Joint Detection and Coreference Resolution of Entities and Events with Document-level Context Aggregation

Samuel Kriman and Heng Ji
University of Illinois at Urbana-Champaign
{skriman2,hengji}@illinois.edu

Abstract

Constructing knowledge graphs from unstructured text is an important task that is relevant to many domains. Most previous work focuses on extracting information from sentences or paragraphs, due to the difficulty of analyzing longer contexts. In this paper we propose a new jointly trained model that can be used for various information extraction tasks at the document level. The tasks performed in this paper are entity and event identification, typing, and coreference resolution. In order to improve entity and event extraction, we utilize context-aware representations aggregated from the detected mentions of the corresponding entities and event triggers across the entire document. By extending our system to document-level, we can improve our results by incorporating cross-sentence dependencies and additional contextual information that might not be available at the sentence level, which allows for more globally optimized predictions. We evaluate our system on documents from the ACE05-E$^+$ dataset and find significant improvement over the sentence-level state-of-the-art on entity extraction and event detection.[1]

1 Introduction

Recently, large Transformer models, such as BERT (Devlin et al., 2019), Transformer-XL (Dai et al., 2019), and RoBERTa (Liu et al., 2019), have attracted a lot of attention from the Natural Language Processing (NLP) community. These models are typically pretrained on a large unlabeled corpus, and can be consequently fine-tuned for specific NLP tasks using a relatively small amount of supervised data. By adding shallow classifiers on top of the context-sensitive embeddings produced by these neural networks, state-of-the-art results have been achieved on various subtasks in Information Extraction (Eberts and Ulges, 2019; Wang et al., 2019; Asada et al., 2020).

Despite the ability of Transformer models to efficiently capture information across a long context, most IE work still focuses on extracting information from sentences (Lin et al., 2020; Eberts and Ulges, 2019), or, in some cases, short paragraphs (Wang et al., 2019). Additionally, some work has been done where longer documents are represented by encoding sentences or paragraphs separately (Du and Cardie, 2020; Ebner et al., 2020). While some datasets have been proposed which contain document-level annotations of entities and relations (Yao et al., 2019; Jain et al., 2020; Zaporojets et al., 2021), very little work has been done in effectively utilizing the fully available document-level context in order to produce globally optimal predictions.

The main contribution of this paper is the introduction and evaluation of our new neural IE model, which can be used to jointly perform various IE subtasks in the full document context. Our model receives only the original document text as input. After identifying relevant entity and event trigger mentions in the text, we perform clustering to determine which entities or events each mention belongs to. In order to make full use of the contextual information related to an entity/event in a given document, we aggregate information from all of the corresponding mentions to create a document-level representation, which can then be used for type prediction of entities and events. We focus on constructing a model which can efficiently tackle the challenges that arise in this currently not well explored variant of the task. Our approach achieves an improvement of about 2% absolute gain over the previous results on the ACE05-E$^+$ dataset in terms of F-score for entity extraction and event detection.

[1]Code is available at https://github.com/sam1373/long_ie

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 174–179

2 Model

2.1 Task Definition

We formulate the task of document-level information extraction in the following way. Each gold-standard sample from the dataset consists of the following parts:

1. Document D, represented by a sequence of word tokens $\{w_1, w_2, ..., w_n\}$.

2. The set of entities E, where each entity e is represented by a set of mentions in the document as well as an entity type: $e_i = (\{m_{i1}, m_{i2}, ...\}, l_i)$, where $l_i \in V_{ent}$ (the set of entity types in the dataset).

3. The set of events T, where each event t is represented by a set of event trigger mentions in the document as well as an event type: $t_i = (\{m_{i1}, m_{i2}, ...\}, l_i)$, where $l_i \in V_{ev}$ (the set of event types in the dataset).

The only input to our model is a sequence of tokens w. Given these tokens, the model is required to produce the following output: the predicted set of entities E' and events T', where each entity or event trigger mention corresponds to some span of tokens in D. In order to produce the above described output, the model operates in several steps: token encoding, entity and event trigger mention identification, coreference resolution, cluster aggregation and typing.

2.2 Token Encoding

The first step of our model consists of passing the document through a BERT-like large Transformer pre-trained for language modeling. Since we are working with potentially very long documents, for our model we choose the Longformer (Beltagy et al., 2020) as our encoder. Unlike BERT and most similar models which have quadratically increasing cost for attention, Longformer utilizes a modified more efficient attention pattern, which allows us to encode the entire document with a single Transformer pass. In addition, Longformer is pretrained on text up to 4,096 tokens, compared to 512 for models such as BERT and RoBerta.

Since the Longformer model operates using the Byte Pair Encoding subword tokenization scheme, in order to obtain the encoded representations of a given word we average the representations of corresponding word pieces. We additionally augment the word representations by concatenating a pre-trained GloVe (Pennington et al., 2014) word embedding, in order to allow easier access to word-level information. We find that this augmentation improves evaluation results, particularly event trigger identification and classification.

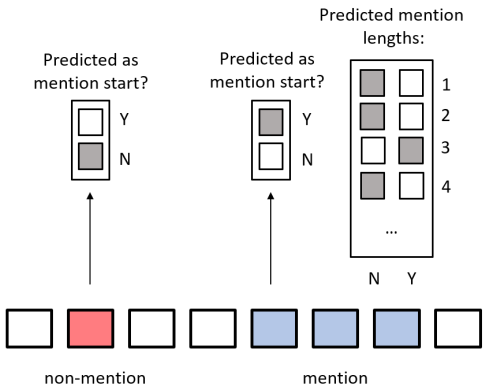

Figure 1: Identifying mention spans

2.3 Mention Identification

In order to extract relevant mentions from the text, we train two classifiers which are applied to each token, and used to determine, respectively, whether the token is the start of at least one relevant mention, and what are the lengths of mentions starting from this token. This is illustrated in Figure 1. Unlike commonly used span-based methods, where a representation is created for all possible mention spans up to a certain length, our approach does not require a significant increase in memory in order to consider longer entities, while still retaining the ability to potentially predict overlapping mentions.

The output of both of the classifiers at this stage is trained using cross-entropy loss. During training, further steps receive representations of gold mentions for input instead of the ones produced by the model.

2.4 Entity Coreference Resolution

Due to the large length of the documents and amount of mentions within them, it becomes impractical to use standard pairwise classification methods for coreference resolution. In order to find the entity and event clusters, we utilize the following method: mention representations are passed through a shallow residual neural network (referred to as the "coreference embedding network") to predict a special embedding for each predicted mention. In order to construct an appropriate embedding for each mention, we first obtain a represen-

tation by max-pooling over the encoded tokens that correspond to the mention span. Additionally, we concatenate a max-pooled representation of the sentence that contains the mention. The obtained mention representations are then passed through the coreference embedding network. This network is trained by using a combination of an attraction and repulsion loss, denoted as $\mathcal{L}_a$ and $\mathcal{L}_r$. Given an n-length batch of mention embeddings $\boldsymbol{m_1}, ..., \boldsymbol{m_n}$, let $C_1, C_2, ...$ denote the sets of mentions referring to the same entity. We use $c(i)$ to refer to the index of the set that mention $\boldsymbol{m_i}$ belongs to, and $o(i)$ to refer to the index of a randomly sampled incorrect mention set (so $\boldsymbol{m_i} \in C_{c(i)}, \boldsymbol{m_i} \notin C_{o(i)}$). Then the loss calculation can be written as follows:

$$\mathcal{L}_a = \sum_{i=1}^{n} ||\boldsymbol{m_i} - \frac{\sum_{\boldsymbol{m_j} \in C_{c(i)}} \boldsymbol{m_j}}{|C_{c(i)}|}||$$

$$\mathcal{L}_r = \sum_{i=1}^{n} Max(\mathcal{T} - ||\boldsymbol{m_i} - \frac{\sum_{\boldsymbol{m_j} \in C_{o(i)}} \boldsymbol{m_j}}{|C_{o(i)}|}||, 0)$$

The first of these losses pulls together mentions that belong to the same entity. The second is used to pull further apart mentions that belong to different clusters by repelling each mention embedding from the mean of another random cluster if the distance is closer than some threshold $\mathcal{T}$, which is picked based on the development set's performance. After obtaining the mention embeddings, we utilize agglomerative clustering (Murtagh and Legendre, 2011) to obtain the actual entity or event clusters.

While previous work has found un-tuned pretrained language model embeddings can achieve good results for document-level coreference resolution (Jain et al., 2020), this method is insufficient for pronoun coreference resolution, as they don't capture enough contextual information to differentiate between similar pronouns that refer to separate entities.

2.5 Cluster-based Information Aggregation

Given the predicted clusters, we produce a representation for each entity or event cluster, which will be later used for entity and event type prediction. In order to obtain the representation, we first pass each mention representation through a residual layer. Afterward max pooling is performed in order to obtain the final cluster representation. The overall architecture for constructing mention representations, as well as aggregating mentions into a cluster representation is shown in Figure 2. Aggregating information in this way helps the model classify entities and events in situations where sentences might not provide the necessary context, such as the one presented in Figure 3. The final class scores are obtain by passing this final representation through a 2-layer linear network.

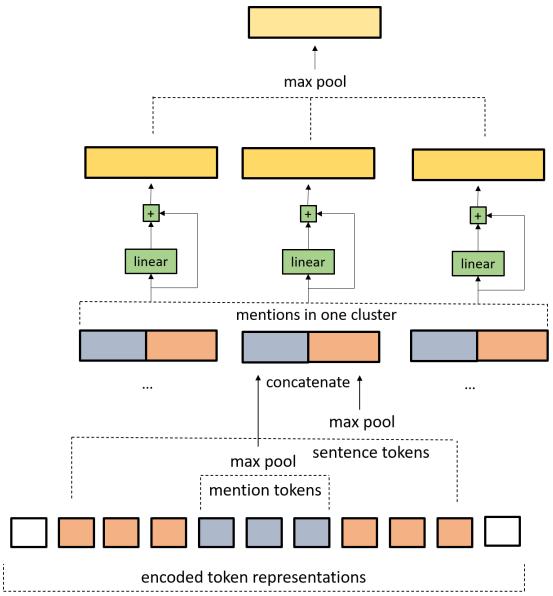

Figure 2: Method for constructing cluster representations by aggregating mentions

3 Experiments

3.1 Dataset

For training and evaluation we use documents from the ACE05-E$^+$ dataset (Lin et al., 2020), which consist of up to 2000 tokens with entity, event and relation annotations. This dataset was introduced as a modified version of the ACE05-E dataset, which adds pronoun mention annotations as well as multi-token triggers, and has the following statistics:

Split	Docs	Entities	Events
Training	599	47,525	4,419
Development	28	3,422	468
Test	40	3,673	424

Table 1: ACE05-E$^+$ dataset statistics

We chose this particular configuration of the dataset for our experiments due to the large amount of annotated pronoun mentions, which can be par-

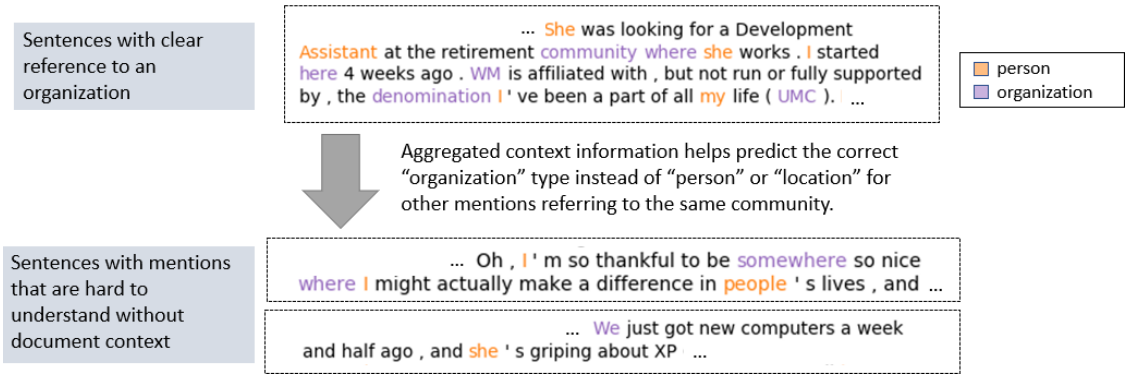

Figure 3: Excerpt from an ACE05-E$^+$ document where access to the surrounding context can be helpful for determining mention types. Mentions are colored to represent types.

ticularly difficult to classify correctly without access to external context.

3.2 Evaluation

Similar to previous work (Zhang et al., 2019; Wadden et al., 2020; Lin et al., 2019), we evaluate detection of entities and event triggers as follows: an entity or event trigger mention is considered to be correctly identified (Trig-I) if both of the offsets are correctly matched, and out of those mentions the ones with the correctly predicted type are considered correctly classified (Entities-C, Trig-C). We compare the full model with the OneIE (Lin et al., 2020) baseline, as well as with variants of our model without additional GloVe embeddings and without aggregation of information between mentions. We also calculate results for our model given gold mention and cluster information. We measure the classification F-score for entities, and the identification and classification F-scores for event triggers. Overall these results, presented in Table 2, demonstrate that document-level context aggregation can improve entity and event detection.

We utilize a multi-step system, where the input of the next step can depend on the outputs of previous steps. This leads to error accumulation, making it hard to determine which modules are working well and which aren't from the final results alone. In order to better understand how much error accumulation occurs at the coreference resolution stage of the model, we also perform evaluation of the produced entity and event trigger mention clusters using two metrics. The first is B^3_{sys} (Cai and Strube, 2010). This metric is a modification of B^3, modified to properly account for system-predicted mentions (as opposed to coreference resolution performed on gold-standard mentions). We base the second metric on "matching" predicted clusters to gold clusters. The cluster matching is performed with the following steps:

Model	Entities-C	Trig-I	Trig-C
OneIE	89.6	75.6	72.8
Our method			
Full Model	**91.96**	**77.67**	**75.06**
- GloVe	91.94	76.69	74.07
- aggregation	91.03	77.32	73.74
+ gold inputs			
mentions	95.97	-	92.69
clusters	97.58	-	94.25

Table 2: Entity and Event Trigger Extraction Results on ACE05-E$^+$ (F-score, %)

1. First, match predicted mentions to golden ones based on the mention span start and end.

2. For each predicted cluster, we check if there exists a gold cluster such that over half of the predicted cluster mentions are matched to over half gold cluster mentions.

3. We compute F-score based on the predicted clusters, gold clusters, and matched clusters based on previous step.

The matching metric is useful as it tells us the amount of entity and event clusters for which our information aggregation approach has the potential to work well. Since more than half of the mentions in a cluster are checked, this metric also has the advantage of only matching at most one predicted

cluster to at most one gold cluster. The results for coreference resolution are presented in Table 3.

Metric	Precision	Recall	F
Entities			
B^3_{sys}	83.5	86.2	84.83
Matching	70.76	72.05	71.40
Event Triggers			
B^3_{sys}	76.56	77.57	77.06
Matching	47.16	56.06	51.23

Table 3: Coreference Resolution Results on ACE05-E$^+$ (%)

4 Related Work

An earlier CRF-based work by Durrett and Klein (2014) shows benefits from joint modeling of coreference resolution across a document, named entity recognition and entity linking, and notes that propagating information between different mentions of an entity in a document can help resolve ambiguous cases of semantic types or entity links.

In previous neural models similar ideas of using document-level contextual information in order to improve typing of entities have been considered (Zhang et al., 2020a). The authors of this work apply an attention mechanism in order to aggregate information between different mentions of the same underlying entity. In contrast with our proposed method, instead of jointly performing coreference resolution, this model only considers mentions with exactly matching strings, which significantly limits the effectiveness of their approach.

Jain et al. (2020) introduce a new document IE dataset, as well as a baseline model which also involves aggregation of information between mentions. However, here mention typing is performed before aggregation, and the cluster representation is instead used for other tasks, such as relation extraction. Another dataset with document-level annotation is RAMS (Ebner et al., 2020), which contains event arguments annotated in a 5-sentence window around each trigger in the documents. Several approaches have been suggested for this task. For example, Zhang et al. (2020b) introduce a two-step process for extracting event arguments, which consists of first detecting the first token, and then expanding to the entire span. Chen et al. (2020) propose to link events to their arguments by feeding each section of a document through BERT, and then processing the mention representations for triggers and potential arguments with another Transformer.

Recently another dataset for multi-task IE was introduced by Zaporojets et al. (2021), with particular focus on entities with mentions in different parts of a document. The authors also propose a baseline model for this dataset, which uses a neural graph-based message passing approach in order to aggregate document-level information.

5 Conclusions and Future Work

Aggregating information across an entire document can be highly effective for classifying entity and event mention types. This is particularly useful in cases where pronouns are used to refer to entities or events that are not explained within the same sentence. In the future, we plan to extend our approach to use document-level context for extraction of relations between entities and event arguments.

Acknowledgement

This research is based upon work supported in part by U.S. DARPA KAIROS Program No. FA8750-19-2-1004, U.S. DARPA AIDA Program No. FA8750-18-2-0014, and Air Force No. FA8650-17-C-7715. 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 DARPA, 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

Masaki Asada, Makoto Miwa, and Yutaka Sasaki. 2020. Using Drug Descriptions and Molecular Structures for Drug-Drug Interaction Extraction from Literature. *Bioinformatics*. Btaa907.

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

Jie Cai and Michael Strube. 2010. Evaluation metrics for end-to-end coreference resolution systems. pages 28–36.

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

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Markus Eberts and Adrian Ulges. 2019. Span-based joint entity and relation extraction with transformer pre-training.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. Scirex: A challenge dataset for document-level information extraction.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2019. A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 65–71, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint end-to-end neural model for information extraction with global features. In *Proc. The 58th Annual Meeting of the Association for Computational Linguistics (ACL2020)*.

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.

Fionn Murtagh and Pierre Legendre. 2011. Ward's hierarchical clustering method: Clustering criterion and agglomerative algorithm.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William Wang. 2019. Fine-tune bert for docred with two-step process.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset.

Klim Zaporojets, Johannes Deleu, Chris Develder, and Thomas Demeester. 2021. Dwie: an entity-centric dataset for multi-task document-level information extraction.

Boliang Zhang, Spencer Whitehead, Lifu Huang, and Heng Ji. 2020a. Global attention for name tagging.

Xiang Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2019. AdaNSP: Uncertainty-driven adaptive decoding in neural semantic parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4265–4270, Florence, Italy. Association for Computational Linguistics.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020b. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. Association for Computational Linguistics.