Event and Entity Coreference using Trees to Encode Uncertainty in Joint Decisions

Nishant Yadav, Nicholas Monath, Rico Angell, and Andrew McCallum
College of Information and Computer Sciences
University of Massachusetts Amherst
{nishantyadav,nmonath,rangell,mccallum}@cs.umass.edu

Abstract

Coreference decisions among event mentions and among co-occurring entity mentions are highly interdependent, thus motivating joint inference. Capturing the uncertainty over each variable can be crucial for inference among multiple dependent variables. Previous work on joint coreference employs heuristic approaches, lacking well-defined objectives, and lacking modeling of uncertainty on each side of the joint problem. We present a new approach of joint coreference, including (1) a formal cost function inspired by Dasgupta's cost for hierarchical clustering, and (2) a representation for uncertainty of clustering of event and entity mentions, again based on a hierarchical structure. We describe an alternating optimization method for inference that when clustering event mentions, considers the uncertainty of the clustering of entity mentions and vice-versa. We show that our proposed joint model provides empirical advantages over state-of-the-art independent and joint models.

1 Introduction

Coreference resolution refers to the task of detecting mentions of various entities and events and identifying groups of mentions referring to the same real-world entity or event. It is a fundamental NLP task that has several downstream applications such as question answering (Chen et al., 2019; Bhattacharjee et al., 2020), textual entailment (Mitkov et al., 2012), building and maintaining KBs (Angeli et al., 2015; Angell et al., 2021), and multi-document summarization (Falke et al., 2017; Huang and Kurohashi, 2021). Often these downstream applications consume a set of documents, and thus require detection of coreference relations between event and entity mentions spread across documents such as multiple news articles.

Event and entity coreference decisions often have rich dependencies on each other. Consider the example in Figure 1 where a system performing

1.) [HP]$_{e1}$ **lands**$_{v1}$ green datacenter consultant [EYP]$_{e2}$.
2.) [HP]$_{e3}$ **acquires**$_{v2}$ [EYP Mission Critical Facilities]$_{e4}$.
3.) [Hewlett-Packard]$_{e5}$ will **acquire**$_{v3}$ [Electronic Data Systems]$_{e6}$ for about $ 13 billion.

Figure 1: Example sentences from ECB+ corpus with event mentions and their argument entity mentions. Color indicate ground-truth coreference clusters.

event coreference independently of entity coreference may erroneously conclude that lands (v1) and acquires (v2) are not coreferent while acquires (v2) and acquire (v3) are coreferent. A joint model that leverages coreference decisions of these events' arguments can help avoid such errors. For instance, coreference relationships between arguments of v1, v2, and v3 could provide crucial evidence in support of v1 and v2 being coreferent, and v2 and v3 being not coreferent.

Previous work that exploits the argument-predicate structure either rely on lexical similarity between arguments while resolving related event mentions (Yang et al., 2015; Choubey and Huang, 2017; Yu et al., 2020, inter alia), or is limited to algorithmic approaches without an explicitly defined cost function (He, 2007; Lee et al., 2012; Barhom et al., 2019). Moreover, these approaches do not capture the uncertainty in coreference decisions during the inference as they represent jointness only at a flat clustering level.

In this work, we present a cost function that captures both the dependency between coreference decisions of event and entity mentions as well as the uncertainty of these decisions. Our proposed cost extends Dasgupta's cost function for hierarchical (tree-structured) clustering (Dasgupta, 2016), generalizing it to model the dependencies between two separate clustering problems of event and entity coreference. To optimize this cost, we describe an efficient inference procedure, which is based on iterated conditional modes. Our inference algorithm captures the uncertainty over clustering decisions of events and entities using the hierar-

Proceedings of the 4th Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2021), pages 100–110
Punta Cana, Dominican Republic, November 10–11, 2021.

chical clusterings or cluster trees over event and entity mentions. The algorithm proceeds by building cluster trees over event and entity mentions, independently. We then alternate between building over event mentions conditioned on the cluster tree over entity mentions, and building cluster trees over entity mentions given the cluster tree over event mentions, until convergence.

Our proposed joint clustering model offers up to 2 CONLL-F_1 points improvement over resolving event and entity mentions independently, and up to 0.75 CONLL-F_1 points improvement over greedy iterative-merge clustering used in prior work.

2 Related Work

Entity coreference resolution, especially in the within-document setting has seen a tremendous amount of improvement over recent years. Early work on entity coreference used hand-crafted syntactic and semantic features (Ng and Cardie, 2002; Daumé III and Marcu, 2005; Durrett and Klein, 2013) while recent top-performing models are neural models that perform mention detection, followed by mention clustering in an end-to-end fashion (Lee et al., 2017, 2018; Meged et al., 2020; Joshi et al., 2020). Event coreference has also seen similar trends with early work using lexical features such as Wordnet synsets, head lemma of the verb (Chen and Ji, 2009; Bejan and Harabagiu, 2010) while more recent work uses event embeddings from pre-trained word embeddings or pretrained language models (Lu and Ng, 2018; Kenyon-Dean et al., 2018; Cattan et al., 2020; Yu et al., 2020). While the majority of the work on event or entity coreference attempts to solve the tasks separately, some prior work does exploit the argument-predicate structure to derive additional features for enriching entity/event representations. In this work, we go a step further and perform joint clustering of event and entity mentions instead of merely using event or entity mentions to derive additional lexical features.

Prior work that jointly predicts event and entity coreference decisions does so at a flat clustering level (Lee et al., 2012; Barhom et al., 2019). The joint clustering method incrementally merges entity or event clusters, and computes merge score between a pair of entity (or event) clusters based on a learned similarity function that incorporates features from the involved mentions as well as clusters of related event (or entity) mentions. However, they use flat clustering of mentions at the given time step to compute merge scores in the next time step. In this work, instead of using an incrementally built flat clustering, we use cluster trees over entity and event mentions that help encode uncertainty over coreference decisions as part of our joint inference.

3 Joint Coreference Model

Problem Definition The task of coreference resolution is a clustering problem where data points are textual spans of event (or entity) mentions and clusters refer to the real-world events (or entities). In this work, we focus on cross-document coreference[1] in which the entity and event mentions participating in the clustering problem come from a corpus of documents. Let $\mathcal{D} = \{D_1, \ldots, D_{n_d}\}$ be a set of n_d documents containing a set of event mentions M_v and a set of entity mentions M_e with ground-truth flat clustering $\mathcal{E}_v^*$ and $\mathcal{E}_e^*$ respectively. Given the documents and set of event and entity mentions as input, the task is to output flat clusterings $\hat{\mathcal{E}}_v$ and $\hat{\mathcal{E}}_e$ of M_v and M_e respectively.

Our approach for joint event and entity coreference uses dependencies between event and entity mentions (§ 3.2). We formalize a cost function that uses an independently trained pairwise similarity (§ 3.4) along with relational similarity (§ 3.5) defined using these dependencies in a way that captures uncertainty over coreference decisions. Finally, we describe our joint inference procedure to optimize the proposed cost function in § 3.7.

3.1 Cluster Trees

We will use a hierarchical clustering or *cluster tree*, denoted $\mathcal{T}$, to represent the uncertainty of coreference decisions. A cluster tree inherently represents multiple alternative flat clusterings, any of which can be selected as the predicted coreference of mentions. These tree structures are a popular choice among cross-document coreference models, and are typically built using hierarchical agglomerative clustering (Green et al., 2012; Kenyon-Dean et al., 2018; Cattan et al., 2020, inter alia).

A cluster tree has mentions as its leaves. For instance, a cluster tree over event mentions would have M_v as its set of leaves. Each internal node represents the cluster of its descendant leaves. An internal node with children x and y is associated with a score from a linkage function, $\mathcal{S}_{x,y}$ which measures similarity between clusters associated with x and y. A canonical way, as used by Cattan

[1] Distinct from the with-in document setting where each document is treated as a separate clustering problem.

et al. (2020) and others, to select a flat clustering from the cluster tree is to use a threshold τ on the linkage score and select the fewest number of clusters possible with linkage greater than τ. In this way, functions of their lowest common ancestor in the tree structure, such as the linkage score, can be informative of the likelihood of two mentions being in the same predicted flat cluster.

3.2 Relational Dependency Edges

Consider the decision of whether two events are coreferent. If the entities involved in the two events are coreferent, we might be more inclined to believe that the two events are coreferent. Similarly, we might be more inclined to believe two entity mentions are coreferent if they participate in the same real-world event. We formalize this intuition by using semantic roles (Carreras and Màrquez, 2005) for modeling dependencies between event mentions and entity mentions.

For each event, we identify entity mentions occurring in specific semantic roles. In this work, we use four semantic role labels – `ARG0`, `ARG1`, `ARG-LOC` and `ARG-TMP`, and collect a set of **relational dependency edges** between entity mentions and event mentions. In particular, we use $\text{arg}_{\texttt{r}}(m_i)$ to denote the entity mention in the semantic role $\texttt{r}$ for event mention m_i, and $\text{arg}_{\texttt{r}}^{-1}(m_i)$ to denote the event mention for which the entity mention m_i occurs in role $\texttt{r}$. In case an event mention m_i does not have an argument in a semantic role $\texttt{r}$, $\text{arg}_{\texttt{r}}(m_i) = \epsilon$. Similarly, $\text{arg}_{\texttt{r}}^{-1}(m_i) = \epsilon$ if an entity mention m_i does not occur in role $\texttt{r}$ for any event mention. Consider the example in Figure 1 where $\text{arg}_{\texttt{ARG0}}$(lands) = HP, $\text{arg}_{\texttt{ARG1}}$(lands) = EYP, and $\text{arg}_{\texttt{ARG1}}^{-1}$(EYP) = lands.

3.3 Joint Cost Function

We will be building two cluster trees– one for event mentions and the other for entity mentions. We formalize a cost function that assigns a cost to the pair of cluster trees considering both mention similarities as well as the dependency edges between event and entity mentions.

We begin by defining a cost for a single tree that does not include dependency edges using Dasgupta's cost (Dasgupta, 2016) for hierarchical clustering. Let $\mathcal{T}_v$ be a cluster tree over event mentions, and $\texttt{LLCA}_{i,j}^{(v)}$ be the number of leaves under the lowest common ancestor (LCA) of event mentions m_i and m_j, and $f_{\text{ind}}^{(v)}$ be pairwise mention similarity function over event mentions computed *independently* of entity mentions. Dasgupta's cost of $\mathcal{T}_v$ is given by

$$J_{\text{ind}}(\mathcal{T}_v) = \sum_{m_i, m_j \in M_v} f_{\text{ind}}^{(v)}(m_i, m_j)\texttt{LLCA}_{i,j}^{(v)}$$

We similarly define cost $J_{\text{ind}}(\mathcal{T}_e)$ for tree $\mathcal{T}_e$ over entity mentions. In words, each pair of mentions pays a cost that is its similarity $f_{\text{ind}}^{(v)}(m_i, m_j)$ times number of leaves under their lowest common ancestor. Intuitively, in order to minimize the cost, a pair of mentions m_i, m_j, that has high pairwise similarity should be merged near the leaf level of the tree (i.e., more likely to be in the same flat cluster) so that they have a small value of $\texttt{LLCA}_{i,j}^{(v)}$ while pairs of mentions with low similarity could be placed thus far apart in the tree e.g., potentially with the tree root as the lowest common ancestor.

To capture information from our relational dependency edges, we introduce a relational similarity term $f_{\text{rel}}^{(v)}$. For a pair of event mentions, $f_{\text{rel}}^{(v)}$ estimates event similarity by using the cluster-assignments of the entity mentions involved in dependency edges with the two event mentions. We propose several ways to define $f_{\text{rel}}^{(v)}$ in §3.5. We define the cost of a cluster tree over event mentions ($\mathcal{T}_v$) given a cluster tree over entity mentions ($\mathcal{T}_e$) as follows:

$$J_{\text{rel}}(\mathcal{T}_v|\mathcal{T}_e) = \sum_{m_i, m_j \in M_v} f_{\text{rel}}^{(v)}(m_i, m_j)\texttt{LLCA}_{i,j}^{(v)}$$

A pair of event mentions has high relational similarity if their arguments are present in the same flat clustering or present close to each other in their cluster tree. And, the cost function $J_{\text{rel}}(\mathcal{T}_v|\mathcal{T}_e)$ is minimized when pairs of event mentions with high relational similarity are placed close in the cluster tree, thus increasing their likelihood of being in the same inferred flat clustering. This is desired behavior based on the assumption that coreferring arguments increase the likelihood of two events being coreferent and vice-versa. We define the joint cost as a weighted combination of the independent and relational costs as follows:

$$\begin{aligned} J_{\text{joint}}(\mathcal{T}_v, \mathcal{T}_e) = \alpha_{\text{ind}}^{(v)} J_{\text{ind}}(\mathcal{T}_v) + \alpha_{\text{ind}}^{(e)} J_{\text{ind}}(\mathcal{T}_e) + \\ \beta_{\text{rel}}^{(v)} J_{\text{rel}}(\mathcal{T}_v|\mathcal{T}_e) + \beta_{\text{rel}}^{(e)} J_{\text{rel}}(\mathcal{T}_e|\mathcal{T}_v) \quad (1) \end{aligned}$$

where $\alpha_{\text{ind}}^{(t)}, \beta_{\text{rel}}^{(t)} \in \mathbb{R}, t \in \{\text{entity}, \text{event}\}$.

We can re-write the joint cost as a function of mention pairs by substituting expressions for each

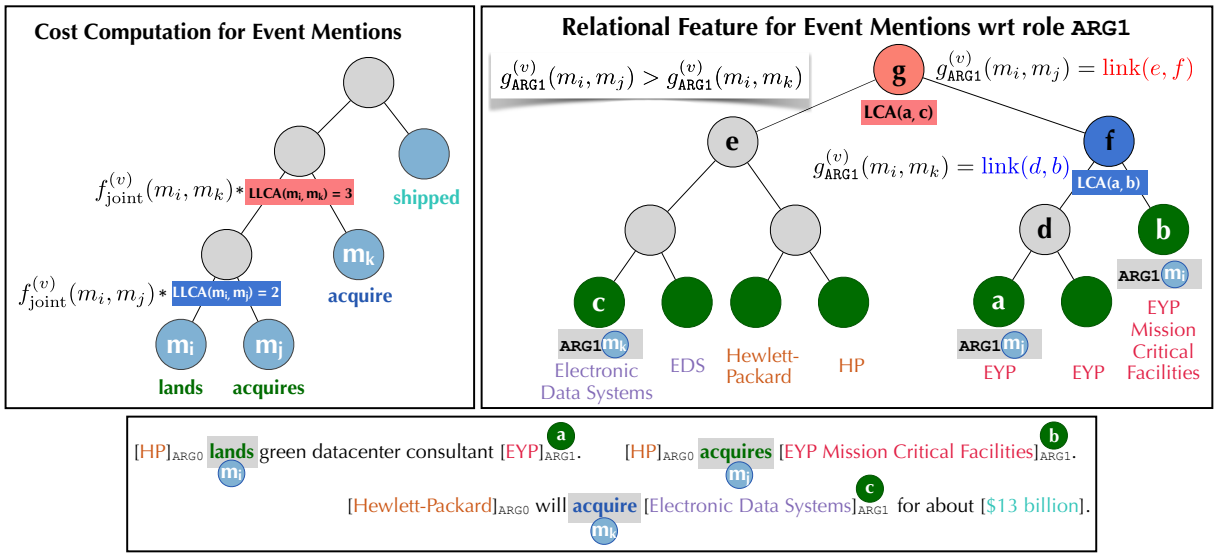


Figure 2: Illustration of the cost function and the joint relational dependence model using the tree structure. Three sentences from the corpus are shown in the lower pane. We focus on a pair of event mentions "lands" and "acquires" from the example in Figure 1. In the upper left pane, we show the computation of two terms in the cost for m_i & m_j and m_i & m_k, which involves the product of their similarities (using the joint model) and the number of leaves of their lowest common ancestors (LCA). In the upper right pane, we show the computation of the LINKAGE-SCORE relational similarity feature ($g^{(v)}_{\text{ARG1}}$) wrt ARG1 of these event mentions using linkage score of their LCA. Observe how this relational feature uses the LCA in a different way than the cost function. Here we use the linkage score at the LCA to define relational feature $g^{(v)}_{\text{ARG1}}$.

term in the cost function as follows:

$$J_{\text{joint}}(\mathcal{T}_v, \mathcal{T}_e) = \underbrace{\sum_{m_i, m_j \in M_v} f^{(v)}_{\text{joint}}(m_i, m_j) \text{LLCA}^{(v)}_{i,j}}_{1} + \underbrace{\sum_{m_i, m_j \in M_e} f^{(e)}_{\text{joint}}(m_i, m_j) \text{LLCA}^{(e)}_{i,j}}_{2} \quad (2)$$

where, for $t \in$ {entity, event}

$$f^{(t)}_{\text{joint}}(m_i, m_j) = \alpha^{(t)}_{\text{ind}} f^{(t)}_{\text{ind}}(m_i, m_j) + \beta^{(t)}_{\text{rel}} f^{(t)}_{\text{rel}}(m_i, m_j), \quad (3)$$

Figure 2 shows example computation of the first term in Eq 2. In the following sections, we describe how we train and compute different components of the joint similarity function.

3.4 Pairwise Mention Scorer ($f^{(t)}_{\text{ind}}$)

We use the recently proposed state-of-the-art independent coreference model proposed by Cattan et al. (2020) as our pairwise mention scorers $f^{(v)}_{\text{ind}}$, $f^{(e)}_{\text{ind}}$ for event and entity mentions respectively. This model encodes event (or entity) mentions in context using RoBERTa (Liu et al., 2019). The mention embeddings and their element-wise product of their embeddings are concatenated to create a mention-pair representation which is then passed through an MLP to return a similarity score for the given pair of mentions. Following Cattan et al. (2020), we freeze RoBERTa parameters and only train parameter of the MLP using binary cross-entropy loss over all pairs of mentions.

3.5 Relational Similarity ($f^{(t)}_{\text{rel}}$)

The core of our joint approach is in the way relational dependency edges between events and entities contribute to the cost function. These relational edges will contribute to the relational similarity $f^{(v)}_{\text{rel}}(m_i, m_j)$ (and thereby to the cost function through the joint similarity). The relational similarities we propose will use the tree structure to represent uncertainty over the final flat clustering assignment of mentions to clusters.

Let's begin by considering the definition of relational similarity for event mentions (the similarity for entity mentions is defined analogously). Given a pair of event mentions m_i, m_j, relational similarity $f^{(v)}_{\text{rel}}(m_i, m_j)$ is computed using their semantic role arguments. Specifically, we define a relational feature $g^{(v)}_{\text{r}}(m_i, m_j)$ using similarity $f^{(e)}_{\text{sim}}$ between entity mentions which are arguments of event men-

tions m_i, m_j in semantic role $\mathtt{r}$ as:

$$g^{(v)}_{\mathtt{r}}(m_i, m_j) = f^{(e)}_{\text{sim}}(\text{arg}_{\mathtt{r}}(m_i), \text{arg}_{\mathtt{r}}(m_j)) \quad (4)$$

Relational similarity $f^{(v)}_{\text{rel}}(m_i, m_j)$ is then computed using weights $\phi^{(v)}$ over these features as:

$$f^{(v)}_{\text{rel}}(m_i, m_j) = \sum_r \phi^{(v)}_r g^{(v)}_{\mathtt{r}}(m_i, m_j)$$
$$= \sum_r \phi^{(v)}_r f^{(e)}_{\text{sim}}(\text{arg}_{\mathtt{r}}(m_i), \text{arg}_{\mathtt{r}}(m_j))$$

The similarity $f^{(e)}_{\text{sim}}$ will be based on the coreference decision of the pair of entity arguments. Most simply, one could use hard assignments of entity mentions to flat clusters, such as:

- FLAT-CLUST: $f^{(e)}_{\text{sim}}(m_i, m_j) = 1$ if m_i & m_j belong to the same cluster, and 0 otherwise.

However, we would like to capture the uncertainty over (hard) flat cluster assignments using the tree-structured clustering. To do this, we define $f^{(e)}_{\text{sim}}$ as a function of $\text{LCA}_{i,j}$, the lowest common ancestor of mentions m_i, m_j in $\mathcal{T}_e$ as follows:

- LINKAGE-SCORE: $f^{(e)}_{\text{sim}}(m_i, m_j) = -\mathcal{S}_{i,j}$ where $\mathcal{S}_{i,j}$ gives the linkage score between the left and right child nodes of $\text{LCA}_{i,j}$ in $\mathcal{T}_e$.
- NODE-ORDER: $f^{(e)}_{\text{sim}}(m_i, m_j) = -\mathcal{O}_{i,j}$, where $\mathcal{O}_{i,j}$ is the position of $\text{LCA}_{i,j}$ in the array of internal nodes of $\mathcal{T}_e$, sorted in increasing order of their linkage scores.

Figure 2 shows computation of relational feature ($g^{(v)}_{\texttt{ARG1}}$) for pairs of event mentions using LINKAGE-SCORE method. Analogous similarity functions are defined for entity mentions, except in terms of the inverse relational dependencies, i.e.:

$$f^{(e)}_{\text{rel}}(m_i,m_j) = \sum_r \phi^{(e)}_r g^{(e)}_{\mathtt{r}}(m_i, m_j)$$
$$= \sum_r \phi^{(e)}_r f^{(v)}_{\text{sim}}(\text{arg}^{-1}_{\mathtt{r}}(m_i), \text{arg}^{-1}_{\mathtt{r}}(m_j))$$

We also experiment with relational similarity features $g^{(v)}_{\mathtt{r}}$ for a pair of event mentions computed using $f^{(e)}_{\text{sim}} = f^{(e)}_{\text{ind}}$ instead of defining $f^{(e)}_{\text{sim}}$ using a (hierarchical) clustering of entity mentions. Additionally, we experiment with relational similarity features computed using $f^{(e)}_{\text{sim}}$ defined using clustering of entity mentions as well as using $f^{(e)}_{\text{sim}} = f^{(e)}_{\text{ind}}$. We define relational features analogously for entity mentions as well.

3.6 Joint-Mention Pair Scorer

In order to compute joint mention-pair similarity score for mentions of type $t \in \{\text{entity}, \text{event}\}$, we first compute a joint mention-pair feature vector using the *trained* pairwise similarity $f^{(t)}_{\text{ind}}$, and relational similarity features ($g^{(t)}_{\mathtt{r}}$) based on the four semantic roles- `ARG0`, `ARG1`, `ARG-LOC` and `ARG-TMP`. However, in order to compute these relational features for event (or entity) mentions, we would need a clustering over entity (or event) mentions to begin with. Thus, in order to bootstrap the process, we initially cluster entity (or event) mentions only using scores from $f^{(e)}_{\text{ind}}$ (or $f^{(v)}_{\text{ind}}$). We then compute relational similarity feature values using the cluster trees or the flat clusterings over event and entity mentions.

We finally train a linear classifier on top of this joint feature vector to generate final joint pairwise similarity as shown in Figure 3. The linear classifier is trained using binary cross-entropy loss to classify a given pair of mentions as coreferent or not.

3.7 Inference

Given trained pairwise mention similarity functions ($f^{(v)}_{\text{ind}}, f^{(e)}_{\text{ind}}$) and trained joint mention-pair scorers ($f^{(v)}_{\text{joint}}, f^{(e)}_{\text{joint}}$), our joint inference procedure starts by clustering event mentions and entity mentions independently using $f^{(v)}_{\text{ind}}$ and $f^{(e)}_{\text{ind}}$ respectively, with average linkage hierarchical agglomerative clustering (Johnson, 1967). We then alternate between rebuilding the cluster tree over event mentions and the cluster tree over entity mentions. Given a cluster tree over entity mentions, we first compute the joint pairwise feature vector, then compute the joint-pairwise similarity ($f^{(v)}_{\text{joint}}$), and use it to compute hierarchical clustering over event mentions. We similarly update clustering over entity mentions given the updated tree over event mentions and alternate until the cost function in Eq 2 has converged.

We use hierarchical agglomerative clustering (HAC) to build cluster trees as it allows use of an arbitrary pairwise scoring function and is widely used for coreference resolution in prior work (Lee et al., 2012; Clark and Manning, 2016; Barhom et al., 2019; Cattan et al., 2020, inter-alia). Moreover, average-linkage HAC provides a 2-approximation to a dissimilarity-based version of Dasgupta's cost function (Cohen-Addad et al., 2019). Our proposed inference alternates between optimizing the first and the second term in Eq 2. Clustering event men-

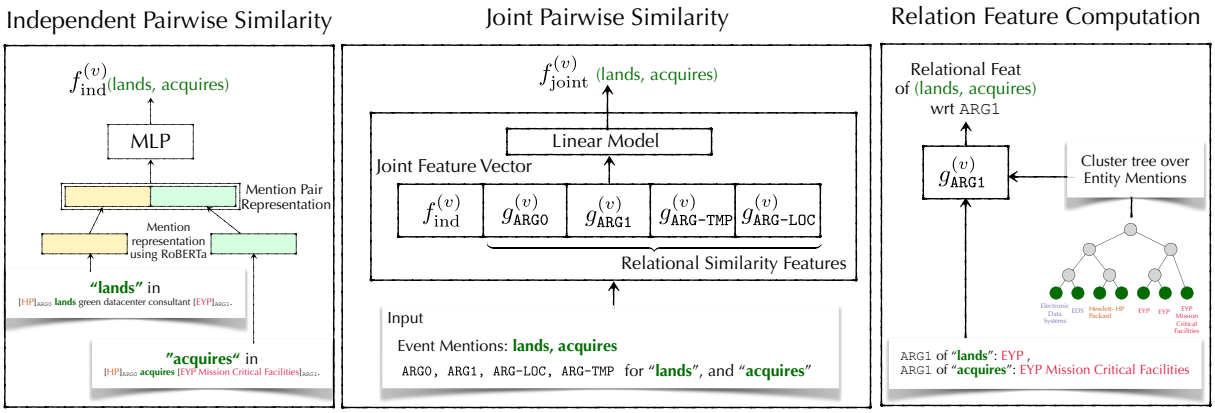


Figure 3: Illustration of joint similarity computation for a pair of event mentions "lands" and "acquires" from the example in Figure 1. Left panel in the figure shows computation of pairwise mention scorer $f^{(v)}_{\text{ind}}$ and the right panel shows computation of relational similarity feature for the two event mentions using entity mentions in semantic role ARG1, and the cluster tree over entity mentions. The central panel shows the joint feature vector computed using pairwise mention score from $f^{(v)}_{\text{ind}}$ and relational similarity features ($g^{(v)}_{\text{ARG0}}, g^{(v)}_{\text{ARG1}}, g^{(v)}_{\text{ARG-LOC}}, g^{(v)}_{\text{ARG-TMP}}$) is fed into a linear model to produce final joint pairwise similarity.

tions given a cluster tree over entity mentions using the joint similarity function $f^{(v)}_{\text{joint}}$ optimizes the first term in Eq. 2 while clustering entity mentions given event mentions using the joint similarity function $f^{(e)}_{\text{joint}}$ optimizes the second term in Eq. 2. Finally, on convergences, we use a threshold to pick a flat clustering over event and entity mentions given their cluster trees. At test time, we use threshold values picked using dev data.

4 Experiments

Dataset We run experiments on the ECB+ corpus which consists of both within- and cross-document coreference resolution for *both* entity and event mentions. ECB+ corpus consists of a set of news articles, grouped under topics. Each topic consists of two sub-topics containing articles about two different real-world events which use similar vocabulary, thus posing a challenge for coreference systems. For example, the first two sentences in Figure 1 are from the same subtopic containing documents of HP's acquisition of EYP while the third sentence comes from a document from a different subtopic which talks about HP's acquisition of EDS. This presents a challenging setting when evaluating cross-document coreference at the topic level as there are instances of coreferent event and entity mentions with divergent surfaces forms as well as mentions of different real-world events and entities with similar surface forms. We show statistics on ECB+ in Table 1.

	Train	Dev	Test	Total
# Topics	25	8	10	43
# Documents	574	196	206	976
# Event Mentions	3808	1245	1780	6833
# Event Clusters	1527	409	805	2741
# Entity Mentions	4758	1476	2055	8289
# Entity Clusters	1286	330	608	2224

Table 1: Statistics on the standard train/dev/test split of ECB+ dataset (Cybulska and Vossen, 2014).

Evaluation Metrics and Setup We use the official CoNLL scorer (Pradhan et al., 2014) and report coreference resolution metrics like MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), $CEAF_e$ (Luo, 2005) and the average of these three metric, the CONLL-F_1 metric. We follow the evaluation setup of (Cybulska and Vossen, 2014), also used in other recent work (Kenyon-Dean et al., 2018; Barhom et al., 2019; Cattan et al., 2020). Following the precedent set by recent work, we remove singletons from both gold as well as predicted clusters during evaluation. We use gold event and entity mentions, evaluate coreference performance at topic level with gold topics, and finally compute a micro average of the metrics across all topics.

4.1 Implementation Details and Baselines

Proposed Models Recall that our joint clustering model requires a similarity measure $f^{(t)}_{\text{sim}}, t \in \{\text{event}, \text{entity}\}$ between mentions to compute relational similarity features which are then use to compute the joint pairwise similarity as described in § 3.5 and § 3.6. We experiment with four differ-

ent choices for this similarity measure: JOINTINF-PW uses independent pairwise mention similarity $f_{\text{ind}}^{(t)}$, JOINTINF-PF uses similarity computed using predicted flat clusterings in addition to $f_{\text{ind}}^{(t)}$, and JOINTINF-NO and JOINTINF-LS use cluster tree based similarity computed using NODE-ORDER and LINKAGE-SCORE methods respectively in addition to $f_{\text{ind}}^{(t)}$.

Baselines We compare our proposed joint clustering model with the following baselines:

- IND: Compute clustering of event and entity mentions *independently* using $f_{\text{ind}}^{(t)}, t \in \{\text{event, entity}\}$. This corresponds to results[2] for models proposed in Cattan et al. (2020).

- GREEDY-MERGE:This refers to the inference procedure used by Lee et al. (2012); Barhom et al. (2019) with the features proposed in this paper. It starts with event and entity mentions in singleton clusters. At each time step, it computes the joint pairwise event (entity) mention similarity using the flat clustering of entity (event) mentions at the given time step, updates the linkage score between every pair of clusters, and finally greedily merges the pair of event or entity mention clusters with the highest linkage score. Like our proposed models, it uses $f_{\text{ind}}^{(t)}$ in addition to similarity computed using predicted flat clusterings for joint similarity computation.

Previous work on joint event and entity clustering (Lee et al., 2012; Barhom et al., 2019) uses different mention encoders and features, and/or different evaluation setting. For this reason, our results are not directly comparable with those. GREEDY-MERGE baseline uses a clustering algorithm similar to that used in previous work, on top of the mention encoder and features used in this work.

Note that, JOINTINF-PF and GREEDY-MERGE are similar in that they both use a flat clustering in the joint similarity computation with one important difference – JOINTINF-PF predicts a flat clustering over *all* event (entity) mentions *before* using updating clustering of entity (event) mentions while GREEDY-MERGE starts with a flat clustering with all singletons, and the flat clustering over event and entity mentions used in joint similarity calculations is updated after every cluster merge. We present further implementation details in Appendix A.

4.2 Results

Tables 2a and 2b show performance of our proposed clustering model and baselines for event and entity mentions respectively in ECB+ test corpus. Our joint clustering provides 1.97 and 1.18 points improvement in CONLL-F_1 over clustering entity and event mentions independently. Our joint clustering model outperforms the GREEDY-MERGE, the widely used joint clustering mechanism in previous work by up to 0.75 CONLL-F_1, thus indicating the importance of encoding uncertainty in clustering decisions as well as the efficacy of our methods in doing so. JOINTINF-NO and JOINTINF-LS which use cluster trees to compute relational similarity does better than JOINTINF-PF indicating that encoding uncertainty over coreference decisions using trees performs better than using a single predicted flat clustering. Finally, we note that JOINTINF-PW, which uses the trained pairwise similarity $f_{\text{ind}}^{(t)}$ instead of clustering-based similarity for computing relational similarity provides improvement over IND baseline, but it is outperformed by methods like JOINTINF-LS, and JOINTINF-NO which use cluster trees to compute relational similarity.

Furthermore, since GREEDY-MERGE recomputes pairwise features and scores after *each* merger of a pair of clusters during hierarchical clustering, it does not yield a time-efficient implementation. In contrast, our proposed inference methods compute joint pairwise scores only at the beginning of each round of inference and can thus leverage existing efficient hierarchical clustering libraries. We observed our proposed inference to converge in up to three rounds thus yielding a time-efficient joint inference procedure.

Figure 4 shows CONLL-F_1 scores for IND, GREEDY-MERGE and JOINTINF-LS for each topic in ECB+ test data. It shows that although joint inference methods such as GREEDY-MERGE and JOINTINF-LS offer performance improvement over IND baseline for majority of the topics, they do not offer improvement on a few topics such as topic 36, and 45 for event coreference.

We further analyze the performance of these models at the level of pairwise coreference decisions. We report a few examples in Table 3 where the joint inference method JOINTINF-LS

[2] IND baseline results differ slightly from those reported in Cattan et al. (2020) as we use a threshold chosen exhaustively on dev set while Cattan et al. (2020) chose a threshold from a heuristically fixed set of threshold values of {0.5, 0.55, 0.6, 0.65, 0.7}.

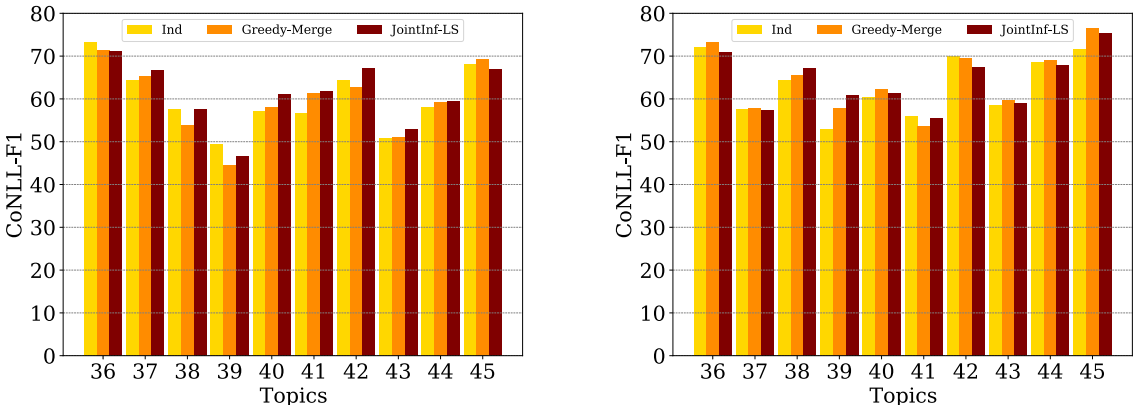
Clustering Model	MUC			B^3			$CEAF_e$			CONLL-F_1
	P	R	F_1	P	R	F_1	P	R	F_1	
Baselines										
IND	74.03	84.20	78.79	48.82	69.01	57.18	43.28	54.46	48.23	61.40
GREEDY-MERGE	77.14	78.25	77.69	56.33	59.66	57.95	44.31	56.98	49.85	61.83 (+0.43)
Proposed Joint Models										
JOINTINF-PW	77.19	79.48	78.32	55.37	62.32	58.64	44.97	57.33	50.41	62.46 (+1.06)
JOINTINF-PF	77.21	78.56	77.88	56.48	60.10	58.23	44.61	56.62	49.90	62.00 (+0.60)
JOINTINF-NO	76.36	80.20	78.23	54.03	63.40	58.35	44.54	56.78	49.92	62.17 (+0.77)
JOINTINF-LS	76.57	82.15	79.26	52.65	65.90	58.53	44.57	56.82	49.96	**62.58 (+1.18)**

(a) Event Coreference

Clustering Model	MUC			B^3			$CEAF_e$			CONLL-F_1
	P	R	F_1	P	R	F_1	P	R	F_1	
Baselines										
IND	77.46	89.05	82.86	49.37	72.29	58.67	48.06	50.73	49.36	63.63
GREEDY-MERGE	79.60	88.64	83.87	52.84	71.78	60.87	48.22	52.60	50.32	65.02 (+1.39)
Proposed Joint Models										
JOINTINF-PW	78.73	89.47	83.76	50.39	73.35	59.74	49.28	49.28	49.28	64.26 (+0.63)
JOINTINF-PF	78.97	90.02	84.14	50.25	74.12	59.89	50.54	49.26	49.89	64.64 (+1.01)
JOINTINF-NO	79.33	87.46	83.20	55.32	70.83	62.12	48.58	54.72	51.47	**65.60 (+1.97)**
JOINTINF-LS	80.02	87.67	83.67	55.07	69.14	61.30	46.16	53.15	49.41	64.79 (+1.16)

(b) Entity Coreference

Table 2: Precision/Recall/F_1 scores for Event and Entity coreference on ECB+ test set over gold mentions spans. Relative improvements in CONLL-F_1 over IND baseline are shown in parenthesis.



(a) Event Mentions (b) Entity Mentions

Figure 4: Performance of IND, GREEDY-MERGE, and JOINTINF-LS on each test topic.

makes correct coreference decisions but IND fails, along with a potential explanation for the correct/incorrect decision made by each model. We also include a similar analysis to show where our joint inference method produces incorrect coreference decisions in Table 4. We would like to note that it is not always possible to provide an explanation for an incorrect coreference decision made by the inference algorithm as there could be multiple sources of error. An incorrect coreference decision could be a result of an incorrect value of pairwise similarity from a deep neural network such as the one used for $f_{\text{ind}}^{(v)}$ and $f_{\text{ind}}^{(e)}$, an incorrectly chosen threshold to obtain flat clustering, or errors due to non-local decisions made by the clustering algorithm which override the mention pairwise similarities amongst other potential reasons.

5 Conclusion

In this paper, we present a joint entity and event coreference model, which formally defines a cost function that extends Dasgupta's cost for hierarchical clustering and represents the uncertainty of coreference decisions along with entity-event dependencies. We present an efficient inference procedure for this cost function and empirically val-

	Event Examples
1a	[Hewlett - Packard]$_{\text{ARG0}}$ to [**buy**]$_{\text{V}}$ consulting [firm EYP Mission Critical Facilities]$_{\text{ARG1}}$.
1b	[HP]$_{\text{ARG0}}$ [**lands**]$_{\text{V}}$ green datacenter consultant [EYP]$_{\text{ARG1}}$ *Explanation*: The two event mentions "buy" and "lands" refer to the same event but have different surface forms. IND fails to cluster them while JOINTINF-LS correctly clusters them together as these two events have coreferent arguments.
2a	[HP]$_{\text{ARG0}}$ [**acquires**]$_{\text{V}}$ [EYP Mission Critical Facilities]$_{\text{ARG1}}$.
2b	[Hewlett-Packard]$_{\text{ARG0}}$ will [**acquire**]$_{\text{V}}$ [Electronic Data Systems]$_{\text{ARG0}}$ for about $ 13 billion. *Explanation*: The two event mentions "acquire" and "acquires" refer to two different real world events but have the same surface form. IND model incorrectly predicts that these two mentions are coreferent while the joint model correctly predicts them as non-coreferent, apparently due to different arguments in ARG1 semantic role.
	Entity Examples
1a	UNHCR condemns [air attack]$_{\text{V}}$ [**on refugee camp in South Sudan**]$_{\text{ARG1}}$.
1b	South Sudan accuses Sudan of [air strike]$_{\text{V}}$ [**on refugee camp**]$_{\text{ARG1}}$. *Explanation*: The two entity mentions refer to the same entity. JOINTINF-LS correctly predicts them as coreferent, apparently because both the entity mentions appear as ARG1 for a coreferent pair of event mentions. IND model incorrectly predicts them as non-coreferent.
2a	The [**UN refugee agency**]$_{\text{ARG0}}$ on Friday strongly [condemned]$_{\text{V}}$ the aerial bombing ...
2b	The mortar strike in Jabaliya was the second attack on a school [run]$_{\text{V}}$ by the [**UN Relief and Works Agency**]$_{\text{ARG0}}$ *Explanation*: The two entity mentions refer to different entities but have some surface form similarity. JOINTINF-LS correctly predicts them as non-coreferent apparently because they appear as ARG0 of two different events but IND model incorrectly predicts them as coreferent.

Table 3: Examples of entity and events pairs (in bold face) from test data where the joint model (JOINTINF-LS) is able to correctly predict coreference decisions and the IND model fails.

	Event Examples
1a	[Apple CEO Tim Cook]$_{\text{ARG0}}$ took the stage in San Francisco to welcome developers and [**announce**]$_{\text{V}}$ new [stuff]$_{\text{ARG1}}$.
1b	Apple has chosen WWDC [week]$_{\text{ARG0}}$ to [**announce**]$_{\text{V}}$ an update to its MacBook Air line. *Explanation*: The two event mentions refer to the same event and have the same surface form. IND model correctly clusters them together while JOINTINF-LS predicts them as being non-coreferent apparently due to mismatch between their ARG0 event mentions. Note that "week" is *incorrectly* identified as ARG0 of "announce" event mention.
2a	[Matt Smith], 26, will make [his]$_{\text{ARG0}}$ [**debut**]$_{\text{V}}$ [in 2010]$_{\text{ARG1}}$, replacing David Tennant ...
2b	Peter Capaldi is officially set to replace [**exiting**]$_{\text{V}}$ [star Matt Smith]$_{\text{ARG0}}$ as the TARDIS leader. *Explanation*: The two event mentions "debut" and "exiting" are not coreferent and have different surface forms. IND model makes a correct prediction while JOINTINF-LS clusters them together as the ARG0 entity mentions refer to the same entity.
	Entity Examples
1a	[**Apple Inc**]$_{\text{ARG0}}$ on Tuesday (Jan 6) [introduced]$_{\text{V}}$ what it claims to ...
1b	[**Apple**]$_{\text{ARG0}}$ [unveils]$_{\text{V}}$ new MacBook Pro with Ivy Bridge at WWDC. *Explanation*: The two event mentions refer to the same entity and have similar surface form. IND model correctly predicts them as coreferent but JOINTINF-LS model makes an incorrect prediction, apparently because the entity mentions appear as ARGO of two distinct events.
2a	4.6-magnitude earthquake [shakes]$_{\text{V}}$ [**Lake County**]$_{\text{ARG1}}$.
2b	Lake County earthquake [shakes]$_{\text{V}}$ [**Napa**]$_{\text{ARG1}}$. *Explanation*: The two event mentions refer to two different entities and have very different surface forms. For this reason, IND model correctly predicts them as being non-coreferent but JOINTINF-LS clusters them together, apparently because the two entity mentions appear as ARG1 for mentions of the same event.

Table 4: Examples of entity and events pairs (in boldface) from test data where the joint model (JOINTINF-LS) fails but IND is able to correctly predict coreference decisions

idate the model by demonstrating state-of-the-art results on the ECB+ dataset, outperforming both independent as well as joint coreference approaches proposed in prior work.

Acknowledgements

We thank Ari Kobren, Akshay Krishnamurthy and members of UMass IESL for helpful discussions and feedback. We would also like to thank Arie Cattan for providing pretrained event and entity coreference models. This work was supported in part by the Center for Data Science and the Center for Intelligent Information Retrieval, in part by the National Science Foundation under Grant No. NSF-1763618, in part by the Chan Zuckerberg Initiative under the project "Scientific Knowledge Base Construction", in part by International Business Machines Corporation Cognitive Horizons Network agreement number W1668553, and in part using high performance computing equipment obtained under a grant from the Collaborative R&D Fund managed by the Massachusetts Technology Collaborative. Rico Angell is supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1938059. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

Rico Angell, Nicholas Monath, Sunil Mohan, Nishant Yadav, and Andrew McCallum. 2021. Clustering-based inference for biomedical entity linking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2598–2608.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

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.

Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422.

Santanu Bhattacharjee, Rejwanul Haque, Gideon Maillette de Buy Wenniger, and Andy Way. 2020. Investigating query expansion and coreference resolution in question answering on bert. In *International Conference on Applications of Natural Language to Information Systems*, pages 47–59. Springer.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2020. Streamlining cross-document coreference resolution: Evaluation and modeling. *arXiv preprint arXiv:2009.11032*.

Jifan Chen, Shih-ting Lin, and Greg Durrett. 2019. Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*.

Zheng Chen and Heng Ji. 2009. Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 54–57.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2124–2133.

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653.

Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. 2019. Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, 66(4):1–42.

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

Sanjoy Dasgupta. 2016. A cost function for similarity-based hierarchical clustering. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 118–127.

Hal Daumé III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 97–104.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982.

Tobias Falke, Christian M. Meyer, and Iryna Gurevych. 2017. Concept-map-based multi-document summarization using concept coreference resolution and global importance optimization. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 801–811.

Spence Green, Nicholas Andrews, Matthew R. Gormley, Mark Dredze, and Christopher D. Manning. 2012. Entity clustering across languages. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 60–69.

Tian Ye He. 2007. *Coreference resolution on entities and events for hospital discharge summaries*. Ph.D. thesis, Massachusetts Institute of Technology.

Yin Jou Huang and Sadao Kurohashi. 2021. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3046–3052.

Stephen C Johnson. 1967. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

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

Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. Resolving event coreference with supervised representation learning and clustering-oriented regularization. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 1–10.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692.

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. *arXiv preprint arXiv:1907.11692*.

Jing Lu and Vincent Ng. 2018. Event coreference resolution: A survey of two decades of research. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5479–5486.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

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.

Ruslan Mitkov, Richard Evans, Constantin Orăsan, Iustin Dornescu, and Miguel Rios. 2012. Coreference resolution: To what extent does it help nlp applications? In *International Conference on Text, Speech and Dialogue*, pages 16–27. Springer.

Vincent Ng and Claire Cardie. 2002. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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.

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

Marc Vilain, John D Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Bishan Yang, Claire Cardie, and Peter Frazier. 2015. A hierarchical distance-dependent bayesian model for event coreference resolution. *Transactions of the Association for Computational Linguistics*, pages 517–528.

Xiaodong Yu, Wenpeng Yin, and Dan Roth. 2020. Paired representation learning for event and entity coreference. *arXiv preprint arXiv:2010.12808*.

A Model and Implementation Details

Implementation Details We train $f_{\text{ind}}^{(t)}$ on training data using code and hyper-parameters from Cattan et al. (2020), and train joint similarity function using sk-learn (Pedregosa et al., 2011) on dev data. We use gold mention spans and document topics during training as well as during inference.

We use a SoTA pre-trained SRL model (Shi and Lin, 2019) to generate semantic role labels for event and entity mention pairs [3]. Since the SRL system used is an end-to-end system and works on top of predicted mention spans, we align the predicted event and entity mentions with gold event and entity mentions. In case a gold event mention span is not detected by the SRL system, we heuristically assign closest left and right occurring entity mention as its `ARGO` and `ARG1` respectively.

[3] Pre-trained model downloaded from `https://demo.allennlp.org/semantic-role-labeling`