Sequential Cross-Document Coreference Resolution

Emily Allaway [*1] **Shuai Wang** [2] **Miguel Ballesteros** [2]
[1] Department of Computer Science, Columbia University, New York, NY
[2] Amazon AI
eallaway@cs.columbia.edu
{wshui, ballemig}@amazon.com

Abstract

Relating entities and events in text is a key component of natural language understanding. Cross-document coreference resolution, in particular, is important for the growing interest in multi-document analysis tasks. In this work we propose a new model that extends the efficient sequential prediction paradigm for coreference resolution to cross-document settings and achieves competitive results for both entity and event coreference while providing strong evidence of the efficacy of both sequential models and higher-order inference in cross-document settings. Our model incrementally composes mentions into cluster representations and predicts links between a mention and the already constructed clusters, approximating a higher-order model. In addition, we conduct extensive ablation studies that provide new insights into the importance of various inputs and representation types in coreference.

1 Introduction

Relating entities and events in text is a key component of natural language understanding. For example, whether two news articles describing hurricanes are referring to the same hurricane event. A crucial component of answering such questions is reasoning about groups entities and events *across multiple* documents.

The goal of coreference resolution is to compute these clusterings of entities or events from extracted spans of text. While within-document coreference has been studied extensively (e.g., Lee et al. (2017, 2018)), there has been relatively less work on the cross-document task. However, growing interest in multi-document applications, such as summarization (e.g., Liu and Lapata (2019); Fabbri et al. (2019)) and reading comprehension (e.g., Yan et al. (2019); Welbl et al. (2018)), highlights the importance of developing efficient and accurate cross-document coreference models to minimize error-propagation in complex reasoning tasks.

In this work we focus on cross-document coreference (CDCR), which implicitly requires within-document coreference (WDCR), and propose a new model that improves both coreference performance and computational complexity. Recent advances in within-document entity coreference resolution have shown that *sequential prediction* (i.e., making coreference predictions from left to right in a text[1]) achieves strong performance (Lee et al., 2017) with lower computational costs. This paradigm is also well suited to real-world streaming settings, where new documents are received every day, since it can easily find corefering events and entities with already processed documents, while most non-sequential models would require a full rerun. In this work, we show how this technique can first be extended to cross-document entity coreference and then adapted to cross-document event coreference.

Our method is also able to take advantage of the history of previously made coreference decisions, approximating a higher-order model (i.e., operating on mentions as well as structures with mentions). Specifically, for every mention, a coreference decision is made not over a set of individual mentions but rather over the *current state of coreference clusters*. In this way, the model is able to use knowledge about the mentions currently in a cluster when making its decisions. While higher-order models have achieved state-of-the-art performance on entity coreference (Lee et al., 2018), they have been used infrequently for event coreference. For example, Yang et al. (2015) use one Chinese restaurant process for WDCR and then a second for CDCR over the within-document clusters. In contrast, our models make within- and cross-document corefer-

[*] Work done during internship at Amazon AI

[1] Note that this is different from mention-pair models, which generally refer to computing scores between all pairs of mentions, regardless of ordering.

ence decisions in a single pass, taking into account all prior coreference decisions at each step.

Our contributions are: (1) we present the first study of sequential modeling for cross-document entity and event coreference and achieve competitive performance with a large reduction in computation time, (2) we conduct extensive ablation studies both on input information and model features, providing new insights for future models.

2 Related Work

Prior work on coreference resolution is generally split into either entity or event coreference. Entity coreference is relatively well studied (Ng, 2010), with the largest focus on within-document coreference (e.g., Raghunathan et al. (2010); Fernandes et al. (2012); Durrett and Klein (2013); Björkelund and Kuhn (2014); Martschat and Strube (2015); Wiseman et al. (2016); Clark and Manning (2016); Lee et al. (2018); Kantor and Globerson (2019)). Recently, Joshi et al. (2019) showed that pre-trained language models, in particular BERT-large (Devlin et al., 2019), achieve state-of-the-art performance on entity coreference. In contrast to prior work on entity coreference, which is primarily sequential (i.e., left to right) and only within-document, our work extends the sequential paradigm to cross-document coreference and also adds incremental candidate composition.

There has been less work on event coreference since the task is generally considered harder. This is largely due to the more complex nature of event mentions (i.e., a trigger and arguments) and their syntactic diversity (e.g., both verb phrases and noun-phrases). Prior work on event coreference typically involves pairwise scoring between mentions followed by a standard clustering algorithm to predict coreference links (Pandian et al., 2018; Choubey and Huang, 2017; Cremisini and Finlayson, 2020; Meged et al., 2020; Yu et al., 2020b; Cattan et al., 2020), classification over a fixed number of clusters (Kenyon-Dean et al., 2018) and template-based methods (Cybulska and Vossen, 2015b,a). While pairwise scoring (e.g., graph-based models, see §3.7) with clustering is effective, it requires tuned thresholds (for the clustering algorithm) and cannot use already predicted scores to inform later ones, since all scores are predicted independently. To the best of our knowledge, our work is the first to apply sequential models to cross-document event coreference.

Although a few previous works attempt to use information about existing clusters through incremental construction (Yang et al., 2015; Lee et al., 2012) or argument sharing (Barhom et al., 2019; Choubey and Huang, 2017), these either continue to rely on pairwise decisions or use shallow, non-contextualized features that have limited efficacy. For example, Xu and Choi (2020) explore a variant of cluster merging for WD entity conreference only that still relies on scores between individual mentions. Additional recent work on WD coreference investigates incremental construction of clusters for prediction (Xia et al., 2020; Toshniwal et al., 2020) and cluster ranking (Yu et al., 2020a) In contrast, our method makes coreference decisions between a mention and all *existing coreference clusters across multiple documents* using contextualized features and so takes advantage of interdependencies between mentions, even across documents, while making all decisions in one pass.

3 Methods

3.1 Overview and Task Definition

We propose a new sequential model for cross-document coreference resolution (see Figure 1) that predicts links between mentions and incrementally constructed coreference clusters computed across multiple documents. In the following sections, we will first describe our model for entity coreference (§3.2-3.5), then discuss adaptations to event coreference (§3.6), and finally conduct a time comparison with prior models (§3.7).

The goal of entity coreference is to determine whether two entity mentions refer to the same real-world entity, with an analogous definition for event coreference. Formally, define an entity mention $x = \langle e, V \rangle$ where e is an entity and V is a set of events in which e participates. We adopt the definition of an event as "a specific occurrence of something that happens" (Cybulska and Vossen, 2014). More specifically $V = [\langle t_1, r_1 \rangle, \ldots, \langle t_n, r_n \rangle]$ where t_i is an event trigger and $r_i \in R$ is the role e takes in in the event with trigger t_i, from a fixed set of argument roles.

3.2 Entity Mention Representation

To construct a representation for entity mention x, we first embed the entity e, along with its context, as h_e using the embeddings from BERT (Devlin et al., 2019) of the start and end sub-word tokens of the entity span. We similarly embed each event

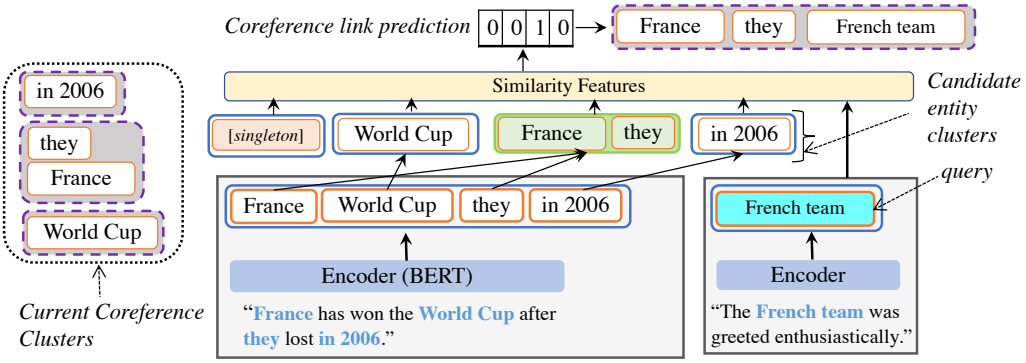

Figure 1: Our model using sequential prediction with incremental clustering for cross-document entity coreference.

$v_i \in V$ as h_{v_i} Then we compute an aggregated event representation h_v using a BiLSTM (Hochreiter and Schmidhuber, 1997) over all of the h_{v_i}, followed by mean-pooling. Finally, we combine the entity representation and event representations using an affine transformation to obtain the full mention representation h_x.

3.3 Incremental Candidate Composition

Let $L^e = \{P_1, \ldots, P_n\}$ be a set of coreference clusters over the antecedents of mention x_i. We compute a candidate cluster representation h_P for each set P of coreferring entity antecedents in L^e. In a similar manner to composition functions in neural dependency parsing, which incrementally combine head-word and modifier information to construct a subtree representation (Dyer et al., 2015, 2016; de Lhoneux et al., 2019), we incrementally combine document- and mention-level information to form a complete candidate cluster representation h_P. That is, for each $x_j \in P$, we combine h_{x_j} and h_{CLS_j}, the CLS token embedding from the document containing x_j, using a non-linear transformation

$$h_{c_j} = \tanh(W_x h_{x_j} + W_{\text{CLS}} h_{\text{CLS}_j} + b_c), \quad (1)$$

where $W_x, W_{\text{CLS}} \in \mathbb{R}^{d_m \times d_m}$ and $b_c \in \mathbb{R}^{d_m}$ are learned parameters. Then we average the representations h_{c_j} for all $x_j \in P$. To allow the model to predict singleton mentions, we add an additional candidate S, with representation $h_s = h_{\text{CLS}}$, to the set of candidates for x_i, where coreference with S indicates x_i is a singleton. As we update L^e, we incrementally update h_P for all $P \in L^e$. Note that L^e can be either the gold coreference clusters over seen mentions (during training) or the current set of predicted clusters (during inference).

3.4 Coreference Link Prediction

We predict coreference links between a query entity mention x and a set of candidates by passing a set of similarity features through a softmax layer. Let $\mathcal{C}_x = \{h_{P_1}, \ldots, h_{P_m}\}$ be the set of m candidate representations (including h_s) for x.

We first compute the similarity between each candidate P_j and the query using both cosine similarity f_{cos} and multi-perspective cosine (MP cosine) similarity f_{mpcos} (Wang et al., 2017). For multi-perspective cosine similarity, we first project the candidate and query into k shared spaces using k separate linear projections. Then, for each of the k new spaces, we compute the cosine similarity between the projected representations in that space.

Next, we combine these features with the product and the difference of the candidate and query to obtain the final feature representation:

$$h_f^{(j)} = [h_x \cdot h_{P_j}; |h_x - h_{P_j}|; f_{cos}; f_{mpcos}]. \quad (2)$$

Then, for all candidates P_j we compute the probability $p(x, P_j)$ that the query x corefers with as

$$p(x \text{ corefers with } P_j) = softmax(W_o h_f + b_o).$$

We predict a link between x and the candidate with maximum coreference probability. If that candidate is S, then x is predicted as a singleton.

3.5 Sequential Cross-Document Prediction

To predict cross-document coreference links, we propose an algorithm that iterates through a list of documents and predicts coreference links between entity mentions in the *current document* and candidate clusters computed across all *preceeding documents* (Figure 2).

We first impose an arbitrary ordering on a list of documents D. Then, for each $i \in \{1, \ldots, |D|\}$ and each entity mention x_n in document $D[i]$ we compute candidates clusters $\mathcal{C}_{x_n}$ (§3.3) from the

Algorithm 1 Training

Require: D: an ordered list of documents
M^e, M^v: gold entity and event mentions
T: set of document topic clusters
G^e, G^v: gold entity and event clustering

1: $L^e \leftarrow []$ (predicted entity clustering)
2: **for** $i \in \{1, \ldots, |D|\}$ **do**
3: $M_i \leftarrow []$
4: **for** $j < i$ **do**
5: **if** $D[i]$ and $D[j]$ have the same topic in T **then**
6: $M_i \leftarrow M_i + \{h_x : x \in D[j] \cap M^e\}$
7: **end if**
8: **end for**
9: $C_x \leftarrow$ ComputeCandidates(M_i, G^e, M^v)
10: **for all** x s.t. $x \in D[i] \cap M^e$ **do**
11: $\{h_f^{(k)}\}_{k=1}^{|C_x|} \leftarrow$ ComputeFeatures(C_x)
12: $\ell \leftarrow$ PredictCoreferenceLink$\left(\{h_f^{(k)}\}_{k=1}^{|C_x|}\right)$
13: $M_i \leftarrow M_i + x$
14: $L^e \leftarrow$ UpdatePredictedClusters(ℓ, x, L^e)
15: $C_x \leftarrow$ ComputeCandidates(M_i, G^e, M^v)
16: **end for**
17: **end for**
18: **return** L^e

Figure 2: Algorithm for coreference resolution with sequential prediction and incremental clustering (§3.3).

coreference clusters across all documents $D[j]$ where $j < i$. Note that this includes both within-document and cross-document clusters.

After computing the candidate clusters for entity mention x_n, we compute similarity features and use these to predict a coreference link ℓ_n between x_n and one candidate $P_j \in \mathcal{C}_{x_n}$ (§3.4). Finally, we update the predicted clustering to account for the new link ℓ_n and compute new candidates for x_{n+1}.

Since the number of possible candidates for each x_n grows as the number of preceding documents (i) increases, we reduce the computational cost by only considering previous documents $D[j]$ that are similar to $D[i]$. We define similar as having the same topic from a fixed set of topics T.

During training, we use gold entity clusters to compute the candidates (as in Figure 2) and gold document topic clusters T. In contrast, during inference we using the currently predicted coreference clusters to compute candidates. That is, we use L^e in place of G^e in lines 9 and 15 in Figure 2. Furthermore, we use predicted topic clusters $\hat{T}$, computed using K-means (§4.5), in place of T.

Our model is trained to minimize cross-entropy loss computed in batches. Here, all M entity mentions in a single document form one batch and the loss is computed after M sequential predictions.

3.6 Adaptations for Event Coreference

We also adapt the same architecture and algorithm to cross-document event coreference resolution. Define an event $x = \langle t, A \rangle$ where t is the event trigger and $A = [\langle e_1, r_1 \rangle, \ldots, \langle e_m, r_m \rangle]$ is the set of its event arguments (i.e., entity-role pairs). If no entity takes some role r_i, then $e_i = \emptyset$.

We compute the event representation h_x analogously to the entity representations (§3.2). That is, we combine the event trigger representation with an aggregated entity representation, computed over event arguments A. We then compute candidate-clusters and predict coreference links in the same manner as for entities (§3.3, §3.4) with an additional feature, indicating whether event arguments corefer, in Equation 2.

Under the definition of event coreference, two events corefer when both their triggers and all of their arguments corefer. In practice, we relax the second requirement to *most* of their arguments, since argument role labeling may be noisy. We compute a binary feature for g_{r_l} for each argument role r_l to indicate the coreference of e_l (the entity with role r_l in x) and $e_l^{(P_j)}$ (the entity with role r_l in candidate cluster P_j). We compute a feature only for roles $r_l \in R$ in which both the candidate and the query have some entity present ($e_l^{(i)} \neq \emptyset$ and $e_l^{(P_j)} \neq \emptyset$). Then, for each $r_l \in R$, if the two entities corefer then $g_{r_l} = 1$ and if they do not corefer then $g_{r_l} = 0$. Finally, we map each g_{r_l} to a learned embedding $f_{r_l} \in \mathbb{R}^{d_f}$ and compute an aggregated argument feature representation $f_r = \frac{1}{|R_{\neq 0}|} \sum_{r_l \in R_{\neq 0}} f_{r_l}$ where $R_{\neq 0}$ is the set of roles filled in both x and P_j. This feature is then concatenated into Equation 2 before prediction.

The cross-document iteration algorithm for event coreference is analogous to Figure 2 with the modification that ComputeFeatures (line 11) now also takes the gold entity coreference clusters G^e.

3.7 Time Comparison with Prior Methods

Algorithms for coreference resolution fall into two paradigms: sequential models (i.e., left to right prediction) and graph-based models (i.e., finding optimal connected components from a graph of pairwise similarity scores). This dichotomy is analogous to that in dependency parsing between transition-based parsers (i.e., greedy left-to-right

models) and graph-based parsers. While the differences between the paradigms have been studied for dependency parsing (McDonald and Nivre, 2007, 2011), comparisons for coreference have been limited to WD entity coreference only (Martschat and Strube, 2015). In part, this is due to the usages of the two paradigms; sequential models are primarily used for WDCR while graph-based models are used for CDCR. However, as in dependency parsing, the sequential models can be made much more computationally efficient than the graph-based models as we show with our model.

Let D be a set of documents with m mentions that form c coreference clusters. In a *graph-based* model, scores are computed between all pairs of mentions in all documents and in a *general sequential* model scores are computed between a specific mention and all antecedents. Our model is a *higher-order sequential* model; it combines the general sequential paradigm with higher-order inference through incremental candidate clustering. Therefore, while both general sequential and graph-based models *always* require computing $\sim m^2$ scores, our model only needs to compute cm scores. Since in practice $c \ll m$, our model is more efficient.

We also note that graph-based models require an additional step to compute clusters using pairwise scores. In practice, agglomerative clustering is often used but for an arbitrary distance matrix this is $\mathcal{O}(m^3)$. In contrast, a higher-order sequential model computes clusters *simultaneously* with scores, alleviating the need for an additional step, and therefore is substantially more efficient.

These improvements in efficiency are even more important in real-world streaming scenarios. Namely, given a known set of clusters C for the document set D, compute coreference with the mentions in a new document. In both a general sequential model and a graph-based model, scores need to be computed between the new mentions, and *all* mentions in D. However, our model only needs to compare the new mentions to the existing *clusters* C. Hence, our model can better handle the temporal-component of many usage settings and is better suited to life-long learning.

4 Experimental Setup

4.1 Data

We conduct experiments using the ECB+ dataset (Cybulska and Vossen, 2014), the largest available dataset for both within-document and cross-document event and entity coreference. The ECB+ dataset is an extension of the Event Coreference Bank dataset (ECB) (Bejan and Harabagiu, 2010), which consists of news articles clustered into topics by seminal events (e.g., "6.1 earthquake Indonesia 2009"). The extension of ECB adds an additional seminal event to each topic (e.g., "6.1 earthquake Indonesia 2013"). Documents on each of the two seminal events then form subtopic clusters within each topic in ECB+.

	Train	Dev	Test
# Topics	25	8	10
# Subtopics	50	16	20
# Documents	574	196	206
# Event Mentions	3808	1245	1780
# Entity Mentions	4758	1476	2055
# Event Clusters	1527	409	805
# Entity Clusters	1286	330	608

Table 1: Data Statistics for ECB+ corpus. Topics: train $\{1, 3, 4, 6\text{-}11, 13, 14, 16, 19\text{-}20, 22, 24\text{-}33\}$, development $\{2, 5, 12, 18, 21, 23, 34, 35\}$. and test 36-45

Following the recommendations of Cybulska and Vossen (2015b), we use only the subset of annotations that have been validated for correctness in our experiments (see Table 1). As a result, our results are comparable to recent studies (e.g., Barhom et al. (2019); Kenyon-Dean et al. (2018); Meged et al. (2020)) but not earlier methods (see Upadhyay et al. (2016) for a more complete overview of evaluation settings). We use the standard partitions of the dataset into train, development and test split by topic and use subtopics (gold or predicted) for document clustering.

4.2 Identifying Event Structures

The ECB+ dataset does not include relations between events and entities. Although prior work used the Swirl (Surdeanu et al., 2007) semantic role labeling (SRL) parser to extract predicate-argument structures, this does not take advantage of recent advances in SRL. In fact, prior works on coreference using ECB+ have added a number of additional rules on top of the parser output to improve its coverage and linking. For example, Barhom et al. (2019) used a dependency parser to identify additional mentions. Therefore, in this work we use the current state-of-the-art SRL parser on the standard CoNLL-2005 shared task (He et al., 2018), which has improved performance by $\sim$10 F1 points both in- and out-of-domain.

Following prior work, we restrict the event struc-

ture to the following four argument roles: ARG0, ARG1, TIME, and LOC. However, we additionally add a type constraint during pre-processing that requires entities of type TIME and LOC only fill matching roles (TIME and LOC respectively).

4.3 Domain Adaptive Pre-training

Since BERT was trained on the BooksCorpus and Wikipedia (Devlin et al., 2019) and the ECB+ dataset contains news articles, there is a domain mismatch. In addition, the use of a domain corpus for pre-training helps address the data scarcity issue for events and entities, indicated by Ma et al. (2020). Therefore, before training our coreference models, we first fine-tune BERT using the English Gigaword Corpus[2] with both BERT losses, as this has been shown to be effective for domain transfer (Gururangan et al., 2020). Following Ma et al. (2020), we randomly sample $50k$ documents ($626k$ sentences) and pre-train for $10k$ steps, using the hyperparameter settings from Devlin et al. (2019).

4.4 Baselines and Models

We experiment with the following baseline variations of our model: **BERT-SeqWD** – computes coreference scores using only the entity (or event) representations, without any cross-document linking, and **BERT-SeqXdoc** – computes coreference scores across documents but without candidate composition. This means the baseline BERT-SeqXdoc computes scores between the query mention and *all* antecedent mentions across *all* prior documents, rather than between the query and the clusters computed with candidate composition. For both event and entity coreference we experiment with our model, **SeqXdoc+IC** with (**+Adapt**) and without adaptive pre-training.

For entity coreference we compare against the following models:

- Barhom et al. (2019) (Bh2019) – graph-based model that alternates between event and entity coreference, updating the argument features for events (and event features for entities) after each iteration,
- Cattan et al. (2020) – general sequential WDCR model from Lee et al. (2017) paired with agglomerative clustering
- Caciularu et al. (2021) – graph-based Bh2019 model using representations from a Longformer (Beltagy et al., 2020) with cross-document attention during pre-training
- Lemma – a strong baseline that links mentions with the same head-word lemma in the same document topic cluster (Barhom et al., 2019).

For event coreference we additionally compare to:

- Meged et al. (2020) – graph-based Bh2019 model with an additional paraphrase-based feature,
- Cremisini and Finlayson (2020) – graph-based prediction of coreference scores over four types of similarity features,
- three graph-based representation learning models with agglomerative clustering – Kenyon-Dean et al. (2018), Yu et al. (2020b) and Zeng et al. (2020).

4.5 Implementation Details

Our models are tuned for a maximum of 80 epochs with early-stopping on the development set (using CoNLL F1) with a patience of 20 epochs. All models are optimized using Adam (Kingma and Ba, 2015) with a learning rate of $2e$-5 and treat all mentions in a document as a batch. We clip gradients to 30 to prevent exploding gradients. Document ordering is fixed within a epoch but randomized between epochs.

We encode each document using BERT-base and a maximum document length of 600 tokens for BERT. A threshold of 600 is the default for our system that also accommodates longer documents and there are no documents over 512 tokens in ECB+. In our system, long documents will not be truncated but rather will be split into multiple document pieces, that will be merged in our algorithm. Following adaptive pre-training, we do not fine-tune BERT.

To encode arguments/events we use an LSTM with hidden size 128, for the argument coreference features we use two learned embeddings of dimension $d_f = 50$, and for the multi-perspective cosine similarity we use $k = 1$ projection layers with dimension 50 for entity coreference and $k = 3$ projection layers with the same dimension for event coreference. We do not tune our hyperparameters.

We follow Barhom et al. (2019) and use K-means to compute document clusters for inference from their implementation with $K = 20$. Specifically, as features, we use the TF-IDF scores of

[2]https://catalog.ldc.upenn.edu/LDC2011T07

unigrams, bigrams, and trigrams in the unfiltered dataset, excluding stop words. We select $K = 20$ as this is the number of gold document clusters in the test data but this can be modified without affecting our algorithm.

5 Results and Analyses

5.1 Coreference Resolution

We evaluate using the standard metrics for coreference resolution: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF-e (Luo, 2005), and CoNLL F1 (their average).

For entity coreference, our model outperforms most prior methods (see Table 2) and for event coreference our model demonstrates strong performance (see Table 3). Although Caciularu et al. (2021) achieve the highest performance, we observe first that much of their gains come from the use of a Longformer (80.4 CoNLL F1 for entities, 84.6 for events) – a language-model specifically designed for long contexts. Additionally, they fine-tune with coreference specific data and special tokens, neither of which our models use. As observed by Xu and Choi (2020), improvements in the underlying language model can result in large-gains in coreference performance without requiring algorithmic improvements. However, our model focuses on improving the coreference prediction algorithm, while using a standard BERT language model.

We observe that our algorithm provides large gains for both event and entity coreference. In particular, while naive applications of the sequential paradigm in the cross-document setting (BERT-SeqWD and BERT-SeqXdoc) perform poorly, the addition of incremental candidate clustering even without adaptive pre-training yields competitive results ($+8.9$ and $+2.8$ CoNLL F1 for entities and events respectively). Adaptive pre-training, which handles domain-mismatch in a similar way to task-specific fine-tuning (e.g., as in Caciularu et al. (2021)), provides further gains (4.1 and 2.2 CoNLL F1 for entities and events respectively). Our results highlight the importance of higher-order inference (e.g., composition) when extending sequential prediction to cross-document settings.

We note that prior work (Barhom et al., 2019) used predicted entity coreference clusters. In a comparable setting, using the output from our best entity coreference model to compute argument coreference features (§3.4), we do not observe any drop in performance (i.e., the performance is identical). In addition, we use predicted document clusters for our experiments on both entity and event coreference. Due to the high-quality document clustering[3], we only observe a drop of $\sim$1 CoNLL F1 point when using these predicted clusters, compared to the gold document clusters. However, we note that such a small decrease relies on the quality of the clustering, as shown by the larger gap (3 F1 points) observed by Cremisini and Finlayson (2020) with less accurate clusters.

Finally, we observe that the sequential paradigm with incremental candidate composition has several additional advantages. First, without candidate composition, sequential coreference resolution is typically a multi-label task. However, *with* composition, each mention now has exactly one correct coreferring antecendent during training (possibly a cluster rather than an individual mention) and this simplifies the learning task. While heuristics (e.g., choosing the closest antecedent as the gold antecendent, discussed in Martschat and Strube (2015)), also convert the task from multi- to single-label, they are problematic because they limit the amount of information that can be used during training. Additionally, candidate composition allows sequential models to make use of information not only about antecedents (in contrast to the graph-based models) but also about prior coreference decisions (in contrast to non-compositional sequential models).

Our results are consistent with prior work on the efficacy of sequential models (cf. mention-ranking models for WD entity coreference) (Martschat and Strube, 2015) and the importance of higher-order inference mechanisms (e.g., incremental candidate clustering) in cross-document tasks (Zhou et al., 2020). In addition, our results demonstrate the importance of algorithmic improvements, in addition to improvements in the underlying language model, for strong coreference performance.

5.2 Feature Ablation

Since mention representations in coreference vary widely, we conduct extensive feature ablations to provide insights for future work (see Table 4).

First we examine the vector representations used to encode mentions. While prior work used ELMo and pre-trained GloVE (Pennington et al., 2014) word and character embeddings, recent models use

[3]Homogeneity: 0.977, Completeness: .980, V-measure: .978, Adjusted Random-Index: .945

	MUC			B^3			CEAF-e			
	P	R	F1	P	R	F1	P	R	F1	C-F1
Lemma	71.3	83	76.7	53.4	84.9	65.6	70.1	52.5	60.0	67.4
Barhom et al. (2019)	78.6	80.9	79.7	65.5	76.4	70.5	65.4	61.3	63.3	71.2
Cattan et al. (2020)	85.7	81.7	83.6	70.7	74.8	72.7	59.3	67.4	63.1	73.1
Caciularu et al. (2021)	**88.1**	**91.8**	**89.9**	**82.5**	**81.7**	**82.1**	**81.2**	**72.9**	**76.8**	**82.9**
BERT-SeqWD + Adapt	78.0	39.2	52.2	89.6	34.5	49.8	34.9	76.1	47.9	50.0
BERT-SeqXdoc + Adapt	80.2	69.8	74.6	76.6	54.2	63.5	49.6	64.8	56.2	64.8
SeqXdoc+IC	83.6	81.5	82.5	76.0	66.7	71.1	65.7	69.3	67.4	73.7
+ Adapt	83.9	84.7	84.3	74.5	70.5	72.4	70.0	68.1	69.2	75.3

Table 2: Entity coreference on the ECB+ test set, combined within- and cross-document scores using predicted document clusters. C-F1 is CoNLL F1. **Bold** indicates best overall, underline indicates our best model.

	MUC			B^3			CEAF-e			
	P	R	F1	P	R	F1	P	R	F1	C- F1
Lemma	76.5	79.9	78.1	71.7	85.0	77.8	75.5	71.7	73.6	76.5
Kenyon-Dean et al. (2018)	67.0	71.0	69.0	71.0	67.0	69.0	71.0	67.0	69.0	71.0
Barhom et al. (2019)	77.6	84.5	80.9	76.1	85.1	80.3	81.0	73.8	77.3	79.5
Cremisini and Finlayson (2020)	**89.4**	84.9	87.1	74.3	69.2	71.6	49.6	60.7	54.6	71.1
Meged et al. (2020)	78.7	84.7	81.6	75.9	85.9	80.5	81.1	74.8	77.8	80.0
Cattan et al. (2020)	85.1	81.9	83.5	82.1	82.7	82.4	75.2	78.9	77.0	81.0
Yu et al. (2020b)	88.1	85.1	86.6	**86.1**	84.7	85.4	79.6	**83.1**	81.3	84.4
Zeng et al. (2020)	85.6	**89.3**	87.5	77.6	**89.7**	83.2	**84.5**	80.1	**82.3**	84.3
Caciularu et al. (2021)	87.1	89.2	**88.1**	84.9	87.9	**86.4**	83.3	81.2	82.2	**85.6**
BERT-SeqWD + Adapt	68.9	28.9	40.7	91.1	48.5	63.3	49.3	83.9	62.1	55.4
BERT-SeqXdoc + Adapt	82.2	66.8	73.7	84.2	66.8	74.5	65.9	80.8	72.6	73.6
SeqXdoc+IC	81.6	85.9	83.7	69.5	80.6	74.4	75.3	67.1	71.0	76.4
+ Adapt	81.7	82.8	82.2	80.8	81.5	81.1	79.8	78.4	79.1	80.8

Table 3: Event coreference on the ECB+ test set, combined within- and cross-document scores using predicted document clusters. C-F1: is CoNLL F1. **Bold** indicates best overall, underline indicates our best model.

	Entity		Event	
	F1	Δ	F1	Δ
Our Model	75.3		80.8	
− Coref feat (§3.6)	-	-	79.6	-1.2
− Args (§3.2)	74.8	-0.9	78.7	-2.1
− Arg comp (§3.2)	74.6	-0.7	78.3	-2.5
− CLS (Eq. 1)	74.5	-0.8	78.9	-1.9
− MP cosine (§3.4)	74.5	-0.8	79.1	-1.7
+ GloVE	70.1	-5.2	76.7	-4.1
+ RoBERTa	71.2	-4.1	78.1	-2.7

Table 4: Feature ablation results (CoNLL F1) on the ECB+ test set. For entity coreference arguments (Args) are events, for event coreference they are entities.

RoBERTa (Cattan et al., 2020; Yu et al., 2020b). We experiment with replacing BERT-base with RoBERTa-base and with using GloVE in addition BERT in our models (see Appendix B for implementation) and observe large drops in performance. We hypothesize that the substantial performance difference between BERT and RoBERTa is due to the Next Sentence Prediciton (NSP) used to train BERT but not RoBERTa. The NSP may force BERT to learn attention multiple sentences, and therefore to understand the document as a whole, an ability that is important for coreference resolution. Therefore, we hypothesize that without task-specific fine-tuning, adaptive pre-training is most beneficial for coreference on ECB+.

We also observe that our entity coreference model is relatively less susceptible to feature changes than the event coreference model. For example, the event coreference model is particularly reliant on the argument features. Both replacing the argument composition BiLSTM with a mean-pooling operation (−Arg comp) and removing all argument information (−Args) result in large drops in performance (-2.5 and -2.1 respectively).

Finally, the contribution of the multi-perspective cosine similarity underscores the importance of cosine similarity as observed by Cremisini and Finlayson (2020). These ablations, including on the importance of document-level information (−CLS) suggest new directions for token and docu-

	Entity		Event	
	F1	Δ	F1	Δ
Our Model	75.3		80.8	
HeSRL - C	75.3	-0.0	80.4	-0.4
HeSRL + BhR + C	74.9	-0.4	79.2	-1.6
Swirl + BhR + C	75.4	+0.1	80.0	-0.8
Swirl + BhR	75.4	+0.1	78.7	-2.1

Table 5: Ablation results (CoNLL F1) on methods for identifying event structures on ECB+ test set. HeSRL is He et al. (2018), BhR is additional rules for aligning the SRL and annotations from (Barhom et al., 2019), C is entity type constraint (see §4.2).

ment representations in coreference.

5.3 Effects of SRL

We investigate the impact of using a recent SRL parser to extract event structures (§4.2), compared to the Swirl parser used in prior work (see Table 5).

We first observe that the additional extraction rules used in Barhom et al. (2019) are not necessary when using the new SRL parser. In fact, these rules actually result in a decrease in performance for both entity and event coreference (-1.6 and -0.4 respectively). In addition, when using the Swirl parser and additional rules (Swirl+Bh-rules), we observe a large drop for event coreference (-2.1) compared to entity coreference. This aligns with the heavier dependence of event coreference models on arguments (§ 5.2), which will lead to greater model sensitivity to errors in the entity-event structures (from the SRL). Furthermore, we also see that the type constraint improves event coreference more when using the Swirl SRL ($\Delta = 1.3$) than when using the new SRL ($\Delta = 0.4$). Note that because we do not use role information for entity coreference (i.e., no argument coreference feature), adding or removing the type constraint does not affect entity coreference. These results highlight the importance of minimizing error propagation from the SRL into the coreference resolution.

6 Conclusion

In this paper, we propose a new model for cross-document coreference resolution that extends the efficient sequential prediction paradigm to multiple documents. The sequential prediction is combined with incremental candidate composition that allows the model to use the history of past coreference decisions at every step. Our model achieves competitive results for both entity and event coreference and our analysis provides strong evidence of the efficacy of both sequential models and higher-order inference in cross-document settings. In future, we intend to adapt this model to coreference across document streams and investigate alternatives to greedy prediction (e.g., beam search).

References

A. Bagga and B. Baldwin. 1998. Algorithms for scoring coreference chains.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and I. Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *ACL*.

C. Bejan and Sanda M. Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *ACL*.

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

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 47–57.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew E. Peters, Arie Cattan, and Ido Dagan. 2021. Cross-document language modeling. *ArXiv*, abs/2101.00406.

Jie Cai and Michael Strube. 2010. Evaluation metrics for end-to-end coreference resolution systems. In *SIGDIAL Conference*.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and I. Dagan. 2020. Streamlining cross-document coreference resolution: Evaluation and modeling. *ArXiv*, abs/2009.11032.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *EMNLP*.

Kevin Clark and Christopher D Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262.

Andres Cremisini and Mark A. Finlayson. 2020. New insights into cross-document event coreference: Systematic comparison and a simplified approach. In *NUSE*.

A. Cybulska and Piek T. J. M. Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *LREC*.

A. Cybulska and Piek T. J. M. Vossen. 2015a. "bag of events" approach to event coreference resolution. supervised classification of event templates. *Int. J. Comput. Linguistics Appl.*, 6:11–27.

A. Cybulska and Piek T. J. M. Vossen. 2015b. Translating granularity of event slots into features for event coreference resolution. In *EVENTS@HLP-NAACL*.

Miryam de Lhoneux, Miguel Ballesteros, and Joakim Nivre. 2019. Recursive subtree composition in lstm-based dependency parsing. *ArXiv*, abs/1902.09781.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Greg Durrett and D. Klein. 2013. Easy victories and uphill battles in coreference resolution. In *EMNLP*.

Chris Dyer, Miguel Ballesteros, W. Ling, A. Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *ACL*.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *HLT-NAACL*.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Eraldo Fernandes, Cicero dos Santos, and Ruy Luiz Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 41–48.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *ACL*.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Mandar Joshi, Omer Levy, Daniel S. Weld, and Luke Zettlemoyer. 2019. Bert for coreference resolution: Baselines and analysis. In *EMNLP/IJCNLP*.

Ben Kantor and A. Globerson. 2019. Coreference resolution with entity equalization. In *ACL*.

Kian Kenyon-Dean, J. Cheung, and Doina Precup. 2018. Resolving event coreference with supervised representation learning and clustering-oriented regularization. *ArXiv*, abs/1805.10985.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

H. Lee, M. Recasens, Angel X. Chang, M. Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *EMNLP-CoNLL*.

Kenton Lee, Luheng He, M. Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *EMNLP*.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *NAACL-HLT*.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *HLT/EMNLP*.

Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros, and Yaser Al-Onaizan. 2020. Resource-enhanced neural model for event argument extraction. *EMNLP*, abs/2010.03022.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418.

R. McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *EMNLP-CoNLL*.

R. McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37:197–230.

Y. Meged, Avi Caciularu, Vered Shwartz, and I. Dagan. 2020. Paraphrasing vs coreferring: Two sides of the same coin. *ArXiv*, abs/2004.14979.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1396–1411.

A. Pandian, Lamana Mulaffer, K. Oflazer, and A. AlZeyara. 2018. Event coreference resolution using neural network classifiers. *ArXiv*, abs/1810.04216.

Jeffrey Pennington, R. Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher D Manning. 2010. A

multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501.

M. Surdeanu, Lluís Màrquez i Villodre, X. Carreras, and P. Comas. 2007. Combination strategies for semantic role labeling. *Journal of Artificial Intelligence Research*, 29.

Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. 2020. Learning to ignore: Long document coreference with bounded memory neural networks. In *EMNLP*.

Shyam Upadhyay, Nitish Gupta, Christos Christodoulopoulos, and D. Roth. 2016. Revisiting the evaluation for cross document event coreference. In *COLING*.

Marc B. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *MUC*.

Z. Wang, W. Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*.

Johannes Welbl, Pontus Stenetorp, and S. Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Sam Wiseman, Alexander M Rush, and Stuart M Shieber. 2016. Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*.

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental neural coreference resolution in constant memory. In *EMNLP*.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *EMNLP*.

Ming Yan, Jiangnan Xia, Chen Wu, B. Bi, Z. Zhao, Ji Zhang, L. Si, Rui Wang, W. Wang, and Haiqing Chen. 2019. A deep cascade model for multi-document reading comprehension. In *AAAI*.

B. Yang, Claire Cardie, and P. Frazier. 2015. A hierarchical distance-dependent bayesian model for event coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:517–528.

Juntao Yu, Alexandra Uma, and Massimo Poesio. 2020a. A cluster ranking model for full anaphora resolution. In *LREC*.

Xiaodong Yu, Wenpeng Yin, and D. Roth. 2020b. Paired representation learning for event and entity coreference. *ArXiv*, abs/2010.12808.

Yutao Zeng, Xiaolong Jin, Saiping Guan, J. Guo, and Xueqi Cheng. 2020. Event coreference resolution with their paraphrases and argument-aware embeddings. In *COLING*.

Xuhui Zhou, Nikolaos Pappas, and Noah A. Smith. 2020. Multilevel text alignment with cross-document attention. In *EMNLP*.

A Implementation details

The dataset is available here: http://www.newsreader-project.eu/results/data/the-ecb-corpus/.

Our models have approximately 9million parameters and are trained with one Tesla V100-SXM2 GPU.

We evaluate our models using three coreference metrics. MUC counts discrepencies in links between the gold and predicted clusters (and thus ignores singletons). B^3 computes, for each mention m, the difference between the gold cluster containing m and the predicted cluster containing m. Finally, CEAF-e finds the injective alignment between predicted and gold clusters that gives the highest similarity under a defined function. For more details on metrics, refer to Cai and Strube (2010).

We report validation results for both entity (Table 6) and event coreference (Table 7).

B Feature Ablation

We experiment with 300-dimensional pre-trained GloVE (Pennington et al., 2014) embeddings in our model. Following (Barhom et al., 2019), we use both GloVE and BERT in the entity mention representations (for entity coreference) and event trigger representations (for event coreference). In the argument representations we use only GloVE.

Let $x = \langle e, V \rangle$ be an entity in document d and let $s_{d_1}, \ldots, s_{d_m}$ be the static GloVE embeddings for the tokens in d. First we apply a non-linear transformation to each $\widetilde{s_{d_i}} = \tanh(W_t s_{d_i} + b_t)$ where $W_t \in \mathbb{R}^{1536 \times 300}$, where 1536 is 2$*$the dimension of the BERT embeddings (2 because we use the start and end tokens of a mention) and 300 is the dimension of GloVE embeddings. Then, we take the average of $\widetilde{s_{d_i}}$ to obtain $s_{\text{CLS}} \in \mathbb{R}^{1536}$, a static document representation. Next we extract the representation for the entity x as in section 3.2, s_x. Finally, we combine these representations with the BERT representations

$$
\begin{aligned}
z_x &= \tanh(W_x^s s_x + W_x^B h_x + b_x) \\
z_{\text{CLS}} &= \tanh(W_{\text{CLS}}^s s_{\text{CLS}} + W_{\text{CLS}}^B h_x + b_{\text{CLS}}) \\
h_c &= W_x z_x + W_{\text{CLS}} z_{\text{CLS}} + b_c
\end{aligned}
$$

where h_c is the representation (as in § 1) and $W_x^s, W_{\text{CLS}}^s, W_x^B, W_{\text{CLS}}^B \in \mathbb{R}^{1536 \times 1536}$ and $b_x, b_{\text{CLS}}, b_c \in \mathbb{R}^{1536}$ are learned parameters.

	MUC	B^3	CEAF-e	CoNLL F1
BERT-Rep	43.4	40.0	35.9	39.8
BERT-Rep-Xdoc	82.7	67.6	56.1	68.8
SeqXdoc	89.1	76.0	71.1	78.7
+ Adapt	90.2	77.5	71.7	79.8

Table 6: Entity coreference F1 on ECB+ dev set, combined within- and cross-document scores using predicted document clusters.

	MUC	B^3	CEAF-e	CoNLL F1
BERT-Rep	32.2	50.6	47.5	43.4
BERT-Rep-Xdoc	78.4	74.6	65.0	72.7
SeqXdoc	84.8	79.1	74.4	79.4
+ Adapt	85.8	81.1	73.7	80.2

Table 7: Event coreference F1 on ECB+ dev set, combined within- and cross-document scores using predicted document clusters.