Capturing the Content of a Document through Complex Event Identification

Zheng Qi, Elior Sulem, Haoyu Wang, Xiaodong Yu, Dan Roth
Department of Computer and Information Science, University of Pennsylvania
{issacqzh, eliors, why16gzl, xdyu, danroth}@seas.upenn.edu

Abstract

Granular events, instantiated in a document by predicates, can usually be grouped into more general events, called *complex events*. Together, they capture the major content of the document. Recent work grouped granular events by defining event regions, filtering out sentences that are irrelevant to the main content. However, this approach assumes that a given complex event is always described in consecutive sentences, which does not always hold in practice. In this paper, we introduce the task of complex event identification. We address this task as a pipeline, first predicting whether two granular events mentioned in the text belong to the same complex event, independently of their position in the text, and then using this to cluster them into complex events. Due to the difficulty of predicting whether two granular events belong to the same complex event in isolation, we propose a context-augmented representation learning approach CONTEXTRL that adds additional context to better model the pairwise relation between granular events. We show that our approach outperforms strong baselines on the complex event identification task and further present a promising case study exploring the effectiveness of using complex events as input for document-level argument extraction.[1].

1 Introduction

Event extraction aims to identify event predicates and arguments from text and then identify their types and roles respectively, helping humans to easily understand the events. It has attracted considerable interest in the last few years (Chen et al., 2015; Nguyen et al., 2016; Sha et al., 2018; Lin et al., 2020; Ebner et al., 2020; Chen et al., 2020b; Li et al., 2021) due to the vast amounts of unstructured text available in domains like e-commerce, healthcare and industry. However, considering each

[1]The data and code are available at http://cogcomp.org/page/publication_view/978

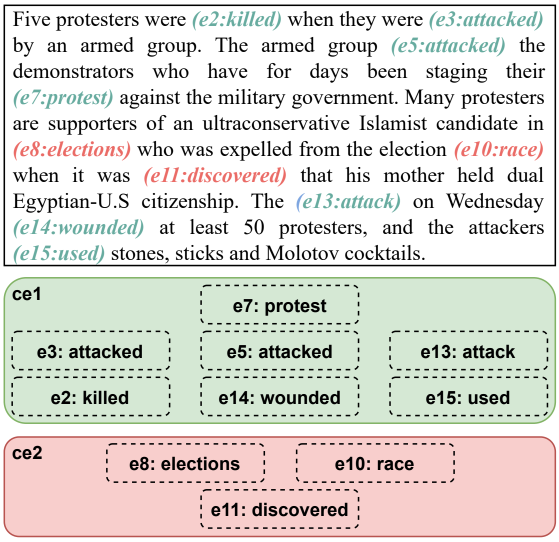

Figure 1: An example of complex events (ce1 and ce2) described in the document. For clarity, not all event mentions are shown in the figure.

granular event instantiated in the document by a predicate in isolation is not sufficient for understanding the entire context of the document. Since granular events can be grouped into more general events, called *complex events*, we suggest using them to capture the major content of the document.

A document could contain any number of complex events where each complex event contains more than one granular event. For example, Figure 1 represents 10 granular events appearing in a document. One can group the granular events into two complex events as follows: (i) **ce1** (in green) that includes the granular events related to a protest, (ii) **ce2** (in red) that includes granular events that, taken together, describe elections. These two complex events represent the major two events that the text describes.

Recently, Chen et al. (2020a) used the notion of event regions, a byproduct of document-level argument extraction, by filtering out sentences that are irrelevant to the main content and then parti-

Proceedings of the 11th Joint Conference on Lexical and Computational Semantics, pages 331 - 340
July 14-15, 2022 ©2022 Association for Computational Linguistics

tioning the text into several parts. Therefore, event regions are defined as consecutive sentences that include relevant arguments. However, compared to the complex event that groups related granular events together, the event region fails to capture the following two scenarios: (i) sentences that include granular events in the same complex event (e.g. the first and the last sentences in Figure 1) are separated by sentences that include granular events in another complex event; (ii) two granular events belonging to different complex events may appear in the same sentence.

Therefore, in this paper, we introduce the task of complex event identification which aims to group granular events instantiated by predicates into complex events, independently from the position of the predicates in the text. For example, in Figure 1, $e2$ and $e13$ belong to the same complex event (**ce1**) while $e8$ belongs to **ce2**.

To perform complex event identification, we first (i) predict whether two granular events belong to the same complex event, independently of their positions in the document, and then (ii) cluster them into complex events based on the pairwise relation predicted from step (i).

However, only considering the joint representations of two granular events is not sufficient to model the pairwise relation. For example, in Figure 1, it is difficult to infer that "demonstrators have for days been staging their *protest* against the government" ($e7$) and "the attackers *used* stones, sticks and Molotov cocktails" ($e15$) belong to the same complex event until we know that "The armed group *attacked* the demonstrators" ($e5$). Moreover, since both "demonstrators have for days been staging their *protest* against the government" ($e7$) and "Many protesters are supporters of a candidate in *elections*" ($e8$) mention some information about the protest, they might be considered to be in the same complex event. However, after reading more parts of the document, we know that $e8$ belong to the election complex event (**ce2**), which occurs before the protest complex event (**ce1**) containing $e7$.

Hence, we propose a context-augmented representation learning approach ContextRL that adds additional context to model the pairwise complex event relation. Specifically, we compute the attention distribution of other granular events in the document based on the joint representation of two granular events and select the one with the highest score as the context event. Regarding two granular events as a single entity, if they belong to the same complex event, the system would add a granular event in the same complex event, to improve the expressiveness of their relatedness; if they are not in the same complex event, then the system would add an additional granular event to make them more distinguishable, relative to this context event.

Since there is not a dataset tailored to the task of complex event identification, we derive the complex event annotation from the HiEve dataset (Glavaš et al., 2014) that focuses on event-event relations. We show that our proposed approach outperforms strong baselines.

Moreover, since related granular events are grouped into the same complex event, the scope of the complex event is supposed to include all the information required for the prediction of the arguments of its granular events. Hence, we conduct a case study on the WikiEvents dataset (Li et al., 2021) to explore the effectiveness of using complex events as the input for the document-level argument extraction task. We show that, when enough granular events are annotated, using complex events as input filters out noisy and irrelevant information, motivating the model to only focus on the related granular events.

The major contributions of this paper can be summarized as follows:

1. We introduce the complex event identification task that allows one to group related granular events, independently from the position of the predicates in the text, into complex events that, together, capture the major content of the document.

2. We present a context-augmented representation learning approach ContextRL tailored to this task, showing that this approach outperforms strong baselines on the complex event annotation derived from the HiEve dataset. We also analyze the effect of the context event.

3. We conduct an exploratory case study on the WikiEvents dataset, showing that using complex events as the input for document-level argument extraction allows the system to only consider relevant sentences and is a promising approach for this task.

2 Related Work

Event Extraction In the last few years, most of the work on event extraction focuses on the sentence level (Chen et al., 2015; Nguyen et al., 2016; Sha et al., 2018; Lin et al., 2020). Experiments are usually performed on the ACE dataset (Walker et al., 2006). In that setting, events correspond to predicates and event extraction consists in (i) identifying the predicates in the sentence (Trigger Identification); (ii) classifying them according to a predefined ontology (Trigger Classification); (iii) identifying the arguments (Argument Identification); (iv) identifying the role of the argument relative to the predicate (Argument Classification).

However, since arguments are usually scattered across sentences, recent works (Ebner et al., 2020; Chen et al., 2020b; Li et al., 2021) extended the argument extraction components (iii) and (iv) to the document level, trying to capture arguments that are not in the same sentence as the predicate. Li et al. (2021) introduced the WIKIEVENTS dataset, going beyond the RAMS dataset (Ebner et al., 2020) by annotating several granular events per document. However, this approach does not address complex events and focuses on argument roles relative to granular events. In Section 4.6, we explore the effectiveness of using complex event as input for document-level argument extraction, experimenting on the WIKIEVENTS dataset.

Event Regions Chen et al. (2020a) addressed document-level argument extraction as well but they also obtain as byproducts event regions, defined as adjacent sentences that include relevant arguments. Complex events differ conceptually from event regions in two main points: (i) sentences that contain predicates of granular events in the same complex event can be separated in the text by sentences that include predicates of granular events in other complex events. (ii) the context of complex events may be overlapping as granular events in different complex events may share the same sentence.

Event-Event Relations Event-event relations include coreference and subevent relations. Event coreference (Lee et al., 2017; Barhom et al., 2019; Yu et al., 2022) allows one to group granular events referring to the same granular event while a subevent relation (Aldawsari and Finlayson, 2019; Wang et al., 2020) indicates that one granular event is a parent or child of another granular event. However, the notion of complex events is broader than both of them: (i) granular events in the same complex event also have other relations than subevent relations, such as temporal and causal relations; (ii) granular events in the same complex event can have different content. For example, $e3$: "attacked" and $e14$: "wounded" in Figure 1 are in the same complex event although they are not coreferred.

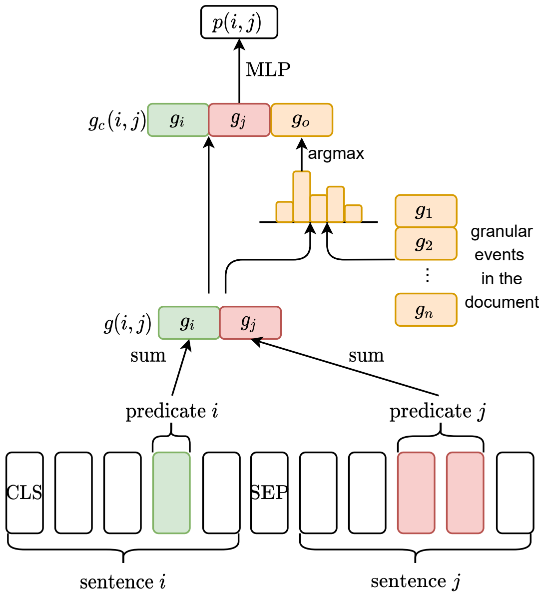

Figure 2: CONTEXTRL framework. g_i,g_j are contextualized representations of predicate i and j respectively. $g(i,j)$ denotes the concatenation of two granular event representations and g_o denotes the context event representation. $g_c(i,j)$ denotes the concatenation of $g(i,j)$ and g_o. $p(i,j)$ denotes the probability of belonging to the same complex event.

3 Method

In this section, we present our context-augmented learning approach CONTEXTRL. We address the complex event identification task as a pipeline, first predicting whether two granular events belong to the same complex event, independently of their position in the text, and then grouping them into complex events based on pairwise predictions. We first introduce our pairwise complex event relation extraction model in Section 3.1 and then introduce the granular event clustering step in Section 3.2.

3.1 Context-Augmented Pairwise Complex Event Relation Extraction

Our context-augmented model takes two sentences that contain predicates and the representations of other granular events (context event candidates) in the document as input, outputting a score indicating how likely two granular events belong to the same complex event. Since it is time-consuming and computationally expensive to encode all other granular events every time, we propose an efficient and effective way to obtain the representations of some granular events except the two granular events without further computation and regard them as context event candidates. During training and evaluation, with a batch size of n, we obtain representations of $2n$ granular events and use $2(n-1)$ granular events except the two granular events as context event candidates. To make sure these $2(n-1)$ granular events are in the same document as the two granular events, we only shuffle pairs within each document instead of shuffling across documents. We show its effectiveness in Section 4.5.

Given two granular events i and j, as shown in Figure 2, we first concatenate the sentences where their predicates appear using *[CLS]* and *[SEP]* and then encode the sequence using RoBERTa (Liu et al., 2019) to learn a contextualized representation for each token in the sequence. The concatenation of two sentences allows each token to learn the context from both sentences. Since granular events are instantiated by predicates, which are consecutive spans within the sentence, we sum up representations of tokens in the predicates element-wisely to obtain the predicate representations g_i and g_j.

Next, to select the context event, we first use the concatenation of two granular event representations $g(i,j)$ and the representations of other granular events in the document s to compute the attention distribution $\alpha(i,j)$ as follows:

$$e_k(i,j) = v^{\intercal}\text{tanh}(W_g g(i,j) + W_s s_k + b_e)$$
$$\alpha(i,j) = \text{Softmax}(e(i,j))$$

where v, W_g, W_s are learnable matrix, b_e is a bias vector, s_k is the representation of k_{th} granular event and $e(i,j)$ is attention scores.

Then we select the granular event with the highest attention score as the context event and concatenate its representation g_o with the representations of two granular events to obtain the context-augmented representation $g_c(i,j)$ as follows:

$$o = \text{argmax}(\alpha(i,j))$$
$$g_c(i,j) = [g_i; g_j; g_o]$$

We also manually set an attention distribution threshold to guarantee that there is a granular event highly related to the two granular events. If the highest attention score is lower than the threshold, we mask the context event with 0.

Finally, we forward the context-augmented representation $g_c(i,j)$ into a linear layer to output the probability of belonging to the same complex event as follows:

$$p(i,j) = \text{Softmax}(W_c g_c(i,j) + b_c)$$

where W_c and b_c are a learnable weight matrix and a bias vector respectively.

3.2 Granular Event Clustering

After obtaining the pairwise complex event relation for each pair of granular events in the document, similar to the clustering step of previous work on the event coreference task (Choubey and Huang, 2017; Kenyon-Dean et al., 2018; Barhom et al., 2019; Cattan et al., 2020), we cluster them into complex events using agglomerative clustering. We define the distance between two granular events as the likelihood of not belonging to the same complex event. Agglomerative clustering merges event clusters until no cluster pairs have a linkage distance lower than the threshold, where the linkage distance is defined as the average distance of all the event pairs across two clusters.

In addition, we assume the scope of the complex event is the set of sentences that contain granular event predicates. Since a sentence may contain multiple predicates, the overlapping of scope between complex events is allowed by nature, which also contrasts with the event region definition shown in Section 2.

4 Experiments and Results

We conduct experiments on the complex event identification task, using our context-augmented representation learning approach CONTEXTRL to first extract pairwise relations and then group granular events into complex events through agglomerative clustering. We further present a promising case study on the WIKIEVENTS dataset (Li et al., 2021), showing the effectiveness of using only complex events as input for document-level argument extraction in Section 4.6.

	# Doc.	# Pairs	# CE	# Events/ CE
Train	60	38124	121	7.01
Dev	20	13810	44	6.93
Test	20	16227	54	7.07

Table 1: Statistics for the HiEve dataset and the complex event annotation derived from the HiEve dataset. CE denotes complex event.

4.1 Dataset

Since there is not a dataset tailored to the complex event identification task, we derive the complex event annotation from HiEve dataset (Glavaš et al., 2014) that annotates subevent and coreference relations. For each document, we first build an undirected acyclic graph where vertices are granular events connected by subevent relations (i.e., two events have either Parent-Child or Child-Parent relation) as edges, and then regard granular events in the same graph as belonging to the same complex event. We summarize the data statistics in Table 1. Note that the replication of this work on other texts requires the annotation of subevent relations with the constraint of not having two parents for the same subevent, unless they are co-referred, as in HiEve. Then, complex events can be derived from the annotation, as described here. We plan to explore the direct annotation of complex events in future work, which requires the compilation of fine-grained guidelines.

4.2 Baselines and Evaluation Metrics

We compare our model with three baselines. The first baseline is a Sequence Classification model (SC) plus the clustering step, where the Sequence Classification model encodes concatenated sentences using RoBERTa (Liu et al., 2019) and forwards the contextualized *[CLS]* token to a linear layer to compute the probability of belonging to the same complex event.

The second baseline is a strong predicate representation learning model (PRL) plus the clustering step, which replaces the contextualized *[CLS]* token with the concatenation of two predicate representations. The difference from our proposed model is that it does not use the context event.

Furthermore, since our complex event annotation is derived from the HiEve dataset that annotates subevent and coreference relations, we also compare our model with Wang et al. (2020), a SOTA joint constrained learning framework for extracting subevent, coreference and temporal relations, plus the clustering step. Since the HiEve dataset does not have temporal annotation, we only use its constraints related to subevent and coreference relations.

In terms of the clustering step, we use agglomerative clustering for the first two baselines that directly identify complex events. However, for the third baseline that extracts subevent relations to build complex events, since not all pairs of granular events in the same complex event have a subevent relation, using the probability of having subevent relations as distance would hinder such pairs from being grouped together. Thus, we follow the same graph-based clustering method as in Section 4.1.

In addition, we note that the method of Chen et al. (2020a) for event regions is not comparable with our method for complex event identification for the following reasons:

- The complex event and event region definitions are conceptually different, as the latter does not group granular events instantiated by predicates but rather partitions the document into segments, based on arguments.
- In the complex event annotation derived from the HiEve dataset, the proportion of complex events with consecutive sentences is only 91/219 = 41.6%, hindering Chen et al. (2020a)'s method from achieving competitive performance.
- Current datasets do not include gold data allowing such a comparison. Specifically, the HiEve dataset does not include argument annotation while the datasets CFEED and MUC-4 used in Chen et al. (2020a) do not annotate granular events.

Since both complex event identification and coreference resolution build clusters of granular events, we use coreference evaluation metrics [2] for evaluation, including MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), $CEAF_e$ (Luo, 2005) and BLANC (Recasens and Hovy, 2011), and report the results in Table 2. We also report CoNLL F_1 which is the average of MUC, B^3 and $CEAF_e$.

In addition, we report intermediary performances. For the pairwise complex event relation extraction task, the precision, recall and F_1 scores are reported in Table 3. For the subevent relation extraction task, we use the same evaluation setting as Wang et al. (2020), testing the model using 20% of the documents. The macro average precision, recall and F_1 scores of Parent-Child and Child-Parent relations are also reported in Table 3. Note that Wang et al. (2020) only kept 40% negative NoRel examples of the test set during evaluation while we evaluate on the entire test set.

[2] https://github.com/conll/reference-coreference-scorers

Model	MUC	B^3	$CEAF_e$	BLANC	CoNLL F_1
Using Subevent Relations for Complex Event Identification					
Wang et al. (Baseline)	72.68	60.38	55.39	47.22	62.82
Direct Complex Event Identification					
SC (Baseline)	51.69	59.94	43.34	48.9	51.66
PRL (Strong Baseline)	76.97	80.51	80.57	74.06	79.35
CONTEXTRL (Ours)	77.21	81.99	81.72	77.08	80.31

Table 2: Complex event identification performance on the complex event annotation derived from the HiEve. The columns correspond to different evaluation metrics. CoNLL F_1 is the average of MUC, B^3 and $CEAF_e$. We present our approach with 3 baselines. Wang et al. extracts subevent relations and then builds complex events by grouping granular events in the same acyclic graph to the same complex event. The last three models directly identify pairwise complex event relations and then cluster granular events into complex events through agglomerative clustering.

4.3 Experimental Setup

We encode the concatenated sequence using RoBERTa-large (Liu et al., 2019) to obtain 1024 dimensional token representations. Since the clustering step requires the pairwise prediction probability for each pair of granular events within the document, we set the max sequence length to 140 so that all pairs in the development set could fit in. The model contains 358.5M parameters in total. We use AdamW (Loshchilov and Hutter, 2017) to optimize the parameters, with a learning rate of 1e-6. For each setting, we train 12 epochs with a batch size of 16, and each epoch takes about 25 minutes. The attention distribution threshold of 0.047 is set based on the performance of the development set. The agglomerative clustering threshold for each setting is finetuned on the development set. We run all experiments on TITAN Xp GPU of size 12 GB.

Model	Precision	Recall	F_1
Subevent Relation Extraction			
Wang et al.	15.88	60.81	25.03
Pairwise Complex Event Relation Extraction			
SC	44.65	13.70	20.96
PRL	56.39	62.19	59.15
CONTEXTRL	55.75	64.85	59.96

Table 3: Subevent relation extraction performance on HiEve and Pairwise complex event relation extraction performance on the complex event annotation derived from the HiEve dataset. For Wang et al., we report the macro average scores of Precision, Recall and F_1. SC denotes the Sequence Classification model. PRL denotes the predicate representation learning model.

4.4 Complex Event Identification Results

In Table 2, we report evaluation metric scores for our approach and baselines. Our context-augmented representation learning approach CONTEXTRL outperforms all baselines, with a CoNLL F_1 score of 80.31. Besides, since it outperforms the SOTA subevent relation extraction model by a large margin, it motivates the study of complex event identification as an independent task.

We also show an example of complex events in the document predicted by CONTEXTRL in Figure 3. Granular events in green belong to a complex event describing the recent filing while granular events in red belong to another complex event describing the crime. These two complex events are interleaved in the document.

4.5 Context Event Analysis

As shown in Table 3, CONTEXTRL outperforms both baselines on the pairwise complex event relation extraction task. It achieves a F_1 score of 59.96, which is 0.81 higher than the strong baseline PRL. Compared with PRL, CONTEXTRL has a much higher recall which indicates it has fewer false negatives and more true positives. However, more true positives but a slightly lower precision indicates it contains more false positives. We discuss the reasons in the following paragraphs.

Effectiveness of Using Other Granular Events in the Batch as Context Event Candidates As shown in Table 4, in the test set, there are 2256 pairs of granular events belonging to the same complex event (positive pairs) and 13971 pairs of granular events not belonging to the same complex event (negative pairs). Of all positive pairs, 2238 (99.2%)

Complex Event Prediction

A new lawyer for OJ Simpson has filed a new attempt to gain his release from prison, alleging he was so badly *(e4: represented)* by lawyers in his *(e6: trial)* that he deserves a retrial. A 94-page document *(e7: filed)* in Court faults the *(e8: trial)* performance of attorneys Galanter and Grasso. It says he wanted to recover from sports memorabilia dealers family photos and personal mementoes *(e10: stolen)* from him. Simpson was convicted of charges including *(e14: kidnapping)* and armed *(e15: robbery)* in a hotel room crammed with two memorabilia dealers and a middle man, Simpson later *(e16: convicted)* of *(e17: felonies)*. Simpson, 64, was *(e18: sentenced)* to nine to 33 years behind bars. The *(e19: filing)* is a common next-step appeals strategy to blame trial and initial appeals attorneys for a defendant's conviction. Almost all grounds that lawyer *(e21: cited)* in the document fault Mr Galanter and Mr Grasso. Mr Grasso said "I'm behind OJ and I hope this *(e25: petition)* helps him get out of prison".

Figure 3: An example showing the prediction of complex events described in a document from the HiEve development set. Granular events in green belong to one complex event while granular events in red belong to another complex event. For clarity, not all event mentions are shown in the figure.

have at least one context event candidate that belongs to the same complex event as the pair of granular events, providing the opportunity of using an additional context event in the same complex event to improve the expressiveness of their relatedness. Of all negative pairs, 9440 (67.57%) have at least one context event candidate that belongs to the same complex event as one of the granular events in the pair. Of the rest of 4531 negative pairs, 4038 (89.12%) have both granular events that are not in any complex event. Such statistics indicate that negative pairs could select an additional context event from diversified candidates to make the pair of granular events distinguishable, relative to this context event.

Use Context Event in Positive Examples As we can see in Table 4, of all 2238 positive pairs that contain at least one context event candidate belonging to the same complex event as the pair of granular events, 655 mask the context event and the prediction accuracy is 64.12%. Of the rest of 1583 positive pairs, 942 use an additional context event that belongs to the same complex event as the pair of granular events, achieving an accuracy of 66.03%, while 641 pairs use other context events, having an accuracy of 64.12%. Therefore, adding an additional context event that belongs to the same complex improves the accuracy, which is equivalent to the number of true positives, and adding a context event not in the same complex event for positive pairs does no harm to the prediction.

Positive Pairs			
Same CE	Real Context	Mask	Total
Yes	1583 (1033)	655 (420)	2238
No	12 (6)	6 (3)	18
Total	1595	661	2256
Negative Pairs			
Same CE	Real Context	Mask	Total
Yes	5989 (5261)	3451 (3063)	9440
No	2700 (2670)	1831 (1816)	4531
Total	8689	5282	13971

Table 4: Analysis of the Pairwise complex event relation extraction performance of CONTEXTRL on complex event annotation. Real Context and Mask denote whether the pair uses a non-masked context event or not. Same CE (Yes or No) denotes whether the batch contains a context event candidate that belongs to the same Complex Event as one (for negative pairs) or two (for positive pairs) of the granular events in the pair. Number in parenthesis denotes the number of pairs predicted correctly.

Use Context Event in Negative Examples As shown in Table 4, of all 13971 negative pairs, 5282 mask the context event and the prediction accuracy is 92.37%. Of the rest of 8689 pairs, 2559 use a context event that belongs to the same complex event as one of the granular events in the pair, achieving an accuracy of 84.16%, while 6130 pairs use other context events, having an accuracy of 94.27%. Therefore, adding a context event in the same complex event as one of the granular events in a negative pair motivates the model to identify them to belong to the same complex event, increasing the number of false positives. Besides, since the model regards two granular events that describe different things as a single entity when computing the attention distribution, it is likely to select a context event not related to any of them in isolation, thus predicting the pair as negative with a great chance.

4.6 Complex Event as the Input of Document-level Argument Extraction

Since arguments are usually scattered across sentences, recent works on Argument Extrac-

	Train	Dev	Test
# Event types	49	35	34
# Arg types	57	32	44
# Docs	206	20	20
# Sentences	5262	378	492
# Events	3241	345	365

Table 5: Statistics for WIKIEVENTS dataset.

tion (Ebner et al., 2020; Chen et al., 2020b; Li et al., 2021) move from the sentence-level to the document-level (i.e., extracting the arguments from the whole document rather than a single sentence). However, the document not only has many noisy and irrelevant entities that prevent the model from extracting the arguments correctly, but also is too long to fit into a transformer-based model which limits the max sequence length.

Since granular events in the same complex event usually describe the same general content and they are unrelated to the granular events in other complex events, we assume the complex event should contain all the information required for the prediction of the arguments of its granular events.

Therefore, we conduct a case study on the WIKIEVENTS dataset to investigate the effectiveness of using complex events as input for document-level argument extraction. If the granular event belongs to a complex event, we use the sentences that contain granular event predicates in the same complex event as input; If the granular event does not belong to any complex event, we still use the entire document as input. We summarize the data statistics in Table 5.

Since WIKIEVENTS dataset does not have complex event annotation, we directly use our model CONTEXTRL trained on the complex event annotation derived from the HiEve dataset to group granular events in each document into complex events. Since the average number of annotated events per sentence in the test set is only 0.74, only using annotated granular events is not sufficient to build complex events. Therefore, we leverage an off-the-shelf verbal and nominal SRL system[3] to extract more granular events from documents. Consequently, 260/365 granular events belong to a complex event and using complex events as input reduces the average word count from 787.90 to 539.25.

[3]https://github.com/CogComp/SRL-English

After training the argument extraction model proposed in Li et al. (2021), we evaluate it on the test set with complex events as input and compare the performance with using the whole document as input. When using the whole document as input, the argument identification and classification head word F_1 scores are 71.21 and 66.55 respectively while using the complex event as input results in F_1 scores of 71.07 and 66.25 respectively. We could see that the model still achieves fairly close performance with much shorter inputs. Moreover, note that the complex event identification system is not trained on the WIKIEVENTS dataset, thus directly using the pre-trained model to identify complex events may also result in low performance.

We further show an "attack" granular event from the document, which has the largest improvement on the argument identification, in Figure 4. Using the complex event as input motivates the model to focus on the "attack" granular event, whereas using the whole document as input adds much irrelevant information (i.e. what is included in the interviews), distracting the model from the "attack" event and thus extracting incorrect arguments. Such difference in input and performance demonstrates the effectiveness of using complex events as the input for document-level argument extraction.

Complex Event as the Context
Osama bin Laden is charged to have had a role in the October 2000 attack on the USS Cole in the Yemeni port of Aden. This report features reporting by a Pulitzer-Prize-nominated team of New York Times reporters.
Whole Document as the Context
photo © 2001 corbis images all rights reserved web site copyright 1995-2014 WGBH educational foundation Hunting Bin Laden Osama bin Laden is charged to have had a role in the October 2000 attack on the USS Cole in the Yemeni port of Aden. This report features reporting by a Pulitzer-Prize-nominated team of New York Times reporters. Tracing the trail of evidence linking bin Laden to terrorist attacks, this report includes interviews with Times reporters. They discuss the terrorist attacks linked to bin Laden's complex network of terrorists, outline the elements of his international organization and details of its alliances and tactics.

Figure 4: An example showing the difference between using the complex event as the input and using the whole document as the input of document-level argument extraction. The predicate "attack" is in blue. Arguments in green are correctly extracted; arguments in red are missed; arguments in orange are extracted incorrectly.

5 Conclusion

In this work, we introduce the task of complex event identification and present a context-augmented approach CONTEXTRL tailored to this task. We show that our approach outperforms strong baselines on the annotation derived from the HiEve dataset and analyze positive effects of the context event. We further show the potential usefulness of using complex events as input for document-level argument extraction. For future work, we plan to directly annotate complex events from scratch with fine-grained guidelines. We also seek to extend our approach towards an end-to-end system with granular event extraction.

Acknowledgements

We thank the members of the Cognitive Computation Group for their insightful feedback ad well as the reviewers of the paper for their useful suggestions. This work was supported by Contracts FA8750-19-2-1004 and FA8750-19-2-0201 with the US Defense Advanced Research Projects Agency (DARPA). Approved for Public Release, Distribution Unlimited. This research is also based upon work supported in part by 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. 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

Mohammed Aldawsari and Mark Finlayson. 2019. Detecting subevents using discourse and narrative features. In *ACL*, pages 4780–4790.

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 *ACL*, pages 4179–4189.

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*.

Pei Chen, Hang Yang, Kang Liu, Ruihong Huang, Yubo Chen, Taifeng Wang, and Jun Zhao. 2020a. Reconstructing event regions for event extraction via graph attention networks. In *AACL*, pages 811–820.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*, pages 167–176.

Yunmo Chen, Tongfei Chen, and Benjamin Van Durme. 2020b. Joint modeling of arguments for event understanding. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 96–101.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *EMNLP*, pages 2124–2133.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *ACL*, pages 8057–8077. Association for Computational Linguistics.

Goran Glavaš, Jan Šnajder, Marie-Francine Moens, and Parisa Kordjamshidi. 2014. HiEve: A corpus for extracting event hierarchies from news stories. In *LREC*, pages 3678–3683.

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

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *EMNLP*, pages 188–197.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *NAACL-HLT*, pages 894–908. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *EMNLP-IJCNLP*, pages 7999–8009.

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.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *ICLR*.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *EMNLP*, pages 25–32.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *ACL*, pages 300–309.

Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *AAAI*.

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*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus ldc2006t06, 2006. *URL https://catalog. ldc. upenn. edu/LDC2006T06*.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint Constrained Learning for Event-Event Relation Extraction. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xiaodong Yu, Wenpeng Yin, and Dan Roth. 2022. Paired representation learning for event and entity coreference. In *Proc. of the Joint Conference on Lexical and Computational Semantics*.