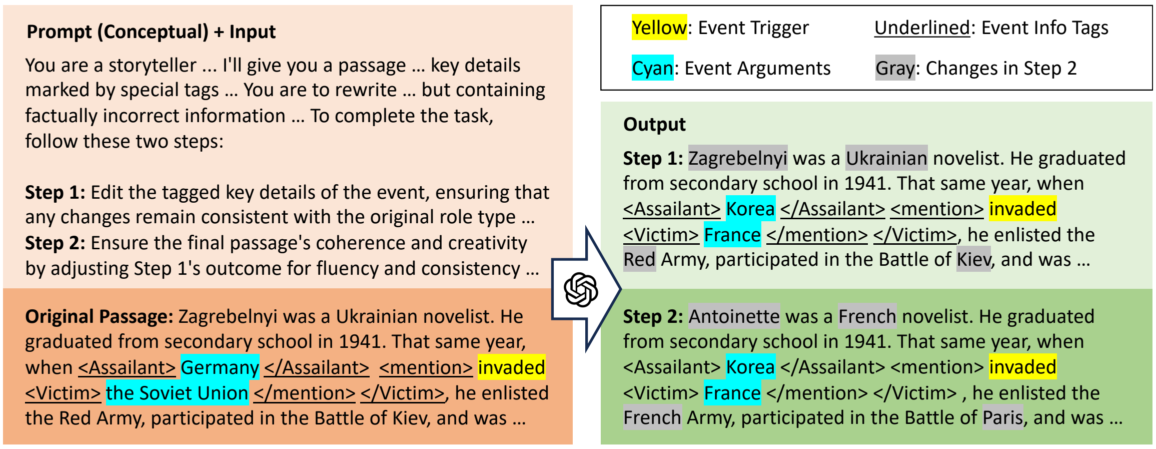
Event Linking are first introduced by Nothman et al. (2012). Recently, Yu et al. (2023a) and Pratapa et al. (2022) make efforts on introducing English and multilingual datasets on the task. However, their approaches overlook the influence of event arguments to the task and neglect the discussion of handling out-of-KB cases.

3 Problem Statement

Event linking connects a mention of an event in the text to relevant entries within a knowledge base. If no matching entry exists, models predict "NIL," as established in prior research (Zhu et al., 2023; Yu et al., 2023a). The event-linking model analyzes a sequence of tokens $\mathbf{x} = x_1, x_2, ..., x_{eve}, ..., x_l$, where x_{eve} signifies the target event mention. Its objective is to output y, which can be either an entry e_i from the knowledge base (size n), or "NIL" if no suitable match is found.

4 Method

We hereby introduce our event linking model with two key innovations, as illustrated in Fig. 2. First,



Prompt (Conceptual) + Input

You are a storyteller … I'll give you a passage … key details marked by special tags … You are to rewrite … but containing factually incorrect information … To complete the task, follow these two steps:

Step 1: Edit the tagged key details of the event, ensuring that any changes remain consistent with the original role type …
Step 2: Ensure the final passage's coherence and creativity by adjusting Step 1's outcome for fluency and consistency …

Original Passage: Zagrebelnyi was a Ukrainian novelist. He graduated from secondary school in 1941. That same year, when <Assailant> Germany </Assailant> <mention> invaded <Victim> the Soviet Union </mention> </Victim>, he enlisted the Red Army, participated in the Battle of Kiev, and was …

Yellow: Event Trigger Underlined: Event Info Tags
Cyan: Event Arguments Gray: Changes in Step 2

Output

Step 1: Zagrebelnyi was a Ukrainian novelist. He graduated from secondary school in 1941. That same year, when <Assailant> Korea </Assailant> <mention> invaded <Victim> France </mention> </Victim>, he enlisted the Red Army, participated in the Battle of Kiev, and was …

Step 2: Antoinette was a French novelist. He graduated from secondary school in 1941. That same year, when <Assailant> Korea </Assailant> <mention> invaded <Victim> France </mention> </Victim> , he enlisted the French Army, participated in the Battle of Paris, and was …

Figure 3: Illustration for our negative data generation processing using LLM.

to help the model distinguish event attribute details, we propose to first tag the event argument information of the input query (§4.1). Second, to improve the model's capability to deal with out-of-KB cases, we introduce a negative data generation method that synthesizes potential out-of-KB examples to train our model (§4.2). Finally, we train the model with these data changes (§4.3).

4.1 Event Argument Tagging

To make event linking models better capture event argument information, we use the UniST model (Huang et al., 2022) trained on the MAVEN dataset (Wang et al., 2020) to first identify the event types of query events. Given predicted event types, we extract event arguments using the TagPrime model (Hsu et al., 2023a) trained on the GENEVA dataset (Parekh et al., 2023). Text with event argument tags is used by our model. Take the passage in Fig. 3 as an example, *"Germany"* will be extracted as the *"Assailant"* of the invasion, *"the Soviet Union"* will be highlighted as the *"Victim"*. More relevant details about event trigger and argument extraction can be found in Appx. §B.

4.2 Negative Data Generation

Prior research on event linking largely overlooked out-of-KB issues, mainly due to the limited availability of diverse training data for such scenarios. To address this gap, we design a pipeline to generate synthetic training data, enhancing the ability of event linking systems to make robust predictions for both in-KB and out-of-KB queries.

Creating out-of-KB event queries is non-trivial because randomly altering the query text does not guarantee that the event falls outside the KB or at least stops referencing the original event. Directly altering the event mention word may lead to text that is incoherent or still references the same event, such as changing *"invaded"* to *"grew"* or *"attacked"* in Fig. 3.

We address the challenge by leveraging our observation that events differ when argument configurations change. To generate a data point, we first sample an in-KB query from the training set, along with its tagged event mention and arguments. We then instruct a large language model (LLM) to adjust this example through a two-step process: first modifying the tagged event arguments and then making edits to ensure coherence and fluency, as demonstrated in Fig. 3. [1]

To increase the likelihood of generating more realistic out-of-KB query cases, we instruct the LLM to create contexts that violates its own knowledge. However, there remains a possibility that the generated context may reference other events within the KB. To minimize this impact, in actual data usage, we treat our generated event query as a "negative" training data point when paired with top KB entries for the original sampled in-KB mention. Further details are provided in §4.3.

4.3 Model

We apply our proposed techniques to the same retrieve-and-rerank model architecture (Wu et al., 2020) used in prior works (Yu et al., 2023a).

The retrieve stage involves a bi-encoder model. A *candidate encoder* first encodes each entry in the KB into a dense space. A text query q with event information tags is then fed into the other encoder

[1] This two step generation is generated through a single prompt. We use GPT-3.5-Turbo (Ouyang et al., 2022) with 2-shot examples (Brown et al., 2020) to instruct the model. More details about the prompt are listed in Appx. §C.

(*query encoder*) and projected into the same dense space. Top-k candidates will be extracted by measuring the dot product similarities between q and every KB entry.

After obtaining top-k KB candidates $c_1, c_2, ..., c_k \quad \forall c_i \in \{e_1, e_2, ...e_n\}$, a cross-encoder is employed to encode every pair (q, c_i) to a score $S(q, c_i)$. The best candidate is selected by ranking the scores $c = \arg\max_{c_i} S(q, c_i)$.

To handle out-of-KB scenarios, prior work (Yu et al., 2023a), lacking out-of-KB training examples, generates the final output c_{final} by setting an *arbitrary* threshold θ:

$$c_{\text{final}} = \begin{cases} c, & \text{if } S(q, c) < \theta \\ \text{NIL}, & \text{otherwise.} \end{cases}$$

Unlike this approach, our method introduces *a learned "NIL" class* trained with our synthetic negative data. During the re-ranking phase, we expand the candidate pool to include $k + 1$ options, the extra one being a randomly initialized embedding that represents "NIL" :

$$c_{\text{final}} = \arg\max_{c_i} S(q, c_i)$$
$$, \text{where } i \in \{0, 1, ..., k\}, c_0 = \text{NIL}.$$

Our cross-encoder is trained to predict "NIL" when the input query q is replaced with the synthetic negative query we generated, illustrated as the "Negative Query" in Fig. 2.

5 Experiments

5.1 Experimental Settings

Datasets from the prior event linking work (Yu et al., 2023a) are used. The data is constructed by Yu et al. (2023a). The event KB used is the collection of Wikipedia pages with English titles. The datasets include the Wikipedia dataset, which contains training and in-domain testing data, and the New York Times (NYT) dataset, which contains out-of-domain and out-of-KB testing data. We introduce the details of the datasets below and list their statistics in Tab. 1.

- **The Wikipedia dataset** contains the training, validation, and test splits. The Wikipedia dataset is collected automatically from hyperlinks in Wikipedia. A hyperlink text is considered an event mention if the linked Wikipedia title has its mapped FIGER type (Ling and Weld, 2012), a fine-grained set of entity tags, being "Event." By construction, the Wikipedia dataset contains in-KB event mentions only.

Dataset	Train	Valid.	Test	
			In-KB	Out-of-KB
Wikipedia	66217	16650	19213	-
NYT	-	-	769	993

Table 1: Statistics of the two datasets.

- **The NYT dataset** is a smaller, manually annotated test set. 2,500 lead paragraphs are sampled from The New York Times Annotated Corpus and then annotated through Amazon Mechanical Turk. The dataset comes from real-life news articles and contains out-of-KB event mentions that is not covered by Wikipedia.

Evaluation Metrics. We follow Yu et al. (2023a) to evaluate models using **accuracy**.

Compared Methods:

- **BM25**: a term-based ranking algorithm for information retrieval.
- **GENRE**: a generation-based entity linking model retrieves entities by generating their unique names. We follow the analysis from (Yu et al., 2023a), who train GENRE on the event-linking training set and perform inference.
- **BLINK** (Wu et al., 2020): the retrieve-and-rerank model architecture introduced for entity linking. We adopt its code but train it on the event linking training set. BLINK processes the entire token sequence, enriched with special tokens to mark the beginning and end of the event mention, i.e., $x_1, x_2, ..., [M_s], x_{eve}, [M_e], ..., x_l$, and apply retrieve-and-rerank training and inference. Here, $[M_s]$ and $[M_e]$ are the special markers.
- **EveLINK** (Yu et al., 2023a): the current SOTA event linking model that also follows retrieve-and-rerank framework. It adopts BLINK but enhances the text query by adding local name entity information. Specifically, EveLink initially performs named entity recognition on $\mathbf{x}$, identifying named entities, $ne_1, ne_2, ...$ and their type $t_1, t_2, ...$ It then combines the token and entity sequences as input to models, i.e., $x_1, x_2, ..., [M_s], x_{eve}, [M_e], ..., x_l,$ $[SEP], [t_{1s}], ne_1, [t_{1e}], [t_{2s}], ne_2, [t_{2e}], ...,$ where $[SEP]$ is the BERT separator, and $[t_{1s}]$, $[t_{1e}]$, ... are the special tokens indicating the start and end of named entity types, respectively.
- **Our method** also employs the retrieve-and-rerank framework. Yet, our method differs from BLINK in that we incorporate event argument

Model	Wikipedia Test			NYT Test		
	All	Verb	Noun	All	Verb	Noun
BM25	9.72	13.08	6.36	3.69	3.18	5.19
GENRE [†]	76.04	71.76	80.32	-	-	-
BLINK	78.74	78.12	79.36	27.13	29.24	20.74
EveLink	79.00	78.07	79.93	32.03	34.34	25.13
Ours	**80.05**	**79.47**	**80.62**	**55.40**	**59.90**	**41.99**

Table 2: Accuracy (%) on both Wikipedia (in-domain, in-KB) and NYT (out-of-domain, out-of-KB) test sets. The best performance is highlighted in bold. [†]We report GENRE (Cao et al., 2021)'s numbers using the results from Yu et al. (2023a).

information of $\mathbf{x}$ and apply negative data generation to augment the training data for better support on "out-of-KB" cases. Specifically, our model's input format can be represented as $x_1, x_2, ..., [r_{1s}], x_{arg1}, [r_{1e}], ..., [M_s], x_{eve}, [M_e], ..., [r_{2s}], x_{arg2}, [r_{2e}], ...$, where x_{arg} denotes the event argument token, and $[r_{1s}]$, $[r_{1e}]$, ..., are special tokens indicating the corresponding argument roles.

All reported results in this section are the average of three random runs. Appx. §D covers all implementation details of the compared methods.

5.2 Main Results

Tab. 2 presents our main results, categorized by the type of event mention (All/Verb/Noun). In the in-domain Wikipedia evaluation, our approach surpasses all baseline methods across all categories. For the out-of-domain, out-of-KB evaluation using the NYT dataset, our method demonstrates its robustness with an over 20% absolute improvement.

5.3 Analysis

In this section, we present studies to verify our two innovations. The exploration of the possibilities of using LLMs in event-linking modeling can be found in Appx. §A.

5.3.1 Bi-Encoder Results

Bi-encoder results are shown in Tab. 3. Directly analyzing the bi-encoder performance allows us to assess the impact of integrating event argument data into the text. Our approach surpasses all baseline methods, showing greater enhancements in the harder cases as the number of candidates decreases.

Model	Wikipedia (in-KB) Test Set Recall					
	R@1	R@2	R@3	R@5	R@10	R@20
BM25	9.72	16.64	20.58	25.48	31.77	38.10
BLINK	54.85	68.14	74.27	80.36	86.22	90.55
EveLink	55.72	67.22	74.74	80.62	86.51	90.91
Ours	**57.28**	**70.14**	**76.10**	**81.69**	**87.40**	**91.34**

Table 3: Bi-encoder recall (%) on the Wikipedia test set. "R@1" stands for recall at 1, and so on. See Appx. §E for more recall values.

Model	Wiki.	NYT
BLINK (no negative data usage)	78.74	27.13
w/ Non-argument-aware method	79.09	54.08
w/ KB Pruning	76.72	**55.85**
w/ Argument-aware method (Ours)	**79.22**	55.18

Table 4: Analysis of alternative negative data generation methods. The best and the second-best are bolded and underlined, respectively. Our method shows the best balance between in-KB and out-of-KB cases.

5.3.2 Effectiveness of Negative Data Generation

We benchmark our approach against two alternative methods for generating negative data to train the cross-encoder: **(1) Non-argument-aware Data Generation**, which also employs GPT-3.5-Turbo but does not incorporate event information into the prompts, as detailed in Appx. §C; **(2) KB Pruning**, a strategy introduced by Dong et al. (2023) in the entity linking field. This method creates negative samples by randomly eliminating 10% of KB entries and marking the associated training data as negative examples external to the KB. Tab. 4 shows the comparison. While KB Pruning ensures high-quality negative examples outside the KB, it negatively affects performance on in-KB tests. In contrast, our method, designed to emphasize event information, effectively balances the use of in-KB and out-of-KB cases.

6 Conclusion

In this work, we introduce an argument-aware method designed to improve event linking models. This approach aids in disambiguating events and generating out-of-KB training examples. Experimental results demonstrate that our method enhances the accuracy for both in-KB and out-of-KB queries. Our findings reveal that the system, trained on flattened data, struggles to process structured textual information effectively. Therefore, implementing our guidance about event arguments can improve its understanding of structured events.

Acknowledgements

We gratefully acknowledge the insightful feedback provided by both the anonymous reviewers and the UCLA PLUS Lab and UCLA-NLP group members on earlier versions of this paper. This research was funded by a Cisco Sponsored Research Award.

Limitation

Although our incorporation of event argument information is shown to be highly effective, it does introduce additional pipelines to extract and tag event information, bringing extra cost to model training and inference. Additionally, since we demonstrated that tagging event information in the query helps improve performance, it would be interesting to explore the possibility of also leveraging structured event information in the KB entries as well. We leave this investigation for future work.

Broader Consideration

We employ LLMs to generate training data for our model. Consequently, the model may inherit biases from the LLM, potentially leading to ethical concerns. Despite the low likelihood and our use of the data for negative examples, we recommend a thorough evaluation of these potential issues prior to deploying the model in real-world applications.

References

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity linking in 100 languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Razvan C. Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Hang Dong, Jiaoyan Chen, Yuan He, Yinan Liu, and Ian Horrocks. 2023. Reveal the unknown: Out-of-knowledge-base mention discovery with entity linking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*.

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*.

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)*.

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), ACL 2023, Toronto, Canada, July 9-14, 2023*.

James Y. Huang, Bangzheng Li, Jiashu Xu, and Muhao Chen. 2022. Unified semantic typing with meaningful label inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Premkumar Natarajan, Kai-Wei

Chang, Nanyun Peng, and Heng Ji. 2023. A reevaluation of event extraction: Past, present, and future challenges. *CoRR*, abs/2311.09562.

Giulio Jacucci, Pedram Daee, Tung Thanh Vuong, Salvatore Andolina, Khalil Klouche, Mats Sjöberg, Tuukka Ruotsalo, and Samuel Kaski. 2021. Entity recommendation for everyday digital tasks. *ACM Trans. Comput. Hum. Interact.*

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*.

Khalil Mrini, Shaoliang Nie, Jiatao Gu, Sinong Wang, Maziar Sanjabi, and Hamed Firooz. 2022. Detection, disambiguation, re-ranking: Autoregressive entity linking as a multi-task problem. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*.

Joel Nothman, Matthew Honnibal, Ben Hachey, and James R. Curran. 2012. Event linking: Grounding event reference in a news archive. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*.

Jiefu Ou, Adithya Pratapa, Rishubh Gupta, and Teruko Mitamura. 2023. Hierarchical event grounding. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

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)*.

Hao Peng, Xiaozhi Wang, Feng Yao, Zimu Wang, Chuzhao Zhu, Kaisheng Zeng, Lei Hou, and Juanzi Li. 2023. 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, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *CoRR*, abs/2309.15088.

Adithya Pratapa, Rishubh Gupta, and Teruko Mitamura. 2022. Multilingual event linking to Wikidata. In *Proceedings of the Workshop on Multilingual Information Access (MIA)*.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zilin Xiao, Ming Gong, Jie Wu, Xingyao Zhang, Linjun Shou, and Daxin Jiang. 2023. Instructed language models with retrievers are powerful entity linkers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*.

Xiaodong Yu, Wenpeng Yin, Nitish Gupta, and Dan Roth. 2023a. Event linking: Grounding event mentions to wikipedia. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*.

Xiaodong Yu, Ben Zhou, and Dan Roth. 2023b. Building interpretable and reliable open information retriever for new domains overnight. *CoRR*, abs/2308.04756.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2022. Entqa: Entity linking as question answering. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Fangwei Zhu, Jifan Yu, Hailong Jin, Lei Hou, Juanzi Li, and Zhifang Sui. 2023. Learn to not link: Exploring NIL prediction in entity linking. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*.

A Experiments with Large Language Models as Ranking Model

To the best of our knowledge, current event linking systems lack the integration of Large Language Models (LLMs). Inspired by recent information retrieval techniques leveraging LLMs (Pradeep et al., 2023), we explore their effectiveness by incorporating an instructed LLM for candidate re-ranking within the existing event linking pipeline, replacing the traditional cross-encoder.

We establish a baseline using GPT-3.5-Turbo and compare our proposed method with this LLM baseline [2]. To ensure a fair comparison, we utilize the same bi-encoder system and sample a subset of 1,000 test examples.

The results presented in Table 5 demonstrate that the method solely employing LLM for re-ranking significantly underperforms our approach. This finding highlights the key advantage of our method, which leverages LLMs for generating negative data, leading to superior performance.

Model	Wikipedia Test			NYT Test		
	All	Verb	Noun	All	Verb	Noun
LLM-reranked	44.64	45.61	43.53	28.56	29.51	25.70
Ours	**79.88**	**77.76**	**82.33**	**57.01**	**61.28**	**44.18**

Table 5: Comparison with the LLM-reranked baseline. Due to the budget constraints, the experiment is conducted on a subset of the whole dataset.

B Event Extraction Details

Our system employs a two-step approach for comprehensive event extraction from text. First, we perform event typing to categorize the event described in the input text. We leverage UniST (Huang et al., 2022) to achieve this, trained on the MAVEN dataset, which offers broad coverage with 168 event types. This ensures our system can handle a wide range of potential input events.

Next, upon identifying the event type, we proceed with event argument extraction. We utilize TagPrime (Hsu et al., 2023a), the state-of-the-art model in this domain. To maintain consistency with MAVEN's ontology, we employ the GENEVA dataset, which aligns closely with MAVEN while encompassing 115 event types due to minor modifications. Consequently, our system offers comprehensive support for 115 event types, encompassing 220 distinct argument roles, effectively covering diverse event types within queries on our experimental dataset.

While alternative event extraction models exist, such as DEGREE (Hsu et al., 2022), UniIE (Lu et al., 2022), AMPERE (Hsu et al., 2023b), and PAIE (Ma et al., 2022), each trained on various datasets, our approach stands out as the most comprehensive event extraction pipeline to date. It captures the vast majority of event types, providing a robust solution for diverse user needs. For a more in-depth exploration of alternative methodologies, we recommend referring to the works of Peng et al. (2023); Huang et al. (2023).

C Data Generation Details

Our negative data generation begins by sampling from our tagged training and validation sets. To ensure the quality of the generated data, we filter out examples where the labeled event mentions are proper nouns or numeric values. Additionally, to better leverage our observations, we exclude examples with fewer than two tagged event arguments.

Following this filtering step, we employ a two-stage process to generate negative data, as detailed in the prompt provided in Fig. 4.

Due to the high cost of GPT-3.5-Turbo, we only generated a final set of 6,600 examples from the training set and 1,600 examples from the validation set to train and develop our method.

We employed the same data generation method for the non-argument-aware baseline, but without tagging event arguments. The prompt was adjusted to reflect the absence of this information (see Fig. 5 for the specific prompt used). Notably, the text content of the few-shot examples remained identical in both cases; the only distinction lies in the presence of event argument tags.

D Implementation Details

Model Training and Inference Pipeline Our system follows a two-stage training approach. First, we train a bi-encoder to encode queries and candidate event mentions. We then use the queries and their top 10 candidates to train the cross-encoder. During inference, we also first retrieve the top 10 candidates using the trained bi-encoder. Then, we re-rank the retrieved candidates using the trained cross-encoder and select the top 1 candidate to compare with the ground truth. To ensure a fair comparison, all baselines and our methods follow

[2] Implementation details can be found in Appx. §D.

```
You are a storyteller, and you can assist me in crafting a narrative based on a
given passage. I’ll give you a passage marking an event and its key details using
specific tags. The event is marked with "<mention> </mention>", while the related
details and the corresponding roles are identified with tags like "<role> </role>",
such as <Victim> Mark </Victim>, indicating that Mark is the event’s "Victim". You
are to rewrite the passage with similar length and structure but containing false
information by changing the key details. Remember that I want a passage that is
factually incorrect.
To complete the task, follow these two steps:
Step 1: Edit the tagged key details of the event, ensuring that any changes remain
consistent with the original role type.
Step 2: Ensure the final generated passage’s coherence and creativity by adjusting
Step 1’s outcome for fluency and consistency. This may include modifying unaltered
parts to enhance logic and flow.

Before each step, state your plans a little bit. Additionally, don’t truncate the
original passage or alter any escaped characters. Also, don’t remove any argument
role tags in the form of "<role> </role>" or event mention tags in the form of "<
mention> </mention>". Present your results as: ’Plan 1: {{outline your changes for
Step 1}}\nFollowing Plan 1, we can generate this passage after Step 1: {{passage
after Step 1}}\nPlan 2: {{outline your changes for Step 2}}\nFollowing Plan 2, we
can generate this passage after Step 2: {{passage after Step 2}}

You can refer to the first two examples we provided and complete the third one on
your own.

Example 1:
{Example 1}

Example 2:
{Example 2}

Example 3:
Passage: {}

Additional information we have for the Passage: This "{event mention text span}"
event is of the type "{event type}".
```

Figure 4: Prompt for our argument-aware data generation.

```
You are a storyteller, and you can assist me in crafting a narrative based on a
given passage. I’ll give you a passage containing a reference to an event. An event
is an occurrence of something that happens in a certain time/place involving some
participants. In the given passage, The textual expression that refers to the event
is called the "mention" of the event. The event mention is marked with surrounding
"<mention> </mention>" tags. You are to rewrite the passage with similar length and
structure but containing factually incorrect information. Remember that I want a
passage that is factually incorrect.

Do not truncate the original passage or alter any escaped characters. Also, do not
remove the event mention tags in the form of "<mention> </mention>" from the passage
. Present your output as: ’New passage: {{new passage}}’

You can refer to the first two examples we provided and complete the third one on
your own.

Example 1:
{Example 1}

Example 2:
{Example 2}

Example 3:
Passage:

New passage:
```

Figure 5: Prompt for non-argument-aware data generation baseline.

```
I would like you to help me with a document re-ranking task. I will give you a short
 passage containing an event. I will also give you a list of 10 documents, each with
 a title and a short description. You task is to rank the given 10 documents in
decreasing order of relevance to the event that the short passage mentions. Do not
remove any documents. Do not include any documents that are not provided. In your
answer, only provide the document titles in the original format.

Input format:
Document 1: <title of document 1>
<short description of document 1>
Document 2: <title of document 2>
<short description of document 2>
...
Document 10: <title of document 10>
<short description of document 10>
Short passage containing an event: <short passage containing an event>

Answer format:
Document d1: <title of document d1 (most relevant document)>
Document d2: <title of document d2 (second most relevant document)>
...
Document d10: <title of document d10 (least relevant document)>

Now, here is the actual input.
{actual input}
```

Figure 6: Prompt for the LLM baseline on the Wikipedia dataset (in-KB).

```
I would like you to help me with a document re-ranking task. I will give you a short
 passage containing an event. I will also give you a list of 10 documents, each with
 a title and a short description. You task is to rank the given 10 documents in
decreasing order of relevance to the event that the short passage mentions. However,
 it is possible that none of the 10 given documents describes the event in the
passage. If you think that the event in the passage is not described by any of the
10 given documents, you should label the passage with a special "NIL" label. Do not
remove any documents. Do not include any documents that are not provided. In your
answer, only provide the document titles in the original format.

Input format:
Document 1: <title of document 1>
<short description of document 1>
Document 2: <title of document 2>
<short description of document 2>
...
Document 10: <title of document 10>
<short description of document 10>
Short passage containing an event: <short passage containing an event>

Answer format (If 1 or more documents describe the event in the passage):
Document d1: <title of document d1 (most relevant document)>
Document d2: <title of document d2 (second most relevant document)>
...
Document d10: <title of document d10 (least relevant document)>

Answer format (If none of the documents describes the event in the passage, just
output this sentence below):
The passage should be labeled as NIL.
Now, here is the actual input.
{actual input}
```

Figure 7: Prompt for the LLM baseline on the NYT dataset (out-of-KB).

this recipe.

Training is conducted on Nvidia A100 40G GPUs. The bi-encoder typically requires approximately 6 hours to train, while the cross-encoder takes around 30 hours.

Model Architectures Both the bi-encoder and cross-encoder leverage the pre-trained BERT-base-uncased model. For the bi-encoder, we set a maximum query and candidate length of 300 tokens. The training utilizes a learning rate of $1e-5$, a batch size of 48, and runs for 15 epochs to optimize GPU memory usage.

The cross-encoder also employs BERT-base-uncased, with a maximum query and candidate length of 256 tokens. Here, the training process utilizes a learning rate of $2e-5$, a batch size of 6, and runs for 20 epochs.

Baseline Configurations For the threshold θ used for baseline NIL prediction (see §4.3), we follow the description in (Yu et al., 2023a)'s paper and set it as 0.5 after normalization.

For the BM25 baseline, we use a query (context) window size of 16, which follows the practice from Pratapa et al. (2022).

For the analysis of negative data generation methods, details on data generation are covered Appx. §C. Additionally, the KB pruning baseline is implemented by randomly pruning 10% of the unique labels present in the training set. We then label the corresponding samples (5984 samples) as out-of-KB.

For the LLM baseline, on the Wikipedia test set, the model is asked to re-rank the top-k documents given the query. On the NYT test set, the model is given an additional option to simply label the query as "NIL" if none of the documents describes the event in the query. The two prompts for the Wikipedia and NYT test sets are shown in Fig. 6 and Fig. 7, respectively.

E Full Bi-encoder Evaluation Results

We present the full bi-encoder evaluation results, which include more recall values, in Tab. 6.

Model	Wikipedia (in-KB) Test Set Recall								
	R@1	R@2	R@3	R@4	R@5	R@8	R@10	R@15	R@20
BM25	9.72	16.64	20.58	23.16	25.48	29.78	31.77	35.58	38.10
BLINK (Wu et al., 2020)	54.85	68.14	74.27	77.95	80.36	84.49	86.22	89.00	90.55
EveLink (Yu et al., 2023a)	55.72	67.22	74.74	78.27	80.62	84.83	86.51	89.21	90.91
Ours	**57.28**	**70.14**	**76.10**	**79.49**	**81.69**	**85.82**	**87.40**	**89.80**	**91.34**

Table 6: Different recall values on the in-domain, in-KB evaluation for the bi-encoder on the Wikipedia test set. The best performance is highlighted in bold. "R@1" stands for recall at 1, and so on.