Pairwise Representation Learning for Event Coreference

Xiaodong Yu[1] **Wenpeng Yin**[2] **Dan Roth**[1]
[1]University of Pennsylvania [2]Temple University
{xdyu, danroth}@seas.upenn.edu wenpeng.yin@temple.edu

Abstract

Natural Language Processing tasks such as resolving the coreference of events require understanding the relations between two text snippets. These tasks are typically formulated as (binary) classification problems over independently induced representations of the text snippets. In this work, we develop a Pairwise Representation Learning (PairwiseRL) scheme for the event mention pairs, in which we jointly encode a pair of text snippets so that the representation of each mention in the pair is induced in the context of the other one. Furthermore, our representation supports a finer, structured representation of the text snippet to facilitate encoding events and their arguments. We show that PairwiseRL, despite its simplicity, outperforms the prior state-of-the-art event coreference systems on both cross-document and within-document event coreference benchmarks. We also conduct in-depth analysis in terms of the improvement and the limitation of pairwise representation so as to provide insights for future work. [1]

1 Introduction

In this work, we study the event coreference resolution problem. Event coreference resolution is commonly modeled as a binary classification problem over independently induced representations on the text snippets of each event mention (Lee et al., 2012; Barhom et al., 2019).[2] Understanding the relations between two text snippets is the essential part in the tasks. In this work, we argue that the representations of prior work are not expressive enough to learn the pairwise relations due to the following two reasons:

(i) *Counterpart Unawareness*. The relationship between two mentions can be different in different contexts. To address different scenarios, it is better for each mention to ensure that its representation is aware of what its counterpart's representation. However, most early work induces mention representations independently by extracting features only from the sentence that contains the mention, without using the context of the other mention (Barhom et al., 2019; Huang et al., 2019). Some more recent work attempts to encode the whole document to represent each mention (Lee et al., 2017; Cattan et al., 2020). This is beneficial for short documents, since the representation of each mention will also include information from the context of the other candidate mention. However, this is not sufficient for cross-document settings, when the comparison is, for example, between two event mentions that appear in separate documents. In this case even encoding large pieces of text leave the candidate mention representations independent of each other.

(ii) *Unstructured representation learning*. An event mention consists of multiple arguments that describe the event: who, when, where, etc. When determining the relationship of two event mentions, the mismatch of some arguments could be decisive. Consider the following two sentences s_1 and s_2 (event trigger is **underlined**; argument #0 is in blue, location is in purple)

s_1: "Over 69,000 people **lost** their lives in the quake, including 68,636 in Sichuan."
s_2: "Up to 6,434 people **lost** their lives in Kobe earthquake and about 4,600 of them were from Kobe."

These two events "lost" are not the same events because the earthquake in Sichuan and the earthquake in Kobe are two different earthquakes, and Sichuan and Kobe do not have any geographic overlap. The mismatch of the locations "Sichuan" and "Kobe" may be enough to determine that the two events are different from each other without even considering the rest of the sentence. Most prior

[1] Our code is available at http://cogcomp.org/page/publication_view/979

[2] Some work maps the two mentions into a single matching score, e.g., (Barhom et al., 2019); this can be treated as a special case of binary classification.

Proceedings of the 11th Joint Conference on Lexical and Computational Semantics, pages 69 - 78
July 14-15, 2022 ©2022 Association for Computational Linguistics

work encodes all of the arguments into a single distributed representation vector and just compares the overall vector representations of two mention triggers. Although contextual representation could encode all of the arguments' information, this is less optimal than explicitly representing all of the arguments, thus making it easier for the model to conduct fine-grained reasoning over each of the argument.

To address the drawbacks of prior representations, we propose *pairwise representation learning* (PAIRWISERL). PAIRWISERL alleviates the aforementioned two limitations with two designs:

Pairwise representation learning. We suggest treating a mention pair, rather than a single mention, as the object for the representation learning. We encode the two mentions' sentences as a whole sequence so that one sentence's token representation is able to interact with the other sentence's from the very beginning. This is advantageous over learning two separate and independent representations because it allows for learning how compatible one mention is with the other mention's context.

Structured representation learning. The observation that mismatching arguments are critical to making the coreference decision indicates that using a single combined representation for all of the arguments could be less informative for cross-mention comparison. In this work, we explicitly represent all the arguments, and compare each argument separately.

To our knowledge, this is the first work that applies pairwise representation learning to event coreference problems. We report our performance on both within-document and cross-document event coreference benchmarks. We show that PAIRWISERL, despite its simplicity, clearly surpasses more complex state-of-the-art event coreference systems on two most popular benchmarks ECB+ (Cybulska and Vossen, 2014) and KBP17 (Getman et al., 2015). We also conduct in-depth analysis in terms of the improvement and the limitation of pairwise representation so as to provide insights for future work.

2 Related Work

In this section, we discuss prior representation learning approaches for event coreference and how pairwise representation learning has been used in other NLP problems.

Event Coreference. Earlier work uses hand-engineered event features to represent events (Chen et al., 2009; Bejan and Harabagiu, 2010).

Most recent neural models use contextual embedding and character-based embedding of event triggers with some pairwise features to represent events (Kenyon-Dean et al., 2018; Huang et al., 2019; Cattan et al., 2020). These works do not use argument information, and expect the contextual embedding includes all the necessary information.

Argument information has been integrated into event representations either by encoding some string-level features (Peng et al., 2016; Choubey and Huang, 2017) or by entity-level coreference co-training (Lee et al., 2012; Barhom et al., 2019).

In contrast, our representation learning of events has a unified system to encode the event triggers and the argument entities, which avoids the costly co-training while learning more advanced features that express the arguments on their own and their interactions with the event triggers.

Pairwise Representation Learning in Other NLP Tasks. Pairwise representation learning has been widely adopted to model the relationships of two pieces of text. The main goal is to learn contextualized sentence representations. Earlier systems commonly implement with attention mechanisms in recurrent (Hermann et al., 2015), convolutional (Yin and Schütze, 2018) or Transformer-style (Vaswani et al., 2017) neural networks to deal with text generation, such as neural machine translation (Bahdanau et al., 2015), document reconstruction (Li et al., 2015), and document summarization (Nallapati et al., 2016); machine comprehension (Hermann et al., 2015), textual entailment (Rocktäschel et al., 2016; Devlin et al., 2019), etc.

In this work, we develop the pairwise representation learning for modeling the relationship of two mentions within two separate sentences rather than the relationship of the two sentences themselves. To the best of our knowledge, we are the first to (i) study pairwise representation for event pairs by letting two mentions learn from each other's context from the beginning [3] , and (ii) build structured representation between events by fine-grained argument reasoning, without any hand-engineered features.

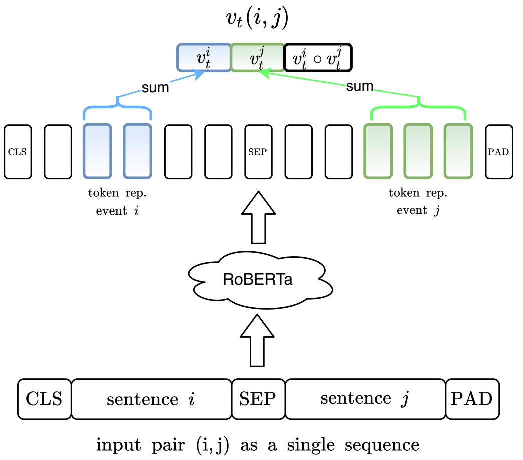

Figure 1: PAIRWISERL learns the trigger-only pairwise representation. v_t^i (resp. v_t^j) is the contextualized representation vector for the trigger in event i (resp. j). The whole trigger-based event pair (i, j) is denoted by $v_t(i, j)$ which is the concatenation: $[v_t^i, v_t^j, v_t^i \circ v_t^j]$.

3 PAIRWISERL for Coreference

PAIRWISERL takes two sentences containing each mention as the input and outputs a score indicating how likely the two mentions refer to the same event. Given the mention pair e_i and e_j with their arguments [arg0; arg1; loc; time], as shown in Fig 1, we concatenate the sentences of e_i and e_j, and encode the concatenated sentence using RoBERTa (Liu et al., 2019). After encoding each token of the sequence to a representation vector, we sum up the token representations of the mention span as the representations for event trigger and event arguments respectively: v_{t} for event trigger, $v_{\text{arg0}}/v_{\text{arg1}}$ for argument #0 or #1, v_{loc} for location and v_{time} for time.

Next, we conduct fine-grained coreference reasoning, as Figure 2 shows. The goal is to let each role of event arguments learn its contribution to the final task. For each role, where role $\in$ {t, arg0, arg1, loc, time}, we first build the following role-wise representation:

$$v_{\text{role}}(i,j) = [v^i_{\text{role}}, v^j_{\text{role}}, v^i_{\text{role}} \circ v^j_{\text{role}}] \quad (1)$$

where $\circ$ is element-wise multiplication. Because these four arguments may not always exist in the local context, if one of the role is missing, then the corresponding v^i_{role} will be a zero vector.

We keep the v_{t} as the main representation in PAIRWISERL, and let each of the remaining four arguments contribute a feature value indicating their own decisiveness. The feature value is learnt with a multi-layer perceptron (MLP) as follows:

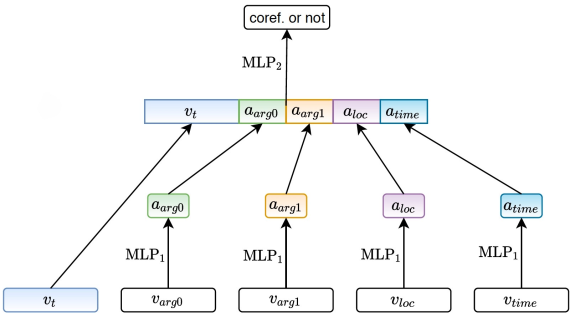

Figure 2: The full reasoning process in PAIRWISERL. The final PAIRWISERL representation is the concatenation of the trigger's representation and four feature values, each coming from a mention argument.

$$a_{\text{role}}(i,j) = \text{MLP}_1(v_{\text{role}}(i,j)) \quad (2)$$

where "role" refers to mention arguments other than the trigger, MLP_1 has four layers and the output of MLP_1 is a single scalar as the argument feature value. As a result, the final representation PAIRWISERL for event coreference is:

$$v(i,j) = [v_{\text{t}}(i,j), a_{\text{arg0}}, a_{\text{arg1}}, a_{\text{loc}}, a_{\text{time}}] \quad (3)$$

Since entities do not have arguments, the final representation PAIRWISERL for entity coreference is:

$$v(i,j) = v_{\text{t}}(i,j) \quad (4)$$

Once obtaining the pairwise representation $v(i, j)$, another four-layer MLP, as shown in Figure 2, will act as a binary classifier (i.e., is coreferential or not)

$$p(i,j) = \text{Softmax}(\text{MLP}_2(v(i,j))) \quad (5)$$

where $p(i, j)[0]$ is the probability that the two mentions i and j are coreferential.

4 Experiments

We apply PAIRWISERL to cross-document and within-document event coreference problems.

4.1 Cross-document Event Coreference

Dataset We use the ECB+ (Cybulska and Vossen, 2014) corpus to train and test our model. ECB+ is the largest and most popular dataset for cross-document Event Coreference, which is extended from ECB (Bejan and Harabagiu, 2010). For each topic in ECB, Cybulska and Vossen (2014) add different but similar events as subtopics. We follow

[3](Zeng et al., 2020) uses a similar method, and is a contemporary work with ours.

	Train	Dev	Test
Topics	25	8	10
Documents	574	196	206
Sentences	1,037	346	457
Event mentions	3,808	1,245	1,780
Event Singletons	1,116	280	623
Event Clusters	1,527	409	805
Entity mentions	4,758	1,476	2055
Entity Singletons	472	125	196
Entity Clusters	1,286	330	608

Table 1: ECB+ statistics. We follow the data split by Cybulska and Vossen (2015): *train*: 1, 3, 4, 6-11, 13-17, 19-20, 22, 24-33; *dev*: 2, 5, 12, 18, 21, 23, 34, 35; *test*: 36-45. Event/Entity Clusters include singletons.

the same setup as previous work (Cybulska and Vossen, 2015; Kenyon-Dean et al., 2018; Barhom et al., 2019). The detailed statistics are shown in Table 1. For both training and evaluation, we use gold event mentions. ECB+ also annotates coreference between entities that are arguments of events. We also use gold entity mentions to evaluate Entity Coreference on ECB+.

Preprocessing:

Argument generation. ECB+ annotates arguments of each event in the same sentence, but does not annotate the role of the arguments and the event that the arguments belong to. To predict arguments for each event mention, we use AI2 SRL system ,[4] which is a reimplementation of Shi and Lin (2019), and then we map the predicted arguments to the gold arguments. If any gold argument span overlaps with a predicted argument span, we assign the predicted role to it.

Topic Clustering. Topic clustering is a common componet of cross-document coreference because it is computationally inefficient to calculate similarity of the mention pairs in all the documents. People prefer to only collect mention pairs within documents that are related. Barhom et al. (2019) implements a strong topic clustering model that uses the K-Means algorithm on the documents represented by TF-IDF scores of unigrams, bi-grams, and trigrams. They choose the K value based on the Silhouette Coefficient method (Rousseeuw, 1987), and perfectly get the number of gold topics. Though there still exist wrong documents in each topic cluster, their nearly perfect clustering allows very simple baseline models to achieve very good results (Barhom et al., 2019). Since we focus on the improvement that the pairwise representation can bring, we use exactly the same topic clustering model they implemented. We use gold topics for training, and predicted topics for inference.

Postprocessing: Mention Clustering. After training the pairwise coreference scorer, following previous work (Choubey and Huang, 2017; Kenyon-Dean et al., 2018; Barhom et al., 2019; Cattan et al., 2020), we apply agglomerative clustering to the event pairs by the score from the trained scorer in Equation 5. Agglomerative clustering merges event clusters until no cluster pairs have a similarity score higher than a threshold. We define the cluster pair similarity score as the average score of all the event pairs across two clusters, and tune the threshold on development data.

Results: We compare with two state-of-the-art cross-document Event Coreference models using different methods: Barhom et al. (2019), which jointly trains Entity Coreference and Event Coreference, and Cattan et al. (2020), which jointly learns mention detection and coreference. We also compare with the same head lemma baseline implemented by Barhom et al. (2019), which simply clusters events with same head lemma.

To reveal the true merit of PAIRWISERL, in Table 2, we separately show the effectiveness of the structured and pairwise representations as proposed in PAIRWISERL. In "Unstructured", our system only uses the trigger representation, Equation 4, to denote the representation of a pair of mention; in "Structured", the structured representation depicted in Equation 3 is used; in "Unpaired", the representations of trigger and arguments are generated with their own sentence only instead of the concatenated two sentences; in "Pairwise", the representations are generated by the two concatenated sentences as described in Sec 3. We see that using only structured representations improves F1 by 1.6 (from 81.3 to 82.9) from the baseline unpaired+unstructured setting, and using only pairwise representation improves F1 by 2.7 (from 81.3 to 84.0). Both 82.9 and 84.0 already outperform the state-of-the-art model Cattan et al. (2020) on all of the evaluation metrics with large margins, particularly when using pairwise representation, 84.0 vs. 81.0 by CoNLL F1 score. When incorporating

[4] https://demo.allennlp.org/semantic-role-labeling

Model	MUC			B^3			CEAF$_e$			CoNLL
	R	P	F1	R	P	F1	R	P	F1	F1
same head lemma	76.5	79.9	78.1	71.7	85	77.8	75.5	71.7	73.6	76.5
Barhom et al. (2019)	77.6	84.5	80.9	76.1	85.1	80.3	81	73.8	77.3	79.5
Cattan et al. (2020)	85.1	81.9	83.5	82.1	82.7	82.4	75.2	78.9	77.0	81.0
Unpaired										
Unstructured	81.7	84.4	83.1	79.8	86.3	82.9	79.6	76.7	78.1	81.3
Structured	84.6	84.6	84.6	83.6	84.2	83.9	80.2	80.2	80.2	82.9
Pairwise										
Unstructured	91.6	83.1	**87.2**	89.4	81.1	85.1	75.0	85.5	79.9	84.0
Structured	88.1	85.1	86.6	86.1	84.7	**85.4**	79.6	83.1	**81.3**	**84.4**
Structured$_{\text{BERT}}$	87.4	81.4	84.3	85.7	80.2	82.9	73.7	80.9	77.1	81.4

Table 2: Cross-document event coreference performance on ECB+. All the models use gold mentions and predicted topics. "Unstructured" means the model only uses the representation of the event trigger. "Structured" means the model uses the structured representation of arguments. "Unpaired" is the baseline model without pairwise representation. "Pairwise" is the model using pairwise representation. Structured$_{\text{BERT}}$ means this baseline model uses BERT (Devlin et al., 2019) as contextual embeddings instead of RoBERTa. Details in Sec 4.1.

Model	MUC			B^3			CEAF$_e$			CoNLL
	R	P	F1	R	P	F1	R	P	F1	F1
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
PAIRWISERL	92.3	86.8	**89.5**	82.1	81.0	**81.5**	68.0	80.2	**73.6**	**81.5**

Table 3: Cross-document Entity coreference performance on ECB+. All the models evaluate on gold mentions and predicted topics.

structured representation into pairwise representation, the system obtains further improvement (from 82.9 to 84.4 CoNLL F1). Please note that both Barhom et al. (2019) and Cattan et al. (2020) have relatively complex systems to learn event features as well as entity features. Our system only models the trigger and arguments representations given the context of two involved mentions. It clearly demonstrates the superiority of our model in learning the event-pair representation.

ECB+ also annotates coreference between entities that are arguments of events. Because entities do not have arguments, we just use PAIRWISERL to learn the pairwise representation as Equation 4. We compare with the same two baselines: Barhom et al. (2019) and Cattan et al. (2020). Both of these two baselines train their model on gold mentions, so the comparison is fair. As shown in Table 3, our system PAIRWISERL significantly outperforms the two baselines: 81.5 vs. 73.1.

	Train	Dev	Test
Documents	360	169	167
Event mentions	12,976	4,155	4,375
Event Singletons	5,256	2,709	2,358
Event Clusters	7,460	3,191	2,963

Table 4: KBP statistics. We use KBP2015 for *train*, KBP 2016 for *dev* and KBP 2017 for *test*. Event Clusters include singletons.

4.2 Within-document Event Coreference

Within-document event coreference focuses on event pairs in the same document, so topic clustering of documents is not needed. We use the same pairwise scorer and mention clustering algorithm described in Section 4.1.

We evaluate on the most widely used KBP benchmark. Similar to Huang et al. (2019) and Lu et al. (2020), we use the KBP 2015 dataset (Ellis et al., 2015) as training data, the KBP 2016 dataset (Ellis et al., 2016) as dev data, and the KBP 2017 (Get-

Model	MUC	B^3	$CEAF_e$	BLANC	AVG-F
Huang et al. (2019)					
Predicted Mentions	35.66	43.20	40.02	32.43	36.75
Lu et al. (2020)					
Predicted Mentions	39.06	47.77	45.97	30.60	40.85
Gold Mentions	-	-	-	-	53.72
Unpaired (Gold Mentions)	60.23	52.34	47.44	45.32	51.33
PAIRWISERL (Gold Mentions)	63.67	58.41	54.66	51.72	**57.12**
$\text{PAIRWISERL}_{\text{BERT}}$ (Gold Mentions)	59.11	53.11	50.6	45.81	52.16

Table 5: Within-document event coreference performance on KBP17. Please note that the KBP15 corpus (training data) only provides trigger annotation, so we only evaluate the performance of trigger representation. "Unpaired" is the baseline model without pairwise representation. $\text{PAIRWISERL}_{\text{BERT}}$ means this baseline model uses BERT as contextual embeddings instead of RoBERTa.

man et al., 2015) as test data. The detailed statistics are shown in Table 4. Because the training data KBP 2015 dataset does not have the annotation of arguments, we evaluate the performance of the representation with trigger only.

We compare with two state-of-the-art systems on the KBP benchmark: Huang et al. (2019), which exploits unlabeled data to learn argument compatibility in order to improve coreference performance, and Lu et al. (2020), which jointly learns event detection and event coreference. Lu et al. (2020) claims the state-of-the-art performance when predicting event coreference given predicted events, and they also report numbers using gold event mentions. Our model does not conduct mention detection, so we report our performance on gold mentions only (this is still fair since the prior SOTA system Lu et al. (2020) reports on gold mentions too) and leave our numbers on predicted mentions as future work. As shown in Table 5, PAIRWISERL outperforms the unpaired baseline model with a big margin: 57.12 vs. 51.33 (on "AVG-F"). This further verifies the effectiveness of the pairwise representation in modeling event coreference regardless of whether it is within-document or cross-document. We also need to give credit to RoBERTa that helps our simple model easily outperform the state-of-the-art model (57.12 vs. 53.72), which is a much more complicated model than ours.

4.3 Implementation Details

For both ECB+ and KBP models, we use $\text{RoBERTa}_{\text{Large}}$ as the encoder. The sizes of four layers of MLP_1 are: 3076/1024/1024/1. The sizes of four layers of MLP_2 are: 3072/1024/1024/1. We set the learning rate as 1e-06, the batch size as 32, and we run 10 epochs for training. All hyperparameters are tuned based on development data, including the threshold of agglomerative clustering.

5 Analysis

To further understand why pairwise representation performs much better than unpaired representation, and what limitations pairwise representation still has, we do a quantitative analysis on the errors of PAIRWISERL and the errors of the unpaired baseline model on ECB+. For each model, we randomly sample 100 errors: 50 false negatives and 50 false positives. False negative means that the gold label of the event pair is "coref", but the model predicts "not coref". False positives mean that the gold label of the event pair is "not coref", but the model predicts "coref". We manually classify these errors into different types, and study the difference between the error distributions of the two models.

5.1 False Negatives

Given event mention pairs with the two sentences, as listed on the bottom of Figure 3, we classify these false negatives into these 7 types: "No direct evidence", "Different contexts". "Similar contexts", "Require argument matches", "Annotation mistakes", "Require commonsense knowledge", and "Other".

"No direct evidence" means that, just by reading the two sentences, there is no evidence in them to decide that these two mentions must be the same event. For example:

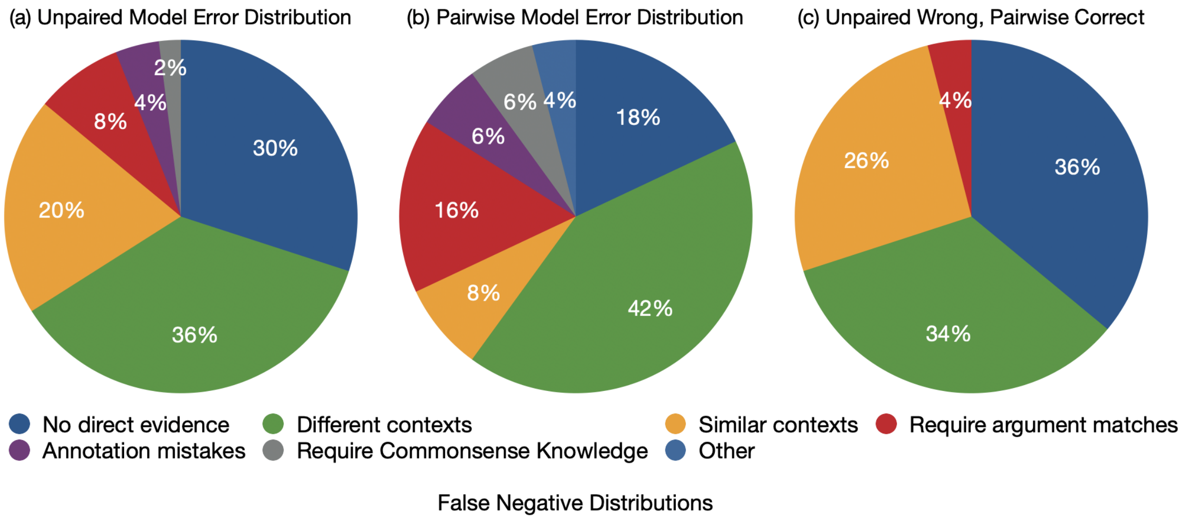

Figure 3: False Negative distributions of unpaired model, and pairwise model. False negative refers to gold coreferential event pairs that the model predicts "not coref". More details in Sec 5.1

s_1: Smith, 26, who played a young political researcher in the show, will become the biggest star of all after **winning** the role of the 11th Doctor.

s_2: The guy is relatively unknown and the skeptics wondered if the right person was **chosen**.

Just by reading these two sentences, we really do not know whether the event "winning" and the event "chosen" are same event or not. To make the correct prediction, more contexts are needed. Most prior work encoded events within only a single sentence; in this work, we use a single sentence as event context for fair comparison. As shown in Figure 3, the unpaired model has 30% mistakes belong to "No direct evidence", while the pairwise model only has 18.4%. This indicates that pairwise model may be more capable to learn the similarity between the context in order to make a "guess" that is more likely to be correct. However, 18.4% is also high. This indicates that sentence-level representation is not enough to represent an event. Event arguments usually appear in multiple sentences. Representing events in a multi-sentence level could be interesting to future work.

"Different contexts" means that the two sentences are too hard for the model to understand and there is no obvious textual similarity for the model to rely on. However, if the model understands the contexts completely, it should make the correct prediction. For example:

s_1: Scott Peterson has been found guilty of first-degree murder, a verdict that means he could be **executed** if these same jurors vote as the "conscience of their community" that he deserves to die for his crimes.

s_2: Laci Peterson's loved ones have "a hole in their hearts that will never be repaired," a prosecutor told jurors today as he asked them to send convicted double-murderer Scott Peterson to his **death** for killing his wife and unborn son.

In this example, sentences are both complicated and sharing limited vocabulary, but by understanding the sentences, we can say that two event mentions are the same event. We regard this error type as hard cases, and the pairwise model suffers from these hard cases. 40.2% mistakes of the pairwise model belong to hard cases "Different contexts". Please note that a higher ratio (40.2% vs. 36%) doesn't mean our pairwise model is worse than the unpaired competitor; this is because our system has resolved most of the simpler cases so the hard cases occupied the majority proportion of remaining errors. Improving the performance on complicated sentences still acts as the main challenge.

"Similar contexts" means that the two sentences are very similar, which should be easy for the model to make the correct prediction. For example:

s_1: A strong **earthquake** struck Indonesia’s Aceh province on Tuesday, killing at least one person and leaving two others missing.

s_2: A powerful **6.1 magnitude earthquake** hit Indonesia’s Aceh province, on the island of Sumatra .

These two sentences have similar context and similar structure, which should be easy to predict two mentions as the same events. We regard this error type as easy cases. Our pairwise model reduces the error rate dramatically from 20% to 8% in this category, which indicates that the pairwise model is very effective to solve these simple cases.

“Require argument matches” means that to make the correct prediction, systems need to use more context or external knowledge to conduct non-trivial argument matching. For example:

s_1: An earthquake with a preliminary magnitude of 4.4 **struck** in Sonoma County this morning near The Geysers, according to the U.S. Geological Survey.

s_2: The temblor **occurred** at 9:27 a.m. , 13 miles east of Cloverdale and 2 miles southeast of The Geysers , where geothermal forces by more than 20 power plants are harnessed to provide energy for several North Bay counties.

In order to make the correct prediction of these two sentences, the model need to realize the match between “9:27 a.m.” and “this morning”, and know that “Sonoma County” is “13 miles east of Cloverdale”, which requires more context or external knowledge.

We also sample 50 errors of unpaired model where the pairwise model could predict correctly. As shown in Figure 3(c), the improvement of the pairwise representation mainly comes from better performance on “No direct evidence”, “Different contexts” and “Similar contexts”. We find that the sentences are usually very long for these errors, which suggests that the pairwise representation is better at understanding the meaning of long sentences than the unpaired representation is.

5.2 False Positives

For the sampled false positives, we also manually classify them into 7 types same as the types of false negatives. The only difference is that, now “Similar contexts” become hard cases, and “Different contexts” become easy cases. As shown in Figure 4, for both the unpaired model and the pairwise model, most of the precision errors belong to “No direct evidence” and “Annotation mistakes”. After carefully studying these errors, we find that it is actually very hard to determine that two mentions are not the same event. For example:

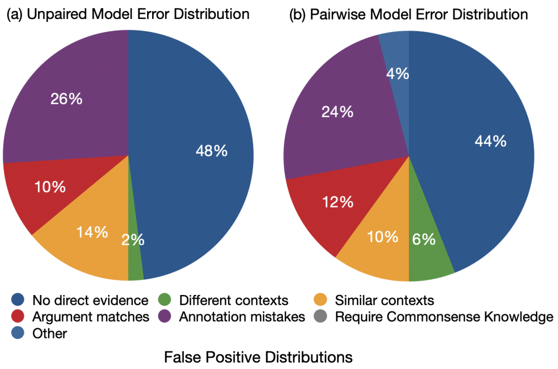

Figure 4: False positive distributions of unpaired model, and pairwise model. False positive refers to gold event pairs that are not coreferential, but the model predicts “coref”. More details in Sec 5.2

s_1: Four bombs were dropped within just a few moments - two **landed** inside the camp itself, while the other two bombs were dropped near the airstrip where a UN helicopter was delivering much needed food aid.

s_2: "Two of the bombs **fell** within the Yida camp , including one close to the school," said UNHCR spokesman Adrian Edwards .

By understanding these two sentences, we think, without knowing whether “the camp itself” in the first sentence is the same camp as “Yida camp” in the second sentence, it is impossible to make the correct prediction. The gold label for this pair is “not coref”, so we can only classify it to “No direct evidence”. We think that these errors again emphasize that the representation of events should be multi-sentences level instead of sentence level. We only use SRL to find event arguments, which limits arguments to be in the same sentences. We think that it may be essential to find events across sentences in future works.

We also find that there exist some errors that we think are *annotation mistakes*. For example:

s_1: Smith, 26, who played a young political researcher in the show, will become the biggest star of all after **winning** the role of the 11th Doctor .

s_2: The BBC says little-known actor Matt Smith will **take over** the title role in the long-running sci-fi series “Doctor Who.”

We do not see any reasons that these two mentions are not the same event, but if there are other contexts indicating that they are not the same event, this error would be classified to “No direct evidence”. So in conclusion, to further improve the performance on false positives, longer-range context will be needed.

6 Conclusion

In this work, we propose a simple representation learning scheme, PAIRWISERL, for event coreference. PAIRWISERL learns a mention-pair representation by forwarding concatenated sentences into RoBERTa, where sentences provide the context of mentions. This representation is applied to both within-document and cross-document event coreference benchmarks and obtains state-of-the-art performance. In addition, we augment this pairwise representation with structured argument features to further improve its performance.

Acknowledgments

This work was supported by Contract FA8750-19-2-1004 with the US Defense Advanced Research Projects Agency (DARPA), the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the BETTER Program, and a Focused Award from Google. Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

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

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, Uppsala, Sweden. Association for Computational Linguistics.

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*.

Zheng Chen, Heng Ji, and R Haralick. 2009. Event coreference resolution: Algorithm, feature impact and evaluation. In *Proceedings of Events in Emerging Text Types (eETTs) Workshop, in conjunction with RANLP, Bulgaria*.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *LREC*, pages 4545–4552.

Agata Cybulska and Piek Vossen. 2015. " bag of events" approach to event coreference resolution. supervised classification of event templates. *Int. J. Comput. Linguistics Appl.*, 6(2):11–27.

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

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M Strassel. 2015. Overview of linguistic resources for the tac kbp 2016 evaluations: Methodologies and results. In *TAC*.

Joe Ellis, Jeremy Getman, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M Strassel. 2016. Overview of linguistic resources for the tac kbp 2015 evaluations: Methodologies and results. In *TAC*.

Jeremy Getman, J. Ellis, Zhiyi Song, Jennifer Tracey, and S. Strassel. 2015. Overview of linguistic resources for the tac kbp 2017 evaluations: Methodologies and results. *Theory and Applications of Categories*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NeurIPS*, pages 1693–1701.

Yin Jou Huang, Jing Lu, Sadao Kurohashi, and Vincent Ng. 2019. Improving event coreference resolution by learning argument compatibility from unlabeled data. In *Proceedings of the 2019 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 785–795.

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, New Orleans, Louisiana. Association for Computational Linguistics.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *ACL*, pages 1106–1115.

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*.

Yaojie Lu, Hongyu Lin, Jialong Tang, Xianpei Han, and Le Sun. 2020. End-to-end neural event coreference resolution. *arXiv preprint arXiv:2009.08153*.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*, pages 280–290. ACL.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, Austin, Texas. Association for Computational Linguistics.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *ICLR*.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.

Wenpeng Yin and Hinrich Schütze. 2018. Attentive convolution: Equipping cnns with rnn-style attention mechanisms. *TACL*, 6:687–702.

Yutao Zeng, Xiaolong Jin, Saiping Guan, Jiafeng Guo, and Xueqi Cheng. 2020. Event coreference resolution with their paraphrases and argument-aware embeddings. In *Proceedings of COLING*, pages 3084–3094.