Document-Level Relation Extraction with Adaptive Thresholding and Localized Context Pooling

Wenxuan Zhou[1], **Kevin Huang**[2], **Tengyu Ma**[3], **Jing Huang**[2]
[1]University of Southern California [2]JD AI Research [3]Stanford University
[1]zhouwenx@usc.edu [2]{kevin.huang, jing.huang}@jd.com [3]tengyuma@stanford.edu

Abstract

Document-level relation extraction (RE) poses new challenges compared to its sentence-level RE counterpart. One document commonly contains multiple entity pairs, and one entity pair occurs multiple times in the document associated with multiple possible relations. In this paper, we propose two novel techniques, adaptive thresholding and localized context pooling, to solve the multi-label and multi-entity problems. The adaptive thresholding replaces the global threshold for multi-label classification in the prior work by a learnable entities-dependent threshold. The localized context pooling directly transfers attention from pre-trained language models to locate relevant context that is useful to decide the relation. We experiment on three document-level RE benchmark datasets: DocRED, a recently released large-scale RE dataset, and two datasets CDR and GDA in the biomedical domain. Our ATLOP[1] (**A**daptive **T**hresholding and **L**ocalized c**O**ntext **P**ooling) model achieves an F1 score of 63.4; and also significantly outperforms existing models on both CDR and GDA.

1 Introduction

Relation extraction (RE) aims to identify the relationship between two entities in a given text and plays an important role in information extraction. Existing work mainly focuses on sentence-level relation extraction, i.e., predicting the relationship between entities in a single sentence (Zeng et al., 2014; Miwa and Bansal, 2016; Zhang et al., 2018). However, large amounts of relationships, such as relational facts from Wikipedia articles and biomedical literature, are expressed by multiple sentences in real-world applications (Verga et al., 2018; Yao et al., 2019). This problem, commonly referred to as document-level relation extraction, necessitates models that can capture complex interactions among entities in the whole document.

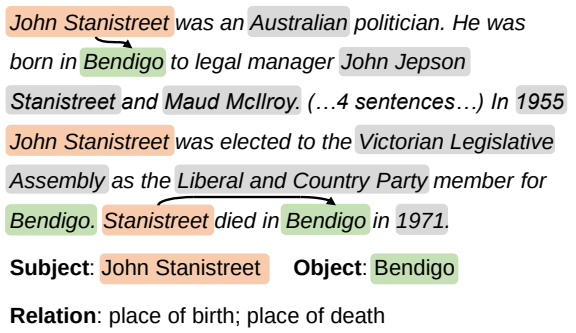

Figure 1: An example of multi-entity and multi-label problems from the DocRED dataset. Subject entity *John Stanistreet* and object entity *Bendigo* express relations *place of birth* and *place of death*. The related entity mentions are connected by lines. Other entities in the document are highlighted in grey.

Compared to sentence-level RE, document-level RE poses unique challenges. For sentence-level RE datasets such as TACRED (Zhang et al., 2017) and SemEval 2010 Task 8 (Hendrickx et al., 2009), a sentence only contains one entity pair to classify. On the other hand, for document-level RE, one document contains multiple entities pairs, and we need to classify the relations of them all at once. It requires the RE model to identity and focus on the part of the document with relevant context for a particular entity pair. In addition, one entity pair can occur many times in the document associated with distinct relations for document-level RE, in contrast to one relation per entity pair for sentence-level RE. This multi-entity (multiple entity pairs to classify in a document) and multi-label (multiple relation types for a particular entity pair) properties of document-level relation extraction make it

This work was conducted while the first author was doing internship at JD AI Research.

[1]Code released at https://github.com/wzhouad/ATLOP.

harder than its sentence-level counterpart. Figure 1 shows an example from the DocRED dataset (Yao et al., 2019). The task is to classify the relation types of pairs of entities (highlighted in color). For a particular entity pair (*John Stanistreet*, *Bendigo*), it expresses two relations *place of birth* and *place of death* by the first two sentences and the last sentence. Other sentences contain irrelevant information to this entity pair.

To tackle the multi-entity problem, most current approaches construct a document graph with dependency structures, heuristics, or structured attention (Peng et al., 2017; Liu and Lapata, 2018; Christopoulou et al., 2019; Nan et al., 2020), and then perform inference with graph neural models (Liang et al., 2016; Guo et al., 2019). The constructed graphs bridge entities that spread far apart in the document and thus alleviate the deficiency of RNN-based encoders (Hochreiter and Schmidhuber, 1997; Chung et al., 2014) in capturing long-distance information (Khandelwal et al., 2018). However, as transformer-based models (Vaswani et al., 2017) can implicitly model long-distance dependencies (Clark et al., 2019; Tenney et al., 2019), it is unclear whether graph structures still help on top of pre-trained language models such as BERT (Devlin et al., 2019). There have also been approaches to directly apply pre-trained language models without introducing graph structures (Wang et al., 2019b; Tang et al., 2020a). They simply average the embedding of entity tokens to obtain the entity embeddings and feed them into the classifier to get relation labels. However, each entity has the same representation in different entity pairs, which can bring noise from irrelevant context.

In this paper, instead of introducing graph structures, we propose a localized context pooling technique. This technique solves the problem of using the same entity embedding for all entity pairs. It enhances the entity embedding with additional context that is relevant to the current entity pair. Instead of training a new context attention layer from scratch, we directly transfer the attention heads from pre-trained language models to get entity-level attention. Then, for two entities in a pair, we merge their attentions by multiplication to find the context that is important to both of them.

For the multi-label problem, existing approaches reduce it to a binary classification problem. After training, a global threshold is applied to the class probabilities to get relation labels. This method involves heuristic threshold tuning and introduces decision errors when the tuned threshold from development data may not be optimal for all instances.

In this paper, we propose the adaptive thresholding technique, which replaces the global threshold with a learnable threshold class. The threshold class is learned with our adaptive-threshold loss, which is a *rank-based* loss that pushes the logits of positive classes above the threshold and pulls the logits of negative classes below in model training. At the test time, we return classes that have higher logits than the threshold class as the predicted labels or return NA if such class does not exist. This technique eliminates the need for threshold tuning, and also makes the threshold adjustable to different entity pairs, which leads to much better results.

By combining the proposed two techniques, we propose a simple yet effective relation extraction model, named ATLOP (**A**daptive **T**hresholding and **L**ocalized c**O**ntext **P**ooling), to fully utilize the power of pre-trained language models (Devlin et al., 2019; Liu et al., 2019). Experiments on three document-level relation extraction datasets, DocRED (Yao et al., 2019), CDR (Li et al., 2016), and GDA (Wu et al., 2019b), demonstrate that our ATLOP model significantly outperforms the state-of-the-art methods. The contributions of our work are summarized as follows:

- We propose adaptive-thresholding loss, which enables the learning of an adaptive threshold that is dependent on entity pairs and reduces the decision errors caused by using a global threshold.
- We propose localized context pooling, which transfers pre-trained attention to grab related context for entity pairs to get better entity representations.
- We conduct experiments on three public document-level relation extraction datasets. Experimental results demonstrate the effectiveness of our ATLOP model that achieves the new state-of-the-art performance on the three benchmark datasets.

2 Problem Formulation

Given a document d and a set of entities $\{e_i\}_{i=1}^{n}$, the task of document-level relation extraction is to predict a subset of relations from $\mathcal{R} \cup \{\text{NA}\}$ between the entity pairs $(e_s, e_o)_{s,\, o=1...n;\, s \neq o}$, where

$\mathcal{R}$ is a pre-defined set of relations of interest, e_s, e_o are identified as subject and object entities, respectively. An entity e_i can occur multiple times in the document by entity mentions $\{m_j^i\}_{j=1}^{N_{e_i}}$. A relation exists between entities (e_s, e_o) if it is expressed by any pair of their mentions. The entity pairs that do not express any relation are labeled NA. At the test time, the model needs to predict the labels of all entity pairs $(e_s, e_o)_{s,\, o=1...n;\, s\neq o}$ in document d.

3 Enhanced BERT Baseline

In this section, we present our base model for document-level relation extraction. We build our model based on existing BERT baselines (Yao et al., 2019; Wang et al., 2019b) and integrate other techniques to further improve the performance.

3.1 Encoder

Given a document $d = [x_t]_{t=1}^{l}$, we mark the position of entity mentions by inserting a special symbol "*" at the start and end of mentions. It is adapted from the commonly-used entity marker technique (Zhang et al., 2017; Shi and Lin, 2019; Soares et al., 2019). We then feed the document into a pre-trained language model to obtain the contextual embeddings:

$$[\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_l] = \text{BERT}([x_1, x_2, ..., x_l]). \quad (1)$$

Following previous work (Verga et al., 2018; Wang et al., 2019a), the document is encoded once by the encoder, and the classification of all entity pairs is based on the same contextual embedding. We take the embedding of "*" at the start of mentions as the mention embeddings. For an entity e_i with mentions $\{m_j^i\}_{j=1}^{N_{e_i}}$, we apply logsumexp pooling (Jia et al., 2019), a smooth version of max pooling, to get the entity embedding $\boldsymbol{h}_{e_i}$.

$$\boldsymbol{h}_{e_i} = \log \sum_{j=1}^{N_{e_i}} \exp\left(\boldsymbol{h}_{m_j^i}\right). \quad (2)$$

3.2 Binary Classifier

Given the embedding $(\boldsymbol{h}_{e_s}, \boldsymbol{h}_{e_o})$ of an entity pair e_s, e_o computed by equation (2), we map the entities to hidden states $\boldsymbol{z}$ with a linear layer followed by non-linear activation, then calculate the probability of relation r by bilinear function and sigmoid activation. This process is formulated as:

$$\boldsymbol{z}_s = \tanh\left(\boldsymbol{W}_s \boldsymbol{h}_{e_s}\right), \quad (3)$$

$$\boldsymbol{z}_o = \tanh\left(\boldsymbol{W}_o \boldsymbol{h}_{e_o}\right), \quad (4)$$

$$\mathrm{P}\left(r|e_s, e_o\right) = \sigma\left(\boldsymbol{z}_s^{\intercal} \boldsymbol{W}_r \boldsymbol{z}_o + b_r\right),$$

where $\boldsymbol{W}_s \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_o \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_r \in \mathbb{R}^{d\times d}, b_r \in \mathbb{R}$ are model parameters. The representation of one entity is the same among different entity pairs. To reduce the number of parameters in the bilinear classifier, we use the group bilinear (Zheng et al., 2019), which splits the embedding dimensions into k equal-sized groups (Tang et al., 2020b) and applies bilinear within the groups:

$$\left[\boldsymbol{z}_s^1; ...; \boldsymbol{z}_s^k\right] = \boldsymbol{z}_s,$$

$$\left[\boldsymbol{z}_o^1; ...; \boldsymbol{z}_o^k\right] = \boldsymbol{z}_o,$$

$$\mathrm{P}\left(r|e_s, e_o\right) = \sigma\left(\sum_{i=1}^{k} \boldsymbol{z}_s^{i\intercal} \boldsymbol{W}_r^i \boldsymbol{z}_o^i + b_r\right), \quad (5)$$

where $\boldsymbol{W}_r^i \in \mathbb{R}^{d/k\times d/k}$ for $i = 1...k$ are model parameters, $\mathrm{P}\left(r|e_s, e_o\right)$ is the probability that relation r is associated with the entity pair (e_s, e_o). In this way, we can reduce the number of parameters from d^2 to d^2/k. We use the binary cross entropy loss for training. During inference, we tune a global threshold θ that maximizes evaluation metrics (F_1 score for RE) on the development set and return r as an associated relation if $\mathrm{P}\left(r|e_s, e_o\right) > \theta$ or return NA if no relation exists.

Our enhanced base model achieves near state-of-the-art performance in our experiments, significantly outperforms existing BERT baselines.

4 Adaptive Thresholding

The RE classifier outputs the probability $\mathrm{P}\left(r|e_s, e_o\right)$ within the range $[0, 1]$, which needs thresholding to be converted to relation labels. As the threshold neither has a closed-form solution nor is differentiable, a common practice for deciding threshold is enumerating several values in the range $(0, 1)$ and picking the one that maximizes the evaluation metrics (F_1 score for RE). However, the model may have different confidence for different entity pairs or classes in which one global threshold does not suffice. The number of relations varies (multi-label problem) and the models may not be globally calibrated so that the same probability does not mean the same for all

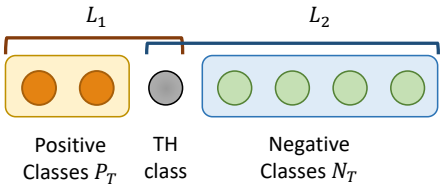

Figure 2: An artificial illustration of our proposed adaptive-thresholding loss. A TH class is introduced to separate positive classes and negative classes: positive classes would have higher probabilities than TH, and negative classes would have lower probabilities than TH.

entity pairs. This problem motivates us to replace the global threshold with a learnable, adaptive one, which can reduce decision errors during inference.

For the convenience of explanation, we split the labels of entity pair $T = (e_s, e_o)$ into two subsets: positive labels $\mathcal{P}_T$ and negative labels $\mathcal{N}_T$, which are defined as follows:

- Positive labels $\mathcal{P}_T \subseteq \mathcal{R}$ are the relations that exist between the entities in T. If T does not express any relation, $\mathcal{P}_T$ is empty.
- Negative labels $\mathcal{N}_T \subseteq \mathcal{R}$ are the relations that do not exist between the entities. If T does not express any relation, $\mathcal{N}_T = \mathcal{R}$.

If an entity pair is classified correctly, the logits of positive labels should be higher than the threshold while those of negative labels should be lower. Here we introduce a threshold class TH, which is automatically learned in the same way as other classes (see Eq.(5)). At the test time, we return classes with higher logits than the TH class as positive labels or return NA if such classes do not exist. This threshold class learns an entities-dependent threshold value. It is a substitute for the global threshold and thus eliminates the need for tuning threshold on the development set.

To learn the new model, we need a special loss function that considers the TH class. We design our adaptive-thresholding loss based on the standard categorical cross entropy loss. The loss function is broken down to two parts as shown below:

$$\mathcal{L}_1 = -\sum_{r \in \mathcal{P}_T} \log \left(\frac{\exp\left(\text{logit}_r\right)}{\sum_{r' \in \mathcal{P}_T \cup \{\text{TH}\}} \exp\left(\text{logit}_{r'}\right)} \right),$$

$$\mathcal{L}_2 = -\log \left(\frac{\exp\left(\text{logit}_{\text{TH}}\right)}{\sum_{r' \in \mathcal{N}_T \cup \{\text{TH}\}} \exp\left(\text{logit}_{r'}\right)} \right),$$

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2.$$

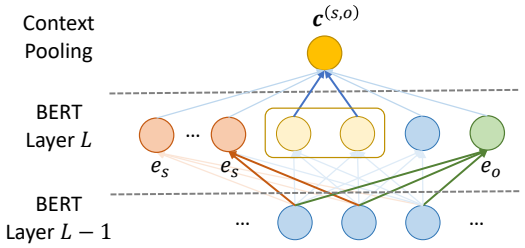

Figure 3: Illustration of localized context pooling. Tokens are weighted averaged to form the localized context $\boldsymbol{c}^{(s,o)}$ of the entity pair (e_s, e_o). The weights of tokens are derived by multiplying the attention weights of the subject entity e_s and the object entity e_o from the last transformer layer so that only the tokens that are important to both entities (highlighted in light yellow) receive higher weights.

The first part $\mathcal{L}_1$ involves positive labels and the TH class. Since there may be multiple positive labels, the total loss is calculated as the sum of categorical cross entropy losses on all positive labels (Menon et al., 2019; Reddi et al., 2019). $\mathcal{L}_1$ pushes the logits of all positive labels to be higher than the TH class. It is not used if there is no positive label. The second part $\mathcal{L}_2$ involves the negative classes and threshold class. It is a categorical cross entropy loss with TH class being the true label. It pulls the logits of negative labels to be lower than the TH class. Two parts are simply summed for the total loss.

The proposed adaptive-thresholding loss is illustrated in Figure 2. It obtains a large performance gain to the global threshold in our experiments.

5 Localized Context Pooling

The logsumexp pooling (see Eq. (2)) accumulates the embedding of all mentions for an entity across the whole document and generates one embedding for this entity. The entity embedding is then used in the classification of all entity pairs. However, since some context may express relations unrelated to the entity pair, it is better to have a localized representation that only attends to the relevant context in the document that is useful to decide to relation(s) for the entity pair.

Therefore we propose the localized context pooling, where we enhance the embedding of an entity pair with an additional context embedding that is related to both entities. In this work, since we use pretrained transformer-based models as the encoder, which has already learned token-level dependencies by multi-head self-attention (Vaswani et al.,

2017), we consider directly using their attention heads for localized context pooling. This method transfers the well-learned dependencies from the pre-trained language model without learning new attention layers from scratch.

Specifically, we use the token-level attention heads $\boldsymbol{A}$ from the last transformer layer in the pre-trained language model, where $\boldsymbol{A}_{ijk,1\le i\le H,1\le j,k\le l}$ represents the importance of token k for token j in the i^{th} attention head. For entity mention that spans from the j' th token ("*" symbol), we take $\boldsymbol{A}_{j=j'}$ as the mention-level attention, then average the attention over mentions of the same entity to obtain entity-level attentions $\{\boldsymbol{A}_i^E\}_{i=1}^m$, where each attention $\boldsymbol{A}_i^E \in \mathbb{R}^{H\times l}$ denotes the importance of context tokens to the i^{th} entity in H attention heads. Then for entity pair (e_s, e_o), we obtain the context tokens that are important to both entities by multiplying their entity-level attentions followed by normalization:

$$\begin{aligned}
\boldsymbol{A}^{(s,o)} &= \boldsymbol{A}_s^E \cdot \boldsymbol{A}_o^E, \\
\boldsymbol{q}^{(s,o)} &= \sum_{i=1}^{H} \boldsymbol{A}_i^{(s,o)}, \\
\boldsymbol{a}^{(s,o)} &= \boldsymbol{q}^{(s,o)} / \mathbf{1}^\intercal \boldsymbol{q}^{(s,o)}, \\
\boldsymbol{c}^{(s,o)} &= \boldsymbol{H}^\intercal \boldsymbol{a}^{(s,o)},
\end{aligned}$$

where $\boldsymbol{c}^{(s,o)}$ is the localized contextual embedding for (e_s, e_o). The contextual embedding is fused into the pooled entity embedding to obtain entity representations that are different for different entity pairs, by modifying the original linear layer in Eq. (3) and Eq. (4) as follows:

$$\boldsymbol{z}_s^{(s,o)} = \tanh\left(\boldsymbol{W}_s \boldsymbol{h}_{e_s} + \boldsymbol{W}_{c_1} \boldsymbol{c}^{(s,o)}\right), \quad (6)$$

$$\boldsymbol{z}_o^{(s,o)} = \tanh\left(\boldsymbol{W}_o \boldsymbol{h}_{e_o} + \boldsymbol{W}_{c_2} \boldsymbol{c}^{(s,o)}\right), \quad (7)$$

where $\boldsymbol{W}_{c_1}, \boldsymbol{W}_{c_2} \in \mathbb{R}^{d\times d}$ are model parameters. The proposed localized context pooling is illustrated in Figure 3.

6 Experiments

6.1 Datasets

We evaluate our ATLOP model on three public document-level relation extraction datasets. The dataset statistics are shown in Table 1.

- **DocRED** (Yao et al., 2019) is a large-scale general-purpose dataset for document-level RE constructed from Wikipedia articles. It consists of 3053 human-annotated documents for training. For entity pairs that express relation(s), about 7% of them have more than one relation label.

- **CDR** (Li et al., 2016) is a human-annotated dataset in the biomedical domain. It consists of 500 documents for training. The task is to predict the binary interactions between Chemical and Disease concepts.

- **GDA** (Wu et al., 2019b) is a large-scale dataset in the biomedical domain. It consists of 29192 articles for training. The task is to predict the binary interactions between Gene and Disease concepts. We follow Christopoulou et al. (2019) to split the training set into an 80/20 split as training and development sets.

Statistics / Dataset	DocRED	CDR	GDA
# Train	3053	500	23353
# Dev	1000	500	5839
# Test	1000	500	1000
# Relations	97	2	2
Avg.# entities per Doc.	19.5	7.6	5.4

Table 1: Statistics of the datasets in experiments.

Hyperparam / Dataset	DocRED	CDR	GDA
Batch size	4	4	16
# Epoch	30	30	10
Learning rate for encoder	5e-5	2e-5	2e-5
Learning rate for classifier	1e-4	1e-4	1e-4
Group size	64	64	64
Dropout	0.1	0.1	0.1
Gradient clipping	1.0	1.0	1.0

Table 2: Hyper-parameters of ATLOP.

6.2 Experiment Settings

Our model is implemented based on Pytorch[2] and Huggingface's Transformers[3]. We use cased BERT-base (Devlin et al., 2019) or RoBERTa-large (Liu et al., 2019) as the encoder on DocRED, and cased SciBERT-base (Beltagy et al., 2019) on CDR and GDA. We use mixed precision training (Micikevicius et al., 2018) based on the Apex library[4]. Our model is optimized with AdamW (Loshchilov and Hutter, 2019) using learning rates $\in \{2\text{e}{-5}, 3\text{e}{-5}, 5\text{e}{-5}, 1\text{e}{-4}\}$, with a lin-

[2] https://pytorch.org/

[3] https://github.com/huggingface/transformers

[4] https://github.com/NVIDIA/apex

Model	Dev		Test	
	Ign F_1	F_1	Ign F_1	F_1
Sequence-based Models				
CNN (Yao et al., 2019)	41.58	43.45	40.33	42.26
BiLSTM (Yao et al., 2019)	48.87	50.94	48.78	51.06
Graph-based Models				
BiLSTM-AGGCN (Guo et al., 2019)	46.29	52.47	48.89	51.45
BiLSTM-LSR (Nan et al., 2020)	48.82	55.17	52.15	54.18
BERT-LSR$_{\text{BASE}}$ (Nan et al., 2020)	52.43	59.00	56.97	59.05
Transformer-based Models				
BERT$_{\text{BASE}}$ (Wang et al., 2019b)	-	54.16	-	53.20
BERT-TS$_{\text{BASE}}$ (Wang et al., 2019b)	-	54.42	-	53.92
HIN-BERT$_{\text{BASE}}$ (Tang et al., 2020a)	54.29	56.31	53.70	55.60
CorefBERT$_{\text{BASE}}$ (Ye et al., 2020)	55.32	57.51	54.54	56.96
CorefRoBERTa$_{\text{LARGE}}$ (Ye et al., 2020)	57.84	59.93	57.68	59.91
Our Methods				
BERT$_{\text{BASE}}$ (our implementation)	54.27 ± 0.28	56.39 ± 0.18	-	-
BERT-E$_{\text{BASE}}$	56.51 ± 0.16	58.52 ± 0.19	-	-
BERT-ATLOP$_{\text{BASE}}$	59.22 ± 0.15	61.09 ± 0.16	59.31	61.30
RoBERTa-ATLOP$_{\text{LARGE}}$	$\mathbf{61.32 \pm 0.14}$	$\mathbf{63.18 \pm 0.19}$	**61.39**	**63.40**

Table 3: Main results (%) on the development and test set of DocRED. We report the mean and standard deviation of F_1 on the development set by conducting 5 runs of training using different random seeds.

Model	CDR	GDA
BRAN (Verga et al., 2018)	62.1	-
CNN (Nguyen and Verspoor, 2018)	62.3	-
EoG (Christopoulou et al., 2019)	63.6	81.5
LSR (Nan et al., 2020)	64.8	82.2
SciBERT$_{\text{BASE}}$ (our implementation)	65.1 ± 0.6	82.5 ± 0.3
SciBERT-E$_{\text{BASE}}$	65.9 ± 0.5	83.3 ± 0.3
SciBERT-ATLOP$_{\text{BASE}}$	$\mathbf{69.4 \pm 1.1}$	$\mathbf{83.9 \pm 0.2}$

Table 4: Test F_1 score (in %) on CDR and GDA dataset. Our ATLOP model with the SciBERT encoder outperforms the current SOTA results.

ear warmup (Goyal et al., 2017) for the first 6% steps followed by a linear decay to 0. All hyper-parameters are tuned on the development set. We list the hyper-parameters on all datasets in Table 2. For models that use a global threshold, we search threshold values from $\{0.1, 0.2, ..., 0.9\}$ and pick the one that maximizes dev F_1.

6.3 Main Results

We compare ATLOP with sequence-based models, graph-based models, and transformer-based models on the DocRED dataset. The experiment results are shown in Table 3. Following Yao et al. (2019), we use F_1 and Ign F_1 in evaluation. The Ign F_1 denotes the F_1 score excluding the relational facts that are shared by the training and dev/test sets.

Sequence-based Models. These models use neural architectures such as CNN (LeCun et al., 2015) and bidirectional LSTM (Schuster and Paliwal, 1997) to encode the entire document, then obtain entity embeddings and predict relations for each entity pair with bilinear function.

Graph-based Models. These models construct document graphs by learning latent graph structures of the document and perform inference with graph neural network (Kipf and Welling, 2017). We include two state-of-the-art graph-based models, AGGCN (Guo et al., 2019) and LSR (Nan et al., 2020), for comparison. The result of AGGCN is from the re-implementation by Nan et al. (2020).

Transformer-based Models. These models adapt pre-trained language models to document-level RE without using graph structures. They can be further divided into pipeline models (BERT-TS (Wang et al., 2019b)), hierarchical models (HIN-BERT (Tang et al., 2020a)), and pre-training methods (CorefBERT and CorefRoBERTa (Ye et al., 2020)). We also include BERT baseline (Wang et al., 2019b) in our comparison.

We find that our re-implemented BERT baseline gets significantly better results than Wang et al. (2019b), and outperforms the state-of-the-art RNN-based model BiLSTM-LSR by 1.2%. It demonstrates that pre-trained language models can capture long-distance dependencies among entities without explicitly using graph structures. After integrating other techniques, our enhanced baseline BERT-E$_{\text{BASE}}$ achieves an F1 score of 58.52%,

Model	Ign F_1	F_1
BERT-ATLOP$_{\text{BASE}}$	59.22	61.09
– Adaptive Thresholding	58.32	60.20
– Localized Context Pooling	58.19	60.12
– Adaptive-Thresholding Loss	39.52	41.74
BERT-E$_{\text{BASE}}$	56.51	58.52
– Entity Marker	56.22	58.28
– Group Bilinear	55.51	57.54
– Logsumexp Pooling	55.35	57.40

Table 5: Ablation study of ATLOP on DocRED. We turn off different components of the model one at a time. These ablation results show that both adaptive thresholding and localized context pooling are effective. Logsumexp pooling and group bilinear both bring noticeable gain to the baseline.

which is close to the current state-of-the-art model BERT-LSR$_{\text{BASE}}$. Our BERT-ATLOP$_{\text{BASE}}$ model further improves the performance of BERT-E$_{\text{BASE}}$ by 2.6%, demonstrating the efficacy of the proposed two novel techniques. Using RoBERTa-large as the encoder, our ALTOP model achieves an F1 score of 63.40%, which is a new state-of-the-art result on DocRED. We held the first position on Colab leaderboard[5] as of September 1st, 2020.

6.4 Results on Biomedical Datasets

Experiment results on two biomedical datasets are shown in Table 4. Verga et al. (2018) and Nguyen and Verspoor (2018) are both sequence-based models that use self attention network and CNN as the encoders, respectively. Christopoulou et al. (2019) and Nan et al. (2020) use graph-based models that construct document graphs by heuristics or structured attention, and perform inference with graph neural network. To our best knowledge, transformer-based pre-trained language models have not been applied to document-level RE datasets in the biomedical domain. In experiments, we replace the encoder with SciBERT$_{\text{BASE}}$, which is pre-trained on multi-domain corpora of scientific publications. The SciBERT$_{\text{BASE}}$ baseline already outperforms all existing methods. Our SciBERT-ATLOP$_{\text{BASE}}$ model further improves the F_1 score by 4.3% and 1.4% on CDR and GDA, respectively, and yields the new state-of-the-art results on these two datasets.

[5] https://competitions.codalab.org/competitions/20717

Strategy	Dev F_1	Test F_1
Global Thresholding	60.14	60.62
Per-class Thresholding	**61.73**	60.35
Adaptive Thresholding	61.27	**61.30**

Table 6: Result of different thresholding strategies on DocRED. Our adaptive thresholding consistently outperforms other strategies on the test set.

6.5 Ablation Study

To show the efficacy of our proposed techniques, we conduct two sets of ablation studies on ATLOP and enhanced baseline, by turning off one component at a time. We observe that all components contribute to model performance. The adaptive thresholding and localized context pooling are equally important to model performance, leading to a drop of 0.89% and 0.97% in dev F_1 score respectively when removed from ATLOP. Note that the adaptive thresholding only works when the model is optimized with the adaptive-thresholding loss. Applying adaptive thresholding to models trained with binary cross entropy results in dev F_1 of 41.74%.

For our enhanced baseline model BERT-E$_{\text{BASE}}$, both group bilinear and logsumexp pooling lead to about 1% increase in dev F_1. We find the improvement from entity markers is minor (0.24% in dev F_1) but still use the technique in the model as it makes the derivation of mention embedding and mention-level attention easier.

6.6 Analysis of Thresholding

Global thresholding does not consider the variations of model confidence in different classes or instances, and thus yields suboptimal performance. One interesting problem is whether we can improve global thresholding by tuning different thresholds for different classes. Thus we experiment on tuning class-dependent thresholds to maximize the F_1 score on the development set of DocRED using the cyclic optimization algorithm (Fan and Lin, 2007). Results are shown in Table 6. We find that using per-class thresholding significantly improves the dev F_1 score to 61.73%, which is even higher than the result of adaptive thresholding. However, this gain does not transfer to the test set. The result of per-class thresholding is even worse than that of global thresholding. While our adaptive thresholding technique uses a learnable threshold that can automatically generalize to the test set.

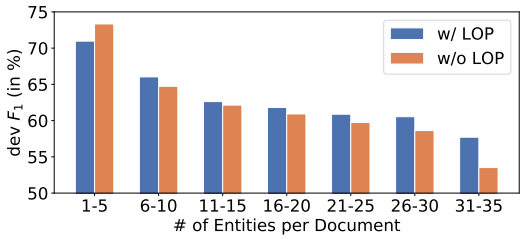

Figure 4: Dev F_1 score of documents with the different number of entities on DocRED. Our localized context pooling achieves better results when the number of entities is larger than 5. The improvement is more significant when the number of entities increases.

John Stanistreet was an Australian politician. He was born in Bendigo to legal manager John Jepson Stanistreet and Maud McIlroy. (… 4 sentences …) In 1955 John Stanistreet was elected to the Victorian Legislative Assembly as the Liberal and Country Party member for Bendigo, but he was defeated in 1958. Stanistreet died in Bendigo in 1971.

Subject: John Stanistreet **Object**: Bendigo

Relation: place of birth; place of death

Figure 5: Context weights of an example from DocRED. We visualize the weight of context tokens $\boldsymbol{a}^{(s,o)}$ in localized context pooling. The model attends to the most relevant context *born* and *died* for entity pair (*John Stanistreet*, *Bendigo*).

6.7 Analysis of Context Pooling

To show that our localized context pooling (LOP) technique mitigates the multi-entity issue, we divide the documents in the development set of DocRED into different groups by the number of entities, and evaluate models trained with or without localized context pooling on each group. Experiment results are shown in Figure 4. We observe that for both models, their performance gets worse when the document contains more entities. The model w/ LOP consistently outperforms the model w/o LOP except when the document contains very few entities (1 to 5), and the improvement gets larger when the number of entities increases. However, the number of documents that only contain 1 to 5 entities is very small (4 in the dev set), and the documents in DocRED contain 19 entities on average. Therefore our localized context pooling still improves the overall F_1 score significantly. This indicates that the localized context pooling technique can capture related context for entity pairs and thus alleviates the multi-entity problem.

We also visualize the context weights of the example in Figure 1. As shown in Figure 5, our localized context pooling gives high weights to *born* and *died*, which are most relevant to both entities (*John Stanistreet*, *Bendigo*). These two tokens are also evidence for the two ground truth relationships *place of birth* and *place of death*, respectively. Tokens like *elected* and *politician* get much smaller weights because they are only related to the subject entity *John Stanistreet*. The visualization demonstrates that the localized context can locate the context that is related to both entities.

7 Related Work

Early research efforts on relation extraction concentrate on predicting the relationship between two entities within a sentence. Various approaches including sequence-based methods (Zeng et al., 2014; Wang et al., 2016; Zhang et al., 2017), graph-based methods (Miwa and Bansal, 2016; Zhang et al., 2018; Guo et al., 2019; Wu et al., 2019a), transformer-based methods (Alt et al., 2019; Shi and Lin, 2019), and pre-training methods (Zhang et al., 2019; Soares et al., 2019) have been shown effective in tackling this problem.

However, as large amounts of relationships are expressed by multiple sentences (Verga et al., 2018; Yao et al., 2019), recent work starts to explore document-level relation extraction. Most approaches on document-level RE are based on document graphs, which were introduced by Quirk and Poon (2017). Specifically, they use words as nodes and inner and inter-sentential dependencies (dependency structures, coreferences, etc.) as edges. This document graph provides a unified way of extracting the features for entity pairs. Later work extends the idea by improving neural architectures (Peng et al., 2017; Verga et al., 2018; Song et al., 2018; Jia et al., 2019; Gupta et al., 2019) or adding more types of edges (Christopoulou et al., 2019; Nan et al., 2020). In particular, Christopoulou et al. (2019) constructs nodes of different granularities (sentence, mention, entity), connects them with heuristically generated edges, and infers the relations with an edge-oriented model (Christopoulou et al., 2018). Nan et al. (2020) treats the document graph as a latent variable and induces it by structured attention (Liu and Lapata, 2018). Their LSR model achieved state-of-the-art performance on document-level RE.

There have also been models that directly apply pre-trained language models without introducing document graphs, since edges such as dependency

structures and coreferences can be automatically learned by pre-trained language models (Clark et al., 2019; Tenney et al., 2019; Vig and Belinkov, 2019; Hewitt and Manning, 2019). In particular, Wang et al. (2019b) proposes a pipeline model that first predicts whether a relationship exists in an entity pair and then predicts the specific relation types. Tang et al. (2020a) proposes a hierarchical model that aggregates entity information from the entity level, sentence level, and document level. Ye et al. (2020) introduces a copy-based training objective to the pre-training stage of language models.

However, none of the models focus on the multi-entity and multi-label problems, which are among the key differences of document-level RE to its sentence-level RE counterpart. Our ATLOP model deals with the problems by two techniques: adaptive thresholding and localized context pooling, and significantly outperforms existing models.

8 Conclusion

In this work, we propose the ATLOP model for document-level relation extraction, which features two novel techniques: adaptive thresholding and localized context pooling. The adaptive thresholding technique replaces the global threshold in multi-label classification with a learnable threshold class that can decide the best threshold for each entity pair. The localized context pooling utilizes pre-trained attention heads to locate relevant context for entity pairs and thus helps in alleviating the multi-entity problem. Experiments on three public document-level relation extraction datasets demonstrate that our ATLOP model significantly outperforms existing models and yields the new state-of-the-art results on all datasets.

References

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Improving relation extraction by pre-trained language representations. *ArXiv*, abs/1906.03088.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP/IJCNLP*.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2018. A walk-based model on entity graphs for relation extraction. *ArXiv*, abs/1902.07023.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *EMNLP/IJCNLP*.

J. Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv*, abs/1412.3555.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention. *ArXiv*, abs/1906.04341.

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

Rong-En Fan and C. Lin. 2007. A study on threshold selection for multi-label classification.

Priya Goyal, P. Dollár, Ross B. Girshick, P. Noordhuis, L. Wesolowski, Aapo Kyrola, Andrew Tulloch, Y. Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677.

Zhijiang Guo, Y. Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *ACL*.

Pankaj Gupta, S. Rajaram, Hinrich Schütze, B. Andrassy, and T. Runkler. 2019. Neural relation extraction within and across sentence boundaries. *ArXiv*, abs/1810.05102.

Iris Hendrickx, S. Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, M. Pennacchiotti, Lorenza Romano, and S. Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *HLT-NAACL 2009*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *NAACL-HLT*.

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

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multiscale representation learning. *ArXiv*, abs/1904.02347.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. *ArXiv*, abs/1805.04623.

Thomas Kipf and M. Welling. 2017. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521:436–444.

J. Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, A. P. Davis, C. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database: The Journal of Biological Databases and Curation*, 2016.

Xiaodan Liang, X. Shen, Jiashi Feng, L. Lin, and S. Yan. 2016. Semantic object parsing with graph lstm. *ArXiv*, abs/1603.07063.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

I. Loshchilov and F. Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

A. Menon, A. Rawat, S. Reddi, and S. Kumar. 2019. Multilabel reductions: what is my loss optimising? In *NeurIPS*.

P. Micikevicius, Sharan Narang, J. Alben, G. Diamos, E. Elsen, D. García, B. Ginsburg, Michael Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. 2018. Mixed precision training. *ArXiv*, abs/1710.03740.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *ArXiv*, abs/1601.00770.

G. Nan, Zhijiang Guo, Ivan Sekulic, and W. Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. *ArXiv*, abs/2005.06312.

Dat Quoc Nguyen and Karin M. Verspoor. 2018. Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings. *ArXiv*, abs/1805.10586.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. *ArXiv*, abs/1609.04873.

S. Reddi, S. Kale, F. Yu, Daniel N. Holtmann-Rice, Jiecao Chen, and S. Kumar. 2019. Stochastic negative mining for learning with large output spaces. *ArXiv*, abs/1810.07076.

Mike Schuster and K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45:2673–2681.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Livio Baldini Soares, N. FitzGerald, Jeffrey Ling, and T. Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *ArXiv*, abs/1906.03158.

Linfeng Song, Yue Zhang, Z. Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph state lstm. In *EMNLP*.

Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shigang Wang, and Pengfei Yin. 2020a. Hin: Hierarchical inference network for document-level relation extraction. *Advances in Knowledge Discovery and Data Mining*, 12084:197 – 209.

Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. 2020b. Orthogonal relation transforms with graph context modeling for knowledge graph embedding. In *ACL*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Pat Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *NAACL-HLT*.

J. Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. *ArXiv*, abs/1906.04284.

Haoyu Wang, M. Tan, Mo Yu, S. Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019a. Extracting multiple-relations in one-pass with pre-trained transformers. *ArXiv*, abs/1902.01030.

Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William W. J. Wang. 2019b. Fine-tune bert for docred with two-step process. *ArXiv*, abs/1909.11898.

L. Wang, Z. Cao, G. Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *ACL*.

Felix Wu, Tianyi Zhang, A. Souza, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019a. Simplifying graph convolutional networks. In *ICML*.

Y. Wu, Ruibang Luo, H. Leung, H. Ting, and T. Lam. 2019b. Renet: A deep learning approach for extracting gene-disease associations from literature. In *RECOMB*.

Yuan Yao, D. Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Z. Liu, Lixin Huang, Jie Zhou, and M. Sun. 2019. Docred: A large-scale document-level relation extraction dataset. In *ACL*.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential reasoning learning for language representation. *ArXiv*, abs/2004.06870.

Daojian Zeng, Kang Liu, Siwei Lai, G. Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *EMNLP*.

Zhengyan Zhang, Xu Han, Z. Liu, Xin Jiang, M. Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *ArXiv*, abs/1905.07129.

Heliang Zheng, J. Fu, Z. Zha, and Jiebo Luo. 2019. Learning deep bilinear transformation for fine-grained image representation. *ArXiv*, abs/1911.03621.