How Good is the Model in Model-in-the-loop Event Coreference Resolution Annotation?

Shafiuddin Rehan Ahmed[1] **Abhijnan Nath**[2] **Michael Regan**[3]
Adam Pollins[1] **Nikhil Krishnaswamy**[2] **James H. Martin**[1]
[1]University of Colorado, Boulder, CO, USA [3]University of Washington, Seattle, WA, USA
[2]Colorado State University, Fort Collins, CO, USA
{shah7567,james.martin,adpo0729}@colorado.edu
{abhijnan.nath,nkrishna}@colostate.edu, mregan@cs.washington.edu

Abstract

Annotating cross-document event coreference links is a time-consuming and cognitively demanding task that can compromise annotation quality and efficiency. To address this, we propose a model-in-the-loop annotation approach for event coreference resolution, where a machine learning model suggests likely corefering event pairs only. We evaluate the effectiveness of this approach by first simulating the annotation process and then, using a novel annotator-centric Recall-Annotation effort trade-off metric, we compare the results of various underlying models and datasets. We finally present a method for obtaining 97% recall while substantially reducing the workload required by a fully manual annotation process.

1 Introduction

Event Coreference Resolution (ECR) is the task of identifying mentions of the same event either within or across documents. Consider the following excerpts from three related documents:

e_1: 55 year old star will $replace_{m_1}$ Matt Smith, who announced in June that he was leaving the sci-fi show.

e_2: Matt Smith, 26, will make his debut in 2010, $replacing_{m_2}$ David Tennant, who leaves at the end of this year.

e_3: Peter Capaldi $takes\ over_{m_3}$ Doctor Who ... Peter Capaldi $stepped\ into_{m_4}$ Matt Smith's soon to be vacant Doctor Who shoes.

e_1, e_2, and e_3 are example sentences from three documents where the event mentions are highlighted and sub-scripted by their respective identifiers (m_1 through m_4). The task of ECR is to automatically form the two clusters $\{m_1, m_3, m_4\}$, and $\{m_2\}$. We refer to any pair between the mentions of a cluster, e.g., (m_1, m_3) as an ECR link. Any pair formed across two clusters, e.g., (m_1, m_2) is referred to as non-ECR link.

Annotating ECR links can be challenging due to the large volume of mention pairs that must be compared. The annotating task becomes increasingly time-consuming as the number of events in the corpus increases. As a result, this task requires a lot of mental effort from the annotator and can lead to poor quality annotations (Song et al., 2018; Wright-Bettner et al., 2019). Indeed, an annotator has to examine multiple documents simultaneously often relying on memory to identify all the links which can be an error-prone process.

To reduce the cognitive burden of annotating ECR links, annotation tools can provide integrated model-in-the-loop for sampling likely coreferent mention pairs (Pianta et al., 2008; Yimam et al., 2014; Klie et al., 2018). These systems typically store a knowledge base (KB) of annotated documents and then use this KB to suggest relevant candidates. The annotator can then inspect the candidates and choose a coreferent event if present.

The model's querying and ranking operations are typically driven by machine learning (ML) systems that are trained either actively (Pianta et al., 2008; Klie et al., 2018; Bornstein et al., 2020; Yuan et al., 2022) or by using batches of annotations (Yimam et al., 2014). While there have been advances in suggestion-based annotations, there is little to no work in evaluating the effectiveness of these systems, particularly in the use case of ECR. Specifically, both the overall coverage, or recall, of the annotation process as well as the degree of annotator effort needed depend on the performance of the model. In order to address this shortcoming, we offer the following contributions:

1. We introduce a method of model-in-the-loop annotations for ECR[1].
2. We compare three existing methods for ECR (differing widely in their computational costs), by adapting them as the underlying ML mod-

[1]repo: github.com/ahmeshaf/model_in_coref

els governing the annotations.

3. We introduce a novel methodology for assessing the workflow by simulating the annotations and then evaluating an annotator-centric Recall-Annotation effort tradeoff.

2 Related Work

Previous work for ECR is largely based on modeling the probability of coreference between mention pairs. These models are built on supervised classifiers trained using features extracted from the pairs. Most recent work uses a transformer-based language model (LM) like BERT (Devlin et al., 2018; Liu et al., 2019) to generate joint representations of mention pairs, a method known as cross-encoding. The cross-encoder is fine-tuned using a coreference scoring objective (Barhom et al., 2019; Cattan et al., 2020; Meged et al., 2020; Zeng et al., 2020; Yu et al., 2020; Caciularu et al., 2021). These methods use scores generated from the scorer to then agglomeratively cluster coreferent events.

Over the years, a number of metrics have been proposed to evaluate ECR (Vilain et al., 1995; Bagga and Baldwin, 1998; Luo, 2005; Recasens and Hovy, 2011; Luo et al., 2014; Pradhan et al., 2014). An ECR system is evaluated using these metrics to determine how effectively it can find event clusters (recall) and how cleanly separated the clusters are (precision). From the perspective of annotation, it may only be necessary to focus on the system's recall or its effectiveness in finding ECR links. However, an annotator might still want to know how much effort is required to identify these links in a corpus to estimate their budget. In the remainder of the paper, we attempt to answer this question by first quantifying annotation effort and analyzing its relation with recall of the system.

We use the Event Coreference Bank Plus (ECB+; Cybulska and Vossen (2014)) and the Gun Violence Corpus (GVC; Vossen et al. (2018)) for our experiments. The ECB+ is a common choice for assessing ECR, as well as the experimental setup of Cybulska and Vossen (2015) and gold topic clustering of documents and gold mention annotations for both training and testing[2]. On the other hand, the GVC offers a more challenging set of exclusively event-specific coreference decisions that require resolving gun violence-related events.

[2]The ECB+ test set has 1,780 event mentions with 5K ECR links among 100K pairwise mentions, while the GVC test set has 1,008 mentions with 2K ECR links in 20K pairs. Full statistics in Table 1 in Appendix A

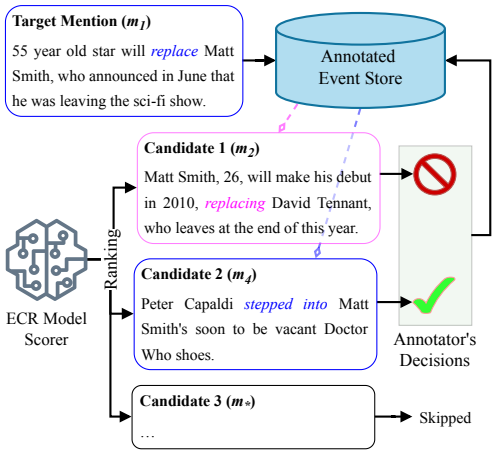

Figure 1: For the target mention (m_1), the Annotated Event Cluster store presents three potential coreferent candidates (m_2, m_4 and m_*). The ranking module (an ECR scorer) then ranks them based on their semantic similarity to m_1. The annotator reviews each candidate one-at-a-time and makes decisions on coreference. m_* is skipped after finding m_4 as coreferent. The cluster store is then updated based on these decisions.

3 Annotation Methodology

We implement an iterative model-in-the-loop methodology[3] for annotating ECR links in a corpus containing annotated event triggers. This approach has two main components - (1) the storage and retrieval of annotated event clusters, which are then compared with each new target event, and (2), an ML model that ranks and prunes the sampled candidate clusters by evaluating their semantic similarity to the target mention.

As illustrated in Figure 1, our annotation workflow queries the Annotated Event Store for the target event (m_1), retrieving three potential coreferring candidates (m_2, m_*, and m_4). The ranking module then evaluates these candidates based on their lexical and semantic similarities to m_1. The annotator then compares each candidate to the target and determines if they are coreferent. Upon finding a coreferent candidate, the target is merged into the coreferent cluster, and any remaining option(s) (m_*) are skipped.

3.1 Ranking

We investigate three separate methods to drive the ranking of candidates distinguished by their computational cost. We use these methods to generate the average pair-wise coreference scores between mentions of the candidate and target events, then

[3]Utilizing the prodi.gy annotation tool. See Appendix D

use these scores to rank candidates. We use a single RTX 3090 (24 GB) for running our experiments.

Cross-encoder (CDLM): In this method, we use the fine-tuned cross-encoder ECR system of Caciularu et al. (2021) to generate pairwise mention scores[4]. Their state of the art system uses a modified Longformer (Beltagy et al., 2020) as the underlying LM to generate document-level representations of the mention pairs (detailed in §B.1). More specifically, we generate a unified representation (Eq. 1) of the mention pair (m_i, m_j) by concatenating the pooled output of the transformer ($\mathrm{E_{CLS}}$), the outputs of the individual event triggers ($\mathrm{E}_{m_i}, \mathrm{E}_{m_j}$), and their element-wise product. Thereafter, pairwise-wise scores are generated for each mention-pair after passing the above representations through a Multi-Layer Perceptron (mlp) (Eq. 2) that was trained using the gold-standard labels for supervision.

$$\mathtt{LF}(m_i, m_j) = \left\langle \mathrm{E_{CLS}}, \mathrm{E}_{m_i}, \mathrm{E}_{m_j}, \mathrm{E}_{m_i} \odot \mathrm{E}_{m_j} \right\rangle \quad (1)$$

$$\mathtt{CDLM}(m_i, m_j) = \mathtt{mlp}(\mathtt{LF}(m_i, m_j)) \quad (2)$$

BERTScore (BERT): (Zhang et al., 2019) BERTScore (BS) is a NLP metric that measures pairwise text similarity by exploiting pretrained BERT models. It calculates cosine similarity of token embeddings with inverse document frequency weights to rate token importance and aggregates them into precision, recall, and F1 scores. This method emphasizes semantically significant tokens, resulting in a more accurate similarity score (details in §B.2).

$$\mathrm{S_{bert}}(m) = \langle t_m,\ [\mathrm{SEP}],\ S_m \rangle \quad (3)$$

$$\begin{aligned}\mathtt{BERT}(m_i, m_j) = \lambda\ &\mathtt{BS}(t_{m_i}, t_{m_j}) \\ &+ (1-\lambda)\ \mathtt{BS}(\mathrm{S_{bert}}(m_i), \mathrm{S_{bert}}(m_j))\end{aligned} \quad (4)$$

To calculate the BERTScore between the mentions, we first construct a combined sentence ($\mathrm{S_{bert}}(m)$; Shi and Lin (2019)) for a mention (m) by concatenating the mention text (t_m) and its corresponding sentence (S_m), as depicted in Equation 3. Subsequently, we compute the BS for each mention pair using $\mathrm{S_{bert}}(m)$ and t_m separately, then extract the F1 from each. We then take the weighted average of the two scores as shown in Equation 4 as our ranking metric. This process, carried out using the distilbert − base − uncased (Sanh et al., 2019) model, requires approximately seven seconds to complete on each test set.

Lemma Similarity (Lemma): The lemma[5] similarity method emulates the annotation process carried out by human annotators when determining coreference based on keyword comparisons between two mentions. To estimate this similarity, we compute the token overlap (Jaccard similarity; JS) between the triggers and sentences containing the respective mentions and take a weighted average of the two similarities (like Eq 4) as shown in Eq 5[6].

$$\begin{aligned}\mathtt{Lemma}(m_i,m_j) = \lambda\ &\mathtt{JS}(t_{m_i}, t_{m_j}) \\ &+ (1-\lambda)\ \mathtt{JS}(S_{m_i}, S_{m_j})\end{aligned} \quad (5)$$

No Ranking (Random): For our baseline approach, we employ a method that directly picks the candidate-mention pairs through random sampling and without ranking, providing a reference point for evaluating the effectiveness of the above three ranking techniques.

3.2 Pruning

To control the comparisons between candidate and target events, we restrict our selection to the top-k ranked candidates. To refine our analysis, we employ non-integer k values, allowing for the inclusion of an additional candidate with a probability equal to the decimal part of k. We vary the values of k from 2 to 20 on increments of 0.5 and then investigate its relation to recall and effort in §4.

3.3 Simulation

To evaluate the ranking methods, we conduct annotation simulations on the events in the ECB+ and GVC development and test sets. These simulations follow the same annotation methodology of retrieving and ranking candidate events for each target but utilize ground-truth for clustering. By executing simulations on different ranking methods and analyzing their performance, we effectively isolate and assess each approach.

4 Evaluation Methodology

We evaluate the performance of the model-in-the-loop annotation with the ranking methods through simulation on two aspects: (1) how well it finds the coreferent links, and (2) how much effort it would take to annotate the links using the ranking method.

[4]This method is compute-intensive since the transformer's encoding process scales quadratically with the number of mentions. Using the trained weights, running inference on the two test sets for our experiments takes approximately forty minutes to calculate the similarities of all the mention pairs. The weights are provided by Caciularu et al. (2021) here.

[5]We use spaCy 3.4 en_core_web_md lemmatizer

[6]λ is a hyper-parameter to control the weightage of the trigger and sentence similarities in Equations 4 and 5, which we tune using the development set. See Appendix C.

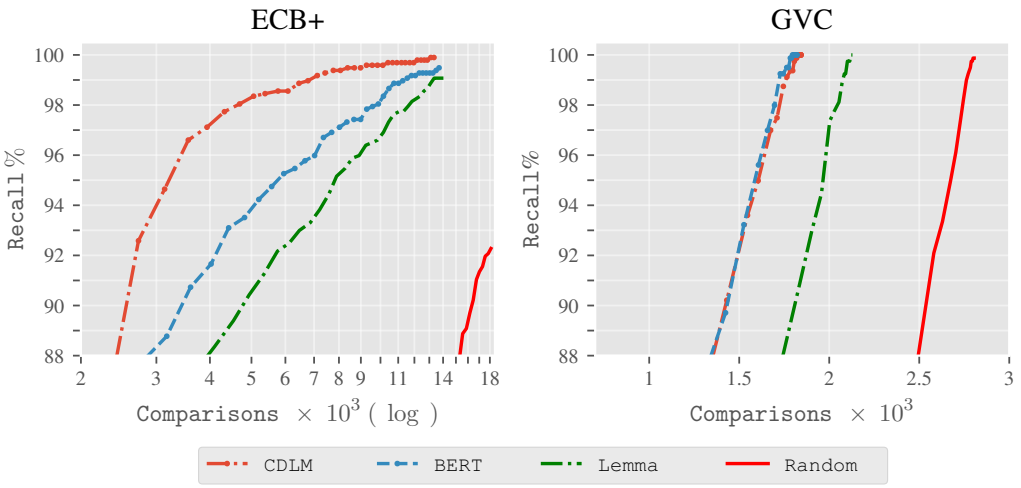

Figure 2: `Recall` and `Comparisons` achieved upon varying the k for each ranking method in the ECR annotation simulation. The three methods result in significantly fewer comparisons than the no-ranking `Random` baseline.

4.1 Recall-Annotation Effort Tradeoff

`Recall`: The recall metric evaluates the percentage of ECR links that are correctly identified by the suggestion model. It is calculated as the ratio of the number of times the true coreferent candidate is among the suggested candidates. The recall error is introduced when the coreferent candidate is erroneously removed based on the top-k value[7].

`Comparisons`: A unit effort represents the comparison between a candidate and target mentions that an annotator would have to make in the annotation process. We count the sampled candidates for each target and stop counting when the coreferent candidate is found. For example, the number of comparisons for the target m_1, in Figure 1, is 2 (m_2 and m_4). We count this number for each target event and present the sum as `Comparisons`.

4.2 Analysis and Discussion

We present an analysis of the various ranking methods employed in our study, highlighting the performance and viability of each approach. We employ the ranking methods on the test sets of ECB+ and GVC. Then, estimate the `Recall` and `Comparisons` measures for different k values, and collate them into the plots as shown in Figure 2.

Performance Comparison: The performance improvement of `CDLM` over `BERT` and `BERT` over `Lemma` can be quantified by examining the graph for the ECB+ and GVC datasets. For example, when targeting a 95% recall for the ECB+ corpus, `CDLM` provides an almost 100 percent improvement over `BERT` reducing the number of comparisons to almost half of the latter. However, both `CDLM` and `BERT` outperform `Lemma` by a significant margin while being drastically better than the `Random` baseline (See Fig. 2). Interestingly, for GVC, the performance gap between `CDLM` and `BERT` is quite close, both needing at least three-fourths as many comparisons as the `Lemma` and crucially outperforming the `Random` baseline. `CDLM`'s inconsistent performance on GVC suggests that a corpus-fine-tuned model such as itself is more effective when applied to a dataset similar to the one it was trained on.

Efficiency and Generalizability of `BERT`: `BERT` offers a compelling advantage in terms of efficiency, as it can be run on low-compute settings. Moreover, `BERT` exhibits greater generalizability out-of-the-box when comparing its performance on both the ECB+ and GVC datasets. This makes it an attractive option for ECR annotation task especially when compute resources are limited or when working with diverse corpora.

5 Conclusion

We introduced a model-in-the-loop annotation method for annotating ECR links. We compared three ranking models through a novel evaluation methodology that answers key questions regarding the quality of the model in the annotation loop (namely, recall and effort). Overall, our analysis demonstrates the viability of the models, with `CDLM` exhibiting the best performance on the ECB+ dataset, followed by `BERT` and `Lemma`. The choice of ranking method depends on the specific use case, dataset, and resource constraints, but all three methods offer valuable solutions for different scenarios.

[7]Note that recall is always 100% if no candidates are ever pruned.

Limitations

It is important to note that the approaches presented in this paper have several constraints. Firstly, the methods presented are restricted to English language only, as `Lemma` necessitates a lemmatizer and, `BERT` and `CDLM` rely on models trained exclusively on English corpora. Secondly, the utilization of the `CDLM` model demands at least a single GPU, posing potential accessibility issues. Thirdly, ECR annotation is susceptible to errors and severe disagreements amongst annotators, which could entail multiple iterations before achieving a gold-standard quality. Lastly, the generated corpora may be biased to the model used during the annotation process, particularly for smaller values of k.

Ethics Statement

We use publicly-available datasets, meaning any bias or offensive content in those datasets risks being reflected in our results. By its nature, the Gun Violence Corpus contains violent content that may be troubling for some.

Acknowledgements

We would like to express our sincere gratitude to the anonymous reviewers whose insightful comments and constructive feedback helped to greatly improve the quality of this paper. We gratefully acknowledge the support of U.S. Defense Advanced Research Projects Agency (DARPA) FA8750-18-2-0016-AIDA – RAMFIS: Representations of vectors and Abstract Meanings For Information Synthesis. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the U.S. government. We would also like to thank ExplosionAI GmbH for partly funding this work. Finally, we extend our thanks to the BoulderNLP group and the SIGNAL Lab at Colorado State for their valuable input and collaboration throughout the development of this work.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189, Florence, Italy. Association for Computational Linguistics.

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

Ari Bornstein, Arie Cattan, and Ido Dagan. 2020. CoRefi: A crowd sourcing suite for coreference annotation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 205–215, Online. Association for Computational Linguistics.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew Peters, Arie Cattan, and Ido Dagan. 2021. CDLM: Cross-document language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2648–2662, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2020. Streamlining cross-document coreference resolution: Evaluation and modeling.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 4545–4552. European Language Resources Association (ELRA).

Agata Cybulska and Piek Vossen. 2015. Translating granularity of event slots into features for event coreference resolution. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 1–10, Denver, Colorado. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. 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.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Conference on*

Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, page 25–32, USA. Association for Computational Linguistics.

Xiaoqiang Luo, Sameer Pradhan, Marta Recasens, and Eduard Hovy. 2014. An extension of BLANC to system mentions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 24–29, Baltimore, Maryland. Association for Computational Linguistics.

Yehudit Meged, Avi Caciularu, Vered Shwartz, and Ido Dagan. 2020. Paraphrasing vs coreferring: Two sides of the same coin. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4897–4907, Online. Association for Computational Linguistics.

Emanuele Pianta, Christian Girardi, and Roberto Zanoli. 2008. The TextPro tool suite. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.

M. Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17:485 – 510.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Zhiyi Song, Ann Bies, Justin Mott, Xuansong Li, Stephanie Strassel, and Christopher Caruso. 2018. Cross-document, cross-language event coreference annotation using event hoppers. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, page 45–52, USA. Association for Computational Linguistics.

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

Kristin Wright-Bettner, Martha Palmer, Guergana Savova, Piet de Groen, and Timothy Miller. 2019. Cross-document coreference: An approach to capturing coreference without context. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 1–10, Hong Kong. Association for Computational Linguistics.

Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic annotation suggestions and custom annotation layers in WebAnno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96, Baltimore, Maryland. Association for Computational Linguistics.

Xiaodong Yu, Wenpeng Yin, and Dan Roth. 2020. Paired representation learning for event and entity coreference.

Michelle Yuan, Patrick Xia, Chandler May, Benjamin Van Durme, and Jordan Boyd-Graber. 2022. Adapting coreference resolution models through active learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7533–7549, Dublin, Ireland. Association for Computational Linguistics.

Yutao Zeng, Xiaolong Jin, Saiping Guan, Jiafeng Guo, and Xueqi Cheng. 2020. Event coreference resolution with their paraphrases and argument-aware embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3084–3094, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A ECB+ Corpus Event Statistics

Table 1 contains the detailed statistics for both the ECB+ and the GVC corpora.

B Model Details

B.1 CDLM

The CDLM model, based on the Longformer architecture, cleverly uses a combination of global and local attention for event trigger words and the rest of the document containing those events respectively. More specifically, the Longformer's increased input capacity of 4096 tokens is utilized

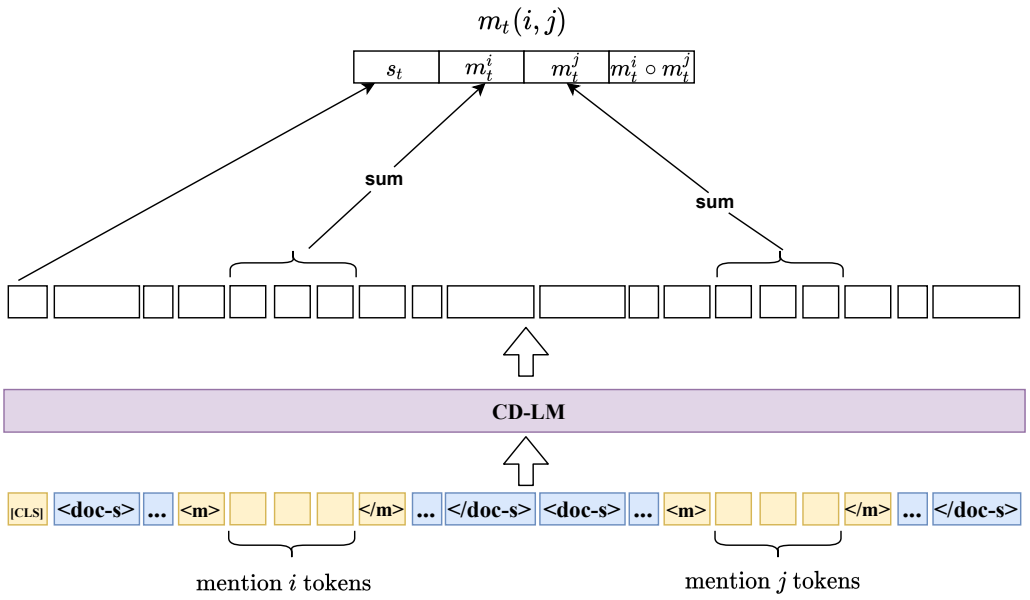

Figure 3: Illustration of Cross-encoding with CDLM from Caciularu et al. (2021).

	ECB+			GVC		
	Train	Dev	Test	Train	Dev	Test
T/ST	25	8	10	170	37	34
D	594	196	206	358	78	74
M	3808	1245	1780	5313	977	1008
C	1464	409	805	991	228	194
S	1053	280	623	252	70	43
P	300K	100K	180K	100K	20K	20K
P_+	15K	6K	6.5K	24K	3.7K	4.1K

Table 1: ECB+ and GVC Corpus statistics for event mentions. T/ST = topics/sub-topics, D = documents, M = event mentions, C = clusters, S = singletons. P = unique mention pairs by topic. P_+ = mention pairs that are coreferent.

to encode much longer documents at finetuning that are usually seen in coreference corpora like the ECB+. As seen in Fig. 3, apart from the document-separator tokens like <doc-s> and <doc-s/> that help contextualize each document in a pair, it adds two special tokens (<m> and </m>) to the model vocabulary while pretraining to achieve a greater level of contextualization of a document pair while attending to the event triggers globally at finetuning. Apart from the event-trigger words, the fine-tuned CDLM model also applies the global attention mechanism on the [CLS] token resulting in a more refined embedding for that document pair while maintaining linearity in the transformer's self-attention.

B.2 BERTScore

BERT-Score is an easy-to-use, low-compute scoring metric that can be used to evaluate NLP tasks that require semantic-similarity matching. This task-agnostic metric uses a base language model like BERT to generate token embeddings and leverages the entire sub-word tokenized reference and candidate sentences (x and $\hat{x}$ in Fig. 4) to calculate the pairwise cosine similarity between the sentence pair. It uses a combination of a greedy-matching subroutine to maximize the similarity scores while

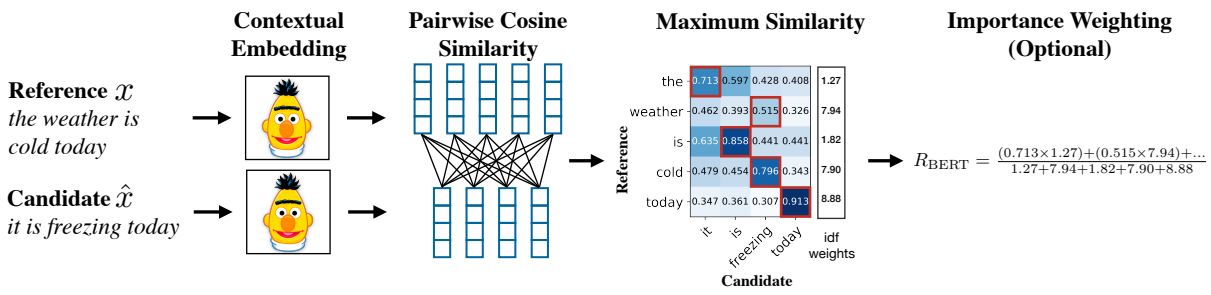

Figure 4: Illustration of the Recall Measure of BERTScore from Zhang et al. (2019).

normalizing the generated scores based on the IDF (Inverse Document Frequency) of the sub-tokens thereby resulting in more human-readable scores. The latter weighting parameter takes care of rare-word occurrences in sentence pairs that are usually more indicative of how semantically similar such pairs are. In our experiments, we use the `distilbert − base − uncased` model to get the pairwise coreference scores, consistent with our goal of deploying an annotation workflow suitable for resource-constrained settings. Such lighter and 'distilled' encoders allow us to optimize resources at inference with minimal loss in performance.

C λ Hyper-parameter Tuning

We employ the evaluation methodology detailed in §4 to determine the optimal value of λ (the weight for trigger similarity and sentence similarity) for both `BERT` and `Lemma` approaches. By conducting incremental annotation simulations on the development sets of ECB+ and GVC, we assess λ values ranging from 0 to 1. The recall-effort curve is plotted for each λ value, as shown in Figure 5, allowing us to identify the one that consistently achieves the highest recall with the fewest comparisons. Remarkably, the optimal value for both methods is found to be 0.7, and this value remains consistent across the two datasets and approaches.

D Annotation Interface using Prodigy

Figure 6 illustrates the interface design of the annotation methodology on the popular model-in-the-loop annotation tool - Prodigy (prodi.gy). We use this tool for the simplicity it offers in plugging in the various ranking methods we explained. The recipe for plugging it in to the tool along with other experiment code: `github.com/ahmeshaf/model_in_coref`.

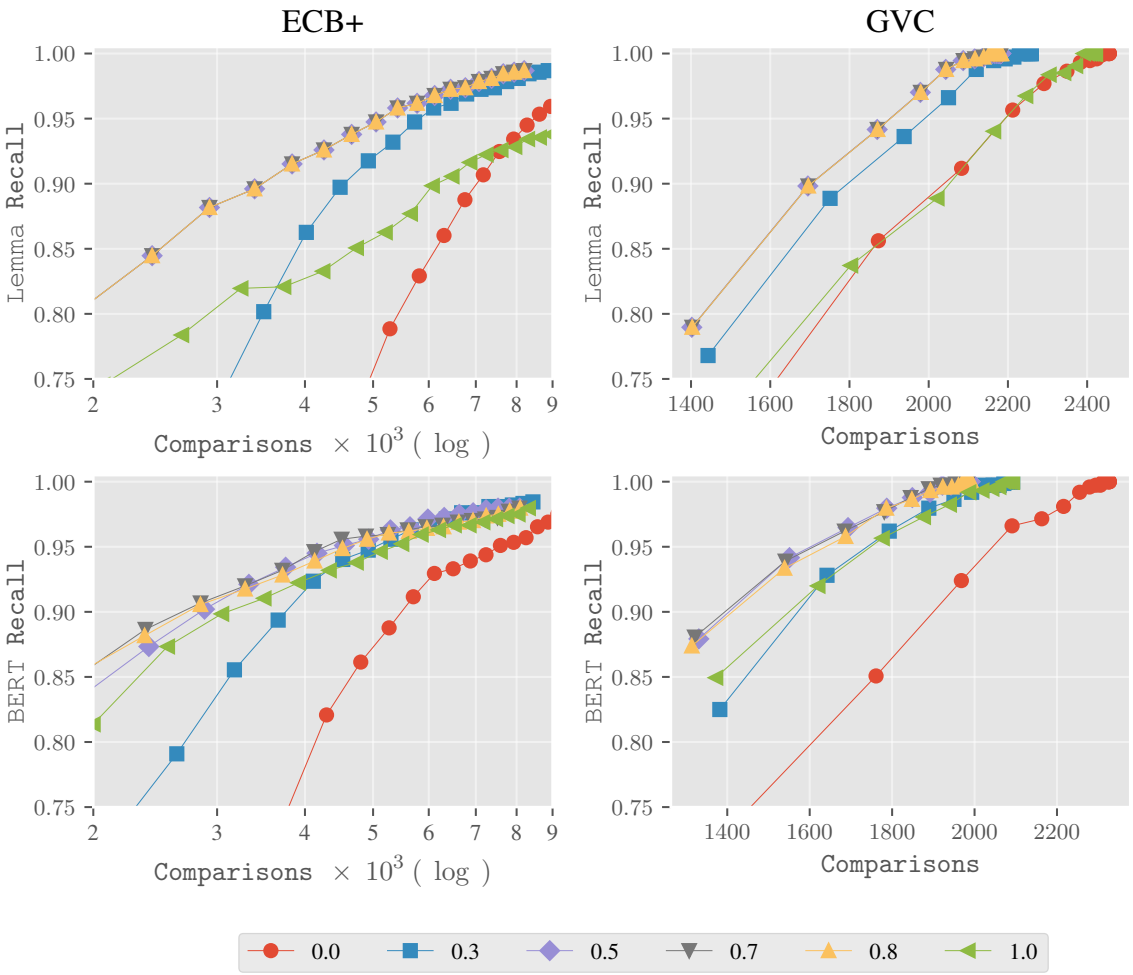

Figure 5: Trigger and Sentence Similarity weight (λ) Hyper-parameter tuning on the development sets of ECB+ and GVC. We deduce $\lambda = 0.7$ is optimal for both methods for both datasets.

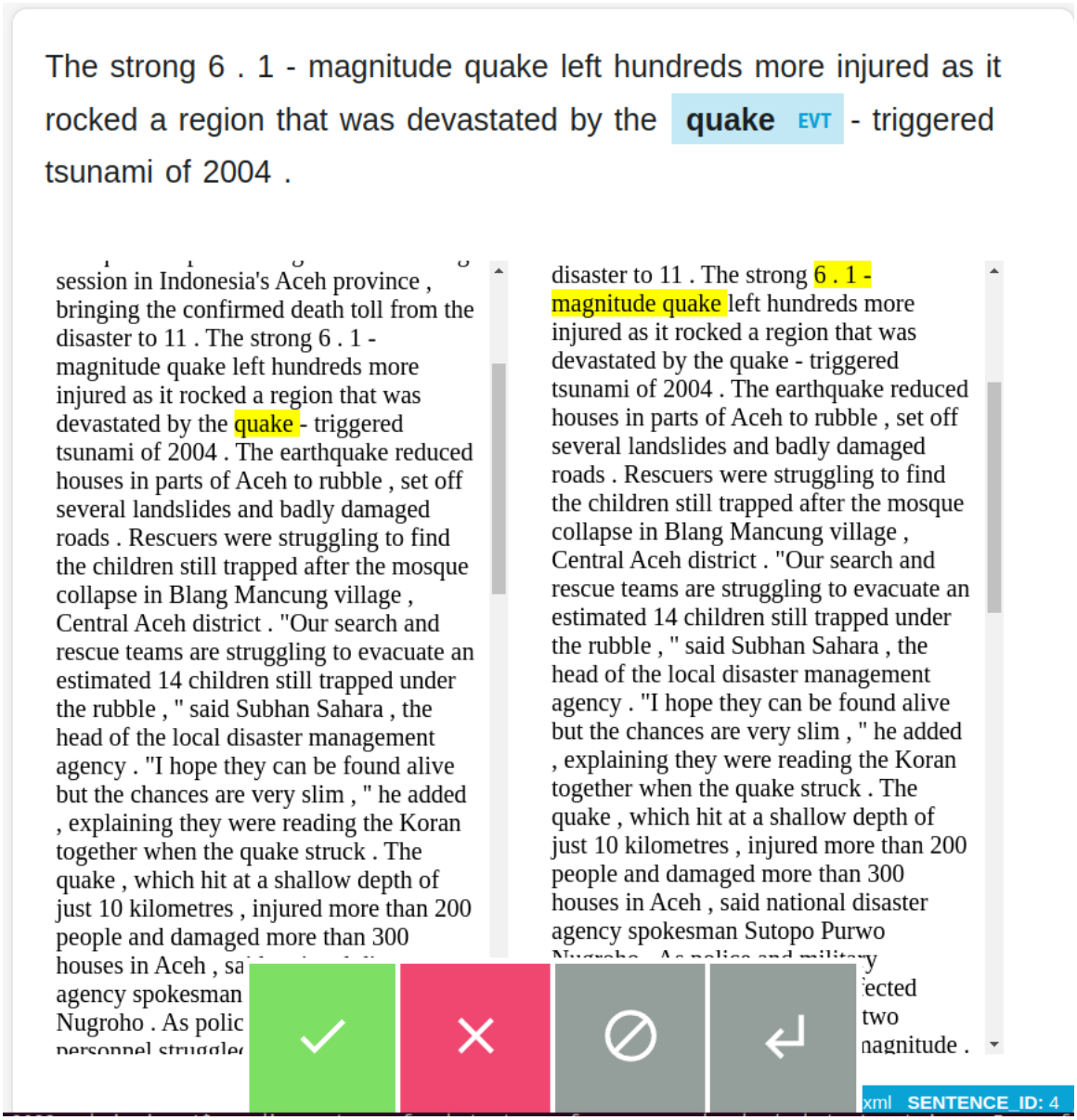

Figure 6: The model-in-the-loop ECR annotation using the Prodigy Annotation Tool. The target event is on the left and the Candidate cluster is on the right.