Span-based Joint Entity and Relation Extraction with Transformer Pre-training

Markus Eberts and **Adrian Ulges**[1]

Abstract. We introduce SpERT, an attention model for span-based joint entity and relation extraction. Our key contribution is a light-weight reasoning on BERT embeddings, which features entity recognition and filtering, as well as relation classification with a localized, marker-free context representation. The model is trained using strong within-sentence negative samples, which are efficiently extracted in a single BERT pass. These aspects facilitate a search over all spans in the sentence.

In ablation studies, we demonstrate the benefits of pre-training, strong negative sampling and localized context. Our model outperforms prior work by up to 2.6% F1 score on several datasets for joint entity and relation extraction.

1 Introduction

Transfomer networks such as BERT [8], GPT [26], Transformer-XL [7], RoBERTa [19] or MASS [30] have recently attracted strong attention in the NLP research community. These models use multi-head self-attention as a key mechanism to capture interactions between tokens [1, 32]. This way, context-sensitive embeddings can be obtained that disambiguate homonyms and express semantic and syntactic patterns. Transformer networks are commonly pre-trained on large document collections using language modelling objectives. The resulting models can then be transferred to target tasks with relatively small supervised training data, resulting in state-of-the-art performance in many NLP tasks such as question answering [37] or contextual emotion detection [5].

This work investigates the use of Transformer networks for relation extraction: Given a pre-defined set of target relations and a sentence such as "Leonardo DiCaprio starred in Christopher Nolan's thriller Inception", our goal is to extract triplets such as ("Leonardo DiCaprio", *Plays-In*, "Inception") or ("Inception", *Director*, "Christopher Nolan"). The task comprises of two subproblems, namely the identification of entities (entity recognition) and relations between them (relation classification). While common methods tackle the two problems separately [36, 39, 38], more recent work uses joint models for both steps [3, 21]. The latter approach seems promising, as on the one hand knowledge about entities (such as the fact that "Leonardo DiCaprio" is a person) is of interest when choosing a relation, while knowledge of the relation (*Director*) can be useful when identifying entities.

We present a model for joint entity and relation extraction that utilizes the Transformer network BERT as its core. A span-based approach is followed: Any token subsequence (or *span*) constitutes a potential entity, and a relation can hold between any pair of spans. Our model performs a full search over all these hypotheses. Unlike previous work based on BIO/BILOU labels [3, 18, 24], a span-based approach can identify *overlapping* entities such as "codeine" within "codeine intoxication". Since Transformer models like BERT are computationally expensive, our approach conducts only a single forward pass per input sentence and performs a light-weight reasoning on the resulting embeddings, In contrast to other recent approaches [21, 34], our model features a much simpler downstream processing using shallow entity/relation classifiers. We use a local context representation without using particular markers, and draw negative samples from the same sentence in a single BERT pass. These aspects facilitate an efficient training and a full search over all spans. We coin our model "Span-based Entity and Relation Transformer" (SpERT)[2]. In summary, our contributions are:

- We present a novel approach towards span-based joint entity and relation extraction. Our approach appears to be simple but effective, consistently outperforming prior work by up to 2.6% (relation extraction F1 score).
- We investigate several aspects crucial for the success of our model, showing that (1) negative samples from the same sentence yield a training that is both efficient and effective, and a sufficient number of strong negative samples appears to be vital. (2) A localized context representation is beneficial, especially for longer sentences. (3) We also study the effects of pre-training and show that fine-tuning a pre-trained model yields a strong performance increase over training from scratch.

2 Related Work

Traditionally, relation extraction is tackled by using separate models for entity detection and relation classification, whereas neural networks constitute the state of the art. Various approaches for relation classification have been investigated such as RNNs [39], recursive neural networks [29] or CNNs [38]. Also, Transformer models have been used for relation classification [33, 35]: The input text is fed once through a Transformer model and the resulting embeddings are classified. Note, however, that pre-labeled entities are assumed to be given. In contrast to this, our approach does not rely on labeled entities and jointly detects entities and relations.

Joint Entity and Relation Extraction Since entity detection and relation classification may benefit from exploiting interrelated signals, models for the joint detection of entities and relations have

[1] RheinMain University of Applied Sciences, Germany, {markus.eberts, adrian.ulges}@hs-rm.de

[2] The code for reproducing our results is available at https://github.com/markus-eberts/spert.

recently drawn attention (e.g., [3, 2, 21, 31, 40, 16]). Most approaches detect entities by sequence-to-sequence learning: Each token is tagged according to the well-known BIO scheme (or its BILOU variant).

Miwa and Sasaki [23] tackle joint entity and relation extraction as a table-filling problem, where each cell of the table corresponds to a word pair of the sentence. The diagonal of the table is filled with the BILOU tag of the token itself and the off-diagonal cells with the relations between the respective token pair. Relations are predicted by mapping the entities' last words. The table is filled with relation types by minimizing a scoring function based on several features such as POS tags and entity labels. A beam search is employed to find an optimal table-filling solution. Gupta et al. [10] also formulate joint entity and relation extraction as a table-filling problem. Unlike Miwa and Sasaki they employ a bidirectional recurrent neural network to label each word pair.

Miwa and Bansal [22] use a stacked model for joint entity and relation extraction. First, a bidirectional sequential LSTM tags the entities according to the BILOU scheme. Second, a bidirectional tree-structured RNN operates on the dependency parse tree between an entity pair to predict the relation type. Zhou et al. [42] utilize a BILOU-based combination of a bidirectional LSTM and a CNN to extract a high level feature representation of the input sentence. Since named entity extraction is only performed for the most likely relations, the approach predicts a lower number of labels compared to the table-filling approaches. Zheng et al. [41] first encode input tokens with a bidirectional LSTM. Another LSTM then operates on each encoded word representation and outputs the entity boundaries (akin to BILOU scheme) alongside their relation type. Conditions where one entity is related to multiple other entities are not considered. Bekoulis et al. [3, 2] also employ a bidirectional LSTM to encode each word of the sentence. They use character embeddings alongside Word2Vec embeddings as input representations. Entity boundaries and tags are extracted with a Conditional Random Field (CRF). In contrast to Zheng et al. [41], Bekoulis et al. also detect cases in which a single entity is related to multiple others.

While the above approaches heavily rely on LSTMs, our approach uses an attention-based Transformer type network. The attention mechanism has also been used in joint models: Nguyen and Verspoor [24] use a BiLSTM-CRF-based model for entity recognition. Token representations are shared with the relation classification task, and embeddings for BILOU entity labels are learned. In relation classification, entities interact via a bi-affine attention layer. Chi et al. [6] use similar BiLSTM representations. They detect entities with BIO tags and train with an auxiliary language modeling objective. Relation classifiers attend into the BiLSTM encodings. Note, however, that neither of the two works utilize Transformer type networks.

More similar to our work is the recent approach by Li et al. [18], who also apply BERT as their core model and use a question answering setting, where entity- and relation-specific questions guide the model to head and tail entities. The model requires manually defined (pseudo-)question templates per relation, such as "find a weapon which is owned by [?]".Entities are detected by a relation-wise labeling with BILOU-type tags, based on BERT embeddings. In contrast to this approach, our model requires no explicit formulation of questions. Also, our approach is span-based instead of BILOU.

Span-based Approaches As BIO/BILOU-based models only assign a single tag to each token, a token cannot be part of multiple entities at the same time, such that situations with overlapping (often *nested*) entities cannot be covered. Think of the sentence "Ford's Chicago plant employs 4,000 workers", where both "Chicago" and "Chicago plant" are entities. Here, *span-based* approaches – which perform an exhaustive search over all spans and offer the fundamental benefit of covering overlapping entities – have been investigated. Applications include coreference resolution [14, 15], semantic role labeling [25, 12], and the improvement of language modeling by learning to predict spans instead of single words [13].

Recently, some span-based models towards joint entity and relation extraction have been proposed [20, 9], using span representations derived from a BiLSTM over concatenated ELMo, word and character embeddings. These representations are then shared across the downstream tasks. While Dixit and Al-Onaizan [9] focus on joint entity and relation extraction, Luan et al. [20] conduct a beam search over the hypothesis space, estimating which spans participate in entity classes, relations and coreferences.

Luan et al.'s follow-up model DyGIE [21] adds a graph propagation step to capture the interaction of spans. A dynamic span graph is constructed, in which embeddings are propagated using a learned gated mechanism. Using this refinement of span representations, further improvements are demonstrated. More recently, Wadden et al.'s DyGIE++ [34] has replaced the BiLSTM encoder with BERT. DyGIE++ constitutes the only Transformer-based span approach towards joint entity and relation extraction yet. In contrast to DyGIE and DyGIE++, our model utilizes a much simpler downstream processing, omitting any graph propagation and using shallow entity and relation classifiers. Instead, we found localized context representation and strong negative sampling to be of vital importance. We include a quantitative comparison with DyGIE++ in the experimental section.

3 Approach

Our model uses a pre-trained BERT [8] model as its core, as illustrated in Figure 1: An input sentence is tokenized, obtaining a sequence of n byte-pair encoded (BPE) tokens [28]. Byte-pair encoding represents infrequent words (such as *treehouse*) by common subwords (*tree* and *house*) and is utilized in BERT to limit the vocabulary size and to map out-of-vocabulary words. The BPE tokens are passed through BERT, obtaining an embedding sequence $(\mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_n, \mathbf{c})$ of length $n+1$ (the last token $\mathbf{c}$ represents a special classifier token capturing the overall sentence context). Unlike classical relation classification, our approach detects entities among all token subsequences (or *spans*). For example, the token sequence (we,will,rock,you) maps to the spans (we), (we,will), (will,rock,you), etc. . We classify each span into entity types (a), filter non-entities (b), and finally classify all pairs of remaining entities into relations (c).

(a) Span Classification Our span classifier takes an arbitrary candidate span as input. Let $s := (\mathbf{e}_i, \mathbf{e}_{i+1}, ..., \mathbf{e}_{i+k})$ denote such a span. Also, we assume $\mathcal{E}$ to be a pre-defined set of entity categories such as *person* or *organization*. The span classifier maps the span s to a class out of $\mathcal{E} \cup \{none\}$. $none$ represents spans that do not constitute entities.

The span classifier is displayed in detail in the dashed box in Figure 1 (see Step (a)). Its input consists of three parts:

- The span's BERT embeddings (red) are combined using a fusion, $f(\mathbf{e}_i, \mathbf{e}_{i+1}, ..., \mathbf{e}_{i+k})$. Regarding the fusion function f, we found max-pooling to work best, but will investigate other options in the experiments.

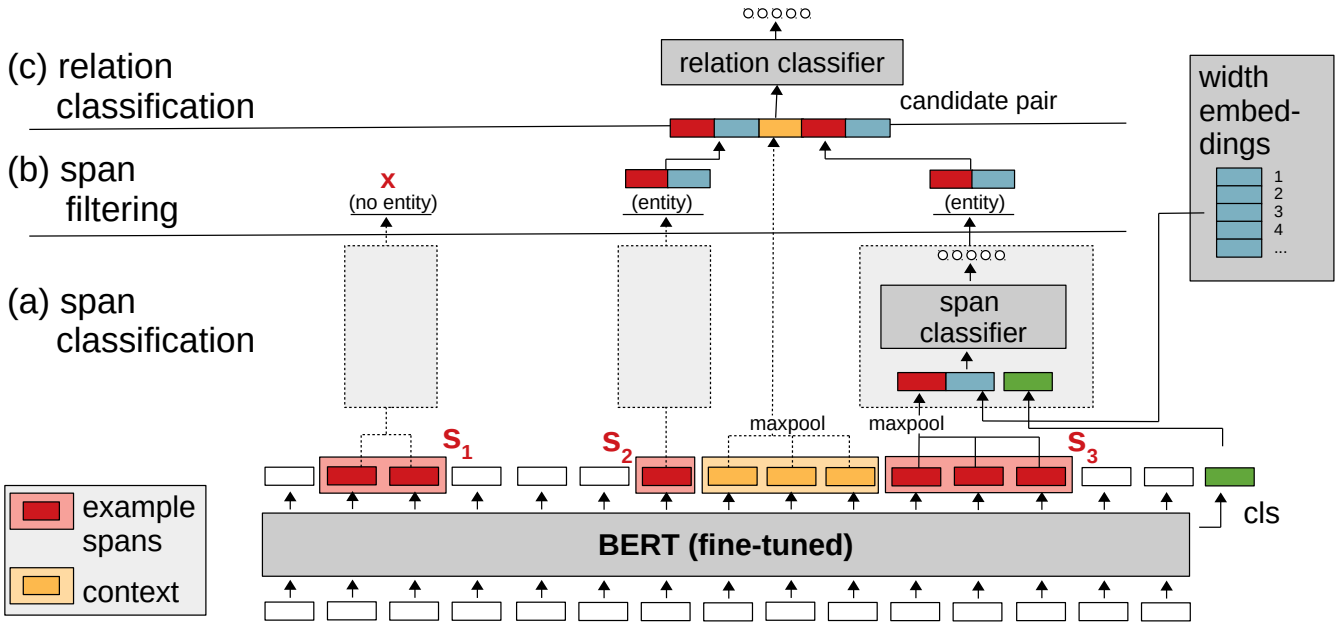

Figure 1. Our approach towards joint entity and relation extraction SpERT first passes a token sequence through BERT. Then, (a) all spans within the sentence are classified into entity types, as illustrated for three sample spans s_1, s_2, s_3 (red). (b) Spans classified as non-entites (here, s_1) are filtered. (c) All pairs of remaining entities (here, (s_2, s_3)) are combined with their context (the span between the entities, yellow) and classified into relations.

- Given the span width $k+1$, we look-up a width embedding $\mathbf{w}_{k+1}$ (blue) from a dedicated embedding matrix, which contains a fixed-size embedding for each span width $1, 2, ...$ [14]. These embeddings are learned by backpropagation, and allow the model to incorporate a prior over the span width (note that spans which are too long are unlikely to represent entities).

This yields the following span representation (whereas $\circ$ denotes concatenation):

$$\mathbf{e}(s) := f(\mathbf{e}_i, \mathbf{e}_{i+1}, ..., \mathbf{e}_{i+k}) \circ \mathbf{w}_{k+1}. \quad (1)$$

Finally, we add the classifier token $\mathbf{c}$ (Figure 1, green), which represents the overall sentence (or *context*). Context forms an important source of disambiguation, as keywords (such as *spouse* or *says*) are strong indicators for entity classes (such as *person*). The final input to the span classifier is:

$$\mathbf{x}^s := \mathbf{e}(s) \circ \mathbf{c} \quad (2)$$

This input is fed into a softmax classifier:

$$\hat{\mathbf{y}}^s = \text{softmax}\left(W^s \cdot \mathbf{x}^s + \mathbf{b}^s\right) \quad (3)$$

which yields a posterior for each entity class (incl. *none*).

(b) Span Filtering By looking at the highest-scored class, the span classifier's output (Equation 3) estimates which class each span belongs to. We use a simple approach and filter all spans assigned to the *none* class, leaving a set of spans $\mathcal{S}$ which supposedly constitute entities. Note that – unlike prior work [23, 20] – we do not perform a beam search over the entity/relation hypotheses. We pre-filter spans longer than 10 tokens, limiting the cost of span classification to $O(n)$.

(c) Relation Classification Let $\mathcal{R}$ be a set of pre-defined relation classes. The relation classifier processes each candidate pair (s_1, s_2) of entities drawn from $\mathcal{S} \times \mathcal{S}$ and estimates if any relation from $\mathcal{R}$ holds. The input to the classifier consists of two parts:

1. To represent the two entity candidates s_1, s_2, we use the fused BERT/width embeddings $\mathbf{e}(s_1), \mathbf{e}(s_2)$ (Eq. 1).
2. Obviously, words from the context such as *spouse* or *president* are important indicators of the expressed relation. One possible context representation would be the classifier token $\mathbf{c}$. However, we found $\mathbf{c}$ to be unsuitable for long sentences expressing a multitude of relations. Instead, we use a more localized context drawn from the direct surrounding of the entities: Given the span ranging from the end of the first entity to the beginning of the second entity (Figure 1, yellow), we combine its BERT embeddings by max-pooling, obtaining a context representation $\mathbf{c}(s_1, s_2)$. If the range is empty (e.g., in case of overlapping entities), we set $\mathbf{c}(s_1, s_2) = \mathbf{0}$.

Just like for the span classifier, the input to the relation classifier is obtained by concatenating the above features. Note that – since relations are asymmetric in general – we need to classify both (s_1, s_2) and (s_2, s_1), i.e. the input becomes

$$\mathbf{x}_1^r := \mathbf{e}(s_1) \circ \mathbf{c}(s_1, s_2) \circ \mathbf{e}(s_2)$$
$$\mathbf{x}_2^r := \mathbf{e}(s_2) \circ \mathbf{c}(s_1, s_2) \circ \mathbf{e}(s_1).$$

Both $\mathbf{x}_1^r$ and $\mathbf{x}_2^r$ are passed through a single-layer classifier:

$$\hat{\mathbf{y}}_{1/2}^r := \sigma\left(W^r \cdot \mathbf{x}_{1/2}^r + \mathbf{b}^r\right) \quad (4)$$

where σ denotes a sigmoid of size $\#\mathcal{R}$. Any high response in the sigmoid layer indicates that the corresponding relation holds between s_1 and s_2. Given a confidence threshold α, any relation with a score $\geq \alpha$ is considered activated. If none is activated, the sentence is assumed to express no known relation between the two entities.

3.1 Training

We learn the size embeddings $\mathbf{w}$ (Figure 1, blue) as well as the span/relation classifiers' parameters ($W^s, \mathbf{b}^s, W^r, \mathbf{b}^r$) and fine-tune BERT in the process. Our training is supervised: Given sentences with annotated entities (including their entity types) and relations, we define a joint loss function for entity classification and relation classification:

$$\mathcal{L} = \mathcal{L}^s + \mathcal{L}^r,$$

whereas $\mathcal{L}^s$ denotes the span classifier's loss (cross-entropy over the entity classes including *none*) and $\mathcal{L}^r$ denotes the binary cross-entropy over relation classes. Both losses are averaged over each batches' samples. No class weights are applied. A training batch consists of B sentences, from which we draw samples for both classifiers:

- For the span classifier, we utilize all labeled entities $\mathcal{S}^{gt}$ as positive samples, plus a fixed number N_e of random non-entity spans as negative samples. For example, given the sentence "In 1913, Olympic legend [Jesse Owens]$_{\text{People}}$ was born in [Oakville, Alabama]$_{\text{Location}}$." we draw negative samples such as "Owens" or "born in".
- To train the relation classifier, we use ground truth relations as positive samples, and draw N_r negative samples from those entity pairs $\mathcal{S}^{gt} \times \mathcal{S}^{gt}$ that are not labeled with any relation. For example, given a sentence with the two relations ("Marge", *Mother*, "Bart") and ("Bart", *Teacher*, "Skinner"), the unconnected entity pair ("Marge", *, "Skinner") constitutes a negative sample for any relation. We found such *strong* negative samples – in contrast to sampling random span pairs – to be of vital importance.

Note that the above process samples training examples *per sentence*: Instead of generating samples scattered over multiple sentences – which would require us to feed all those sentences through the deep and computationally expensive BERT model – we run each sentence only once through BERT (*single-pass*). This way, multiple positive/negative samples pass a single shallow linear layer for the entity and relation classifier respectively, which speeds-up the training process substantially.

4 Experiments

We compare SpERT with other joint entity/relation extraction models and investigate the influence of several hyperparameters. The evaluation is conducted on three publicly available datasets:

- **CoNLL04**: The CoNLL04 dataset [27] contains sentences with annotated named entities and relations extracted from news articles. It includes four entity (*Location*, *Organization*, *People*, *Other*) and five relation types (*Work-For*, *Kill*, *Organization-Based-In*, *Live-In*, *Located-In*). We employ the training (1,153 sentences) and test set (288 sentences) split by Gupta et al. [10]. For hyperparameter tuning, 20% of the training set is used as a held-out development part.
- **SciERC**: SciERC [20] is derived from 500 abstracts of AI papers. The dataset includes six scientific entity (*Task*, *Method*, *Metric*, *Material*, *Other-Scientific-Term*, *Generic*) and seven relation types (*Compare*, *Conjunction*, *Evaluate-For*, *Used-For*, *Feature-Of*, *Part-Of*, *Hyponym-Of*) in a total of $2,687$ sentences. We use the same train ($1,861$ sentences), validation (275 sentences) and test (551) split as in [20].
- **ADE**: The ADE dataset [11] consists of $4,272$ sentences and $6,821$ relations extracted from medical reports that describe the adverse effects arising from drug use. It contains a single relation type *Adverse-Effect* and the two entity types *Adverse-Effect* and *Drug*. As in previous work, we conduct a 10-fold cross validation.

We evaluate SpERT on both entity recognition and relation extraction. An entity is considered correct if its predicted span and entity label match the ground truth. A relation is considered correct if its relation type as well as the two related entities are both correct (in span and type). Only for SciERC, entity type correctness is not considered when evaluating relation extraction [20]. Following previous work, we measure the precision, recall and F1 score for each entity and relation type, and report the macro-averaged values for the ADE dataset and the micro-averaged ones for SciERC. For ADE, the F1 score is averaged over the folds. On CoNLL04, F1 scores were reported both as micro and macro averages in prior work, which is why we report both metrics.

For most of our experiments we use the BERT$_{\text{BASE}}$ (cased) model[3] as a sentence encoder, pre-trained on English language [8]. On the SciERC dataset, we follow [34] and replace BERT with SciBERT (cased) [4], a BERT model pre-trained on a large corpus of scientific papers. We initialize our classifiers' weights with normally distributed random numbers (μ=0, σ=0.02). We use the Adam Optimizer with a linear warmup and linear decay learning rate schedule and a peak learning rate of $5e{-}5$, a dropout before the entity and relation classifier with a rate of 0.1 (both according to [8]), a batch size of B=2, and width embeddings $\mathbf{w}$ of 25 dimensions. No further optimizations were conducted on those parameters. We choose the number of epochs (20), the relation filtering threshold ($\alpha = 0.4$), as well as the number of negative entity and relation samples per sentence (N_e=N_r=100) based on the CoNLL04 development set. We do not specifically tune our model for the other two datasets but use the same hyperparameters instead.

4.1 Comparison with State of the Art

Table 1 shows the test set evaluation results for the three datasets. We report the average over 5 runs for each dataset. SpERT consistently outperforms the state-of-the-art for both entity and relation extraction on all datasets. While entity recognition performance increased by 1.1% (CoNLL04), 2.8% (SciERC) and 2.1% (ADE) F1 respectively, we observe even stronger performance increases in relation extraction: Compared to Li et al. [18] ("Multi-turn QA" in Table 1), who also rely on BERT as a sentence encoder but use a BILOU approach for entity extraction, our model improves the state-of-the-art on the CoNLL04 dataset by 2.6% (micro) F1. On the challenging and domain-specific SciERC dataset, SpERT outperforms the DyGIE++ model of Wadden et al. [34] by about 2.4% using SciBERT as a sentence encoder. When BERT is used instead, the performance drops by 4.4%, confirming that in-domain language model pre-training is beneficial, which is in line with findings of Wadden et al. [34].

On the ADE dataset, SpERT achieves an improvement of about 2% (SpERT (without overlap) in Table 1) F1 compared to the "Relation-Metric" model by Tran and Kavuluru [31]. Note that ADE also contains 120 instances of relations with overlapping entities, which can be discovered by span-based approaches like SpERT (in contrast to BILOU-based models). These have been filtered in prior

[3] using 12 layers, 768-dimensional embeddings, 12 heads per layer, resulting in a total 110M parameters.

Dataset	Model	Entity			Relation		
		Precision	Recall	F1	Precision	Recall	F1
CoNLL04	Global Optimization [40]†	-	-	85.60	-	-	67.80
	Multi-turn QA [18]†	89.00	86.60	87.80	69.20	68.20	68.90
	Multi-head + AT [2]‡	-	-	83.61	-	-	61.95
	Multi-head [3]‡	83.75	84.06	83.90	63.75	60.43	62.04
	Relation-Metric [31]‡	84.46	84.67	84.57	67.97	58.18	62.68
	Biaffine Attention [24]‡	-	-	86.20	-	-	64.40
	Table-filling [23]*	81.20	80.20	80.70	76.00	50.90	61.00
	Hierarchical Attention [6]*	-	-	86.51	-	-	62.32
	SpERT†	88.25	89.64	**88.94**	73.04	70.00	**71.47**
	SpERT‡	85.78	86.84	**86.25**	74.75	71.52	**72.87**
SciERC	SciIE [20]†	67.20	61.50	64.20	47.60	33.50	39.30
	DyGIE [21]†	-	-	65.20	-	-	41.60
	DyGIE++ [34]†	-	-	67.50	-	-	48.40
	SpERT† (using BERT)	68.53	66.73	67.62	49.79	43.53	46.44
	SpERT† (using SciBERT)	70.87	69.79	**70.33**	53.40	48.54	**50.84**
ADE	CNN + Global features [17]‡	79.50	79.60	79.50	64.00	62.90	63.40
	BiLSTM + SDP [16]‡	82.70	86.70	84.60	67.50	75.80	71.40
	Multi-head [3]‡	84.72	88.16	86.40	72.10	77.24	74.58
	Multi-head + AT [2]‡	-	-	86.73	-	-	75.52
	Relation-Metric [31]‡	86.16	88.08	87.11	77.36	77.25	77.29
	SpERT‡ (without overlap)	89.26	89.26	**89.25**	78.09	80.43	**79.24**
	SpERT‡ (with overlap)	88.99	89.59	**89.28**	77.77	79.96	**78.84**

Table 1. Test set results CoNLL04, SciERC and ADE. Our model SpERT outperforms the state-of-the-art in both entity and relation extraction by up to 2.6% (CoNLL04). *(metrics: micro-average=†, macro-average=‡, not stated=∗)*

work [3, 16, 31]. As a reference for future work on overlapping entity recognition, we also present results on the full dataset (including the overlapping entities). When including this additional challenge, our model performs only marginally worse (-0.4%) compared to not considering overlapping entities. Out of the 120 relations with overlapping entities, 65 were detected correctly ($\approx 54\%$). Examples of relations between overlapping entities correctly predicted by SpERT are included in Table 4 (top).

4.2 Candidate Selection and Negative Sampling

We also study the effect of the number and sampling of negative training examples. Figure 2 shows the F1 score (relations and entities) for the CoNLL04 and SciERC development sets, plotted against the number of negative samples N_e/N_r per sentence. We see that a sufficient number of negative samples is essential: When using only a single negative entity and relation ($N_e{=}N_r{=}1$) per sentence, relation F1 is about 10.5% (CoNLL04) and 9.7% (SciERC). With a high number of negative samples, the performance stagnates for both datasets. However, we found our results to be more stable when using a sufficiently high N_e and N_r (we chose $N_e{=}N_r{=}100$ in all other experiments).

For relation classification, we also assess the effect of using weak instead of strong negative relation samples: Instead of using the entity classifier as a filter for entity candidates $\mathcal{S}$ and drawing *strong* negative training samples from $\mathcal{S}{\times}\mathcal{S}$, we omit span filtering and sample random training span pairs not matching any ground truth relation. With these *weak* samples, our model retains a high recall (84.4%) on the CoNLL04 development set, but the precision decreases drastically to about 4.3%. We observed that the model tends to predict *subspans* of entities to be in relation when using weak samples: For example, in the sentence "[John Wilkes Booth]$_{\text{head}}$, who assassinated [President Lincoln]$_{\text{tail}}$, was an actor", the pairs ("John", "President") or ("Wilkes", "Lincoln") are chosen. Additionally, pairs where one entity is correct and the other one incorrect are also favored by the model. Obviously, span filtering is not only beneficial in terms of training and evaluation speed, but is also vital for accurate localization in SpERT.

4.3 Localized Context

Despite advances in detecting long distance relations using LSTMs or the attention mechanism, the noise induced with increasing context remains a challenge. By using a *localized* context, i.e. the context between entity candidates, the relation classifier can focus on the sentence's section that is often most discriminative for the relation type. To assess this effect, we compare localized context with two other context representations that use the whole sentence:

- **Full context**: Instead of performing a max pooling over the context between entity candidates, a max pooling over all tokens in the sentence is conducted.
- **Cls token**: Just like in the *entity* classifier (Figure 1, green), we use a special classifier token as context, which is able to attend to the whole sentence.

We evaluate the three options on the CoNLL04 development set (Figure 3): When employing SpERT with a localized context, the model reaches an F1 score of 71.0%, which significantly outperforms a max pooling over the whole sentence (65.8%) and using the classifier token (63.9%).

Figure 3 also displays results with respect to the sentence length: We split the CoNLL04 development set into four different parts,

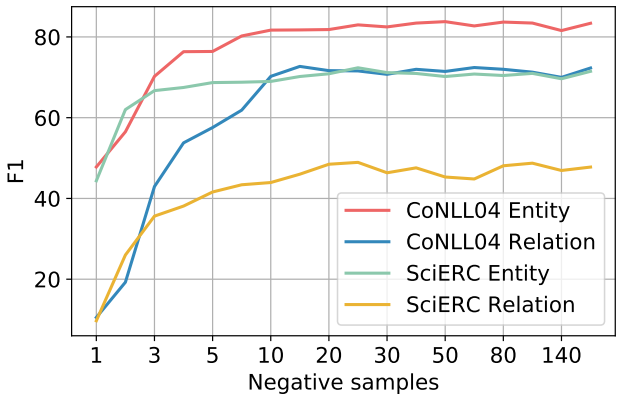

Figure 2. The accuracy of entity and relation classification (F1 on CoNLL04 and SciERC development set) increases significantly with the number of negative samples.

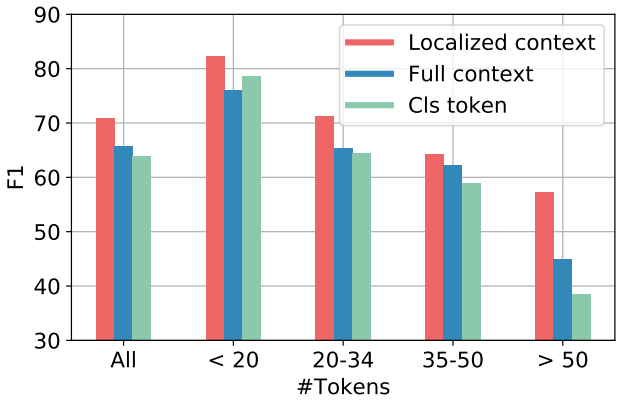

Figure 3. Macro F1 scores of relation classification on the CoNLL04 development set when using different context representations. Localized context (red) performs best overall (left), particularly on long sentences with >50 tokens (right).

namely sentences with <20, $20-34$, $35-50$ and >50 tokens. Obviously, localized context leads to comparable or better results for all sentence lengths, particularly for very long sentences: Here, it reaches an F1 score of 57.3%, while the performance drastically decreases to $44.9/38.5\%$ when using the other options. Table 4 (middle) shows an example of a long sentence with multiple entities: By using a localized context the model correctly predicts the three *Located-In* relations, while relying on the full context leads to many false positive relations such as ("Jackson", *Located-In*, "Colo.") or ("Wyo.", *Located-In*, "McAllen"). This shows that guiding the model towards relevant sections of the input sentence is vital. An interesting direction for future work is to learn the relevant context with respect to the entity candidates, and to incorporate precomputed syntactical information into SpERT.

4.4 Pre-training and Entity Representation

Next, we assess the effect of BERT's language modeling pre-training. It seems intuitive that pre-training on large-scale datasets helps the model to learn semantic and syntactic relations that are hard to capture on a limited-scale target dataset. Therefore, we test three variants of pre-training:

1. **Full**: We use the fully pre-trained BERT model (*LM Pre-trained*, our default setting).
2. **–Layers**: We retain pre-trained token embeddings but train the layers from scratch (using the default initalization [8]).
3. **–Layers,Embeddings**: We train layers and token embeddings from scratch (again, using the default initialization).

As Table 2 shows, training the BERT layers from scratch results in a performance drop of about 17.0% and 29.4% (macro) F1 for entity and relation extraction respectively. Further, training the token embeddings from scratch results in an even stronger drop in F1. These results suggests that pre-training a large network like BERT is challenging on the fairly small joint entity and relation extraction datasets. Therefore, language modeling pre-training is vital for generalization and to obtain a competitive performance.

Finally, we investigate different options for the entity span representation $\mathbf{e}(s)$ other than conducting a max pooling over the entity's tokens, namely a sum and average pooling (note that a size embedding and a context representation is again concatenated to obtain the final entity representation (Equation 1)). Table 3 shows the CoNLL04 (macro) F1 with respect to the different entity representations: We found the averaging of the entity tokens to be unsuitable for both entity (69.2%) and relation extraction (44.8%). Sum pooling improves the performance to $80.3/68.2\%$. Max pooling, however, outperforms this by another increase of 3.8% and 2.8% respectively.

Pre-training	Entity F1	Relation F1
Full	84.04	70.98
– Layers	67.06	41.58
– Layers,Embeddings	50.84	25.22

Table 2. Effect of BERT pre-training on entity and relation extraction (CoNLL04 development set). A fully pre-trained BERT model significantly outperforms two BERTs in which the self-attention layers (–Layers) or the layers and the BPE input token embeddings (–Layers,Embeddings) are trained from scratch.

Pooling	Entity F1	Relation F1
Max	84.04	70.98
Sum	80.26	68.16
Average	69.21	44.75

Table 3. Investigation of different entity span representations $\mathbf{e}(s)$ (summing and averaging of entity's tokens).

4.5 Error Inspection

Although SpERT yields strong results on joint entity and relation extraction, we observed several common errors which leave room for further research. Table 4 (bottom) contains examples of five error cases we found to be common in the evaluated datasets:

- **Incorrect spans**: One common source of error is the prediction of slightly incorrect entity spans, e.g. by adding a nearby word or missing a word annotated in the ground truth. This error occurs especially often in the domain specific ADE and SciERC datasets.
- **Syntax**: Another frequently encountered error is the prediction of a relation (here: *Work-For*) between two entities, which could possibly be related based on their entity types ("Yevhen Saburov", a person, and "Black Sea Fleet", an employer), but are not related in the sentence.

(a) Examples of Overlapping Entites	
	Six days after starting acyclovir she exhibited signs of [[lithium] toxicity].
	A diagnosis of masked [[theophylline] poisoning] should be considered in similar situations involving a rapid decrease of insulin requirements.
(b) Effect of Localized Context	
localized context	Temperatures around the nation at 2 a.m. EST ranged from 2 degrees at [Jackson]$_1$, [Wyo.]$_1$, and [Gunnison]$_2$, [Colo.]$_2$, to 89 degrees at [McAllen]$_3$, [Texas]$_3$.
full context	Temperatures around the nation at 2 a.m. EST ranged from 2 degrees at [[Jackson]$_1$]$_2$, [Wyo.]$_3$, and [Gunnison]$_4$, [[Colo.]$_4$]$_1$, to 89 degrees at [[McAllen]$_5$]$_3$, [[Texas]$_5$]$_2$.
(c) Error Cases	
incorrect spans	[Delayed [bowel injury]] is an infrequently observed complication of [[chromic phosphate]] administration.
syntax	Ambassador Miller is also scheduled to meet with Crimean Deputy [Yevhen Saburov] and [[Black Sea Fleet]] Commander [Eduard Baltin].
logical	[Becton Dickinson] sells needle containers to doctors and hospitals but may develop a container for home use, said [Linda Schmitt], an assistant product manager.
classification	Finally, we briefly describe an experiment which we have done in extending the [[n-best speech / language integration architecture]$_{\text{rel:Used-For}}$]$_{\text{rel:Evaluate-For}}$ to improving [[OCR accuracy]$_{\text{rel:Used-For}}$]$_{\text{rel:Evaluate-For}}$.
missing annotation	[[Norton Winfred Simon]] was born on Feb. 5, 1907, in [Portland, Ore.], and spent his teenage years in [San Francisco] .

Table 4. SpERT relation extraction examples showing that (a) as a span-based approach, our model can deal with overlapping entities, and (b) localized context yields better precision for long sentences compared to using the full sentence as context. (c) showcases various common sources of error. green [*] = true positive relation, blue [*] = false positive relation, red [*] = false negative relation.

- **Logical**: Sometimes, a relation is not explicitly stated in the sentence, but can logically be inferred based on the context. In the depicted case, it is not stated that "Linda Schmitt" is indeed a product manager of "Becton Dickinson", but it is obvious due to her speaking for the company.
- **Classification**: In some rare cases (especially in the SciERC dataset), SpERT correctly predicted two related entities, but assigned a wrong relation type.
- **Missing annotation**: Finally, there are also some cases where a correct prediction is missing in the ground truth. Here, in addition to correctly predicting ("Norton Winfried Simon", *Live-In*, "Portland, Ore."), SpERT also outputs ("Norton Winfried Simon", *Live-In*, "San Francisco"), which is correct but not labeled.

5 Conclusions

We have presented SpERT, a span-based model for joint entity and relation extraction that relies on the pre-trained Transformer network BERT as its core. We show that with strong negative sampling, span filtering, and a localized context representation, a search over all spans in an input sentence becomes feasible. Our results suggest that span-based approaches perform competitive to BILOU-based models and may be the more promising approach for future research due to their ability to identify overlapping entities.

In the future, we plan to investigate more elaborate forms of context for relation classifiers. Currently, our model simply employs the span between the two entities, which proved superior to the full context. Employing additional syntactic features or learned context – while maintaining an efficient exhaustive search – appears to be a promising challenge.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, 'Neural Machine Translation by Jointly Learning to Align and Translate', *CoRR*, **abs/1409.0473**, (2014).

[2] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder, 'Adversarial training for multi-context joint entity and relation extraction', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2830–2836, Brussels, Belgium, (October-November 2018). Association for Computational Linguistics.

[3] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder, 'Joint entity recognition and relation extraction as a multi-head selection problem', *Expert Systems with Applications*, **114**, 34–45, (04 2018).

[4] Iz Beltagy, Kyle Lo, and Arman Cohan, 'SciBERT: A Pretrained Language Model for Scientific Text', *ArXiv*, **abs/1903.10676**, (2019).

[5] Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal, 'SemEval-2019 Task 3: EmoContext Contextual Emotion Detection in Text', in *Proc. of the 13th International Workshop on Semantic Evaluation*, pp. 39–48, Minneapolis, Minnesota, USA, (June 2019). ACL.

[6] Renjun Chi, Bin Wu, Linmei Hu, and Yunlei Zhang, 'Enhancing Joint Entity and Relation Extraction with Language Modeling and Hierarchical Attention', in *Proc. APWeb-WAIM, LNCS 11641*, pp. 314–328, (7 2019).

[7] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov, 'Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context', *CoRR*, **abs/1901.02860**, (2019).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', in *Proc. of NAACL-HLT 2019*, pp. 4171–4186, Minneapolis, Minnesota, (June 2019). ACL.

[9] Kalpit Dixit and Yaser Al-Onaizan, 'Span-level model for relation extraction', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5308–5314, Florence, Italy, (July 2019). Association for Computational Linguistics.

[10] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy, 'Table Filling

Multi-Task Recurrent Neural Network for Joint Entity and Relation Extraction', in *Proc. of COLING 2016*, pp. 2537–2547, Osaka, Japan, (December 2016). The COLING 2016 Organizing Committee.

[11] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo, 'Development of a Benchmark Corpus to Support the Automatic Extraction of Drug-related Adverse Effects from Medical Case Reports', *J. of Biomedical Informatics*, **45**(5), 885–892, (October 2012).

[12] Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer, 'Jointly predicting predicates and arguments in neural semantic role labeling', in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 364–369, Melbourne, Australia, (July 2018). Association for Computational Linguistics.

[13] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy, 'Spanbert: Improving pre-training by representing and predicting spans', *CoRR*, **abs/1907.10529**, (2019).

[14] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer, 'End-to-end Neural Coreference Resolution', in *Proc. of EMNLP 2017*, pp. 188–197, Copenhagen, Denmark, (September 2017). ACL.

[15] Kenton Lee, Luheng He, and Luke Zettlemoyer, 'Higher-Order Coreference Resolution with Coarse-to-Fine Inference', in *Proc. of NAACL-HLT 2018*, volume 2, pp. 687–692, New Orleans, Louisiana, (June 2018). ACL.

[16] Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji, 'A neural joint model for entity and relation extraction from biomedical text', *BMC Bioinformatics*, **18**(1), 198, (2017).

[17] Fei Li, Yue Zhang, Meishan Zhang, and Donghong Ji, 'Joint Models for Extracting Adverse Drug Events from Biomedical Text', in *Proc. of IJCAI 2016*, IJCAI'16, pp. 2838–2844. AAAI Press, (2016).

[18] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li, 'Entity-Relation Extraction as Multi-Turn Question Answering', in *Proc. of ACL 2019*, pp. 1340–1350, Florence, Italy, (July 2019). ACL.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, 'RoBERTa: A Robustly Optimized BERT Pretraining Approach', *CoRR*, **abs/1907.11692**, (2019).

[20] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi, 'Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction', in *Proc. of EMNLP 2018*, pp. 3219–3232, Brussels, Belgium, (October-November 2018). ACL.

[21] Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi, 'A general framework for information extraction using dynamic span graphs', in *Proc. of NAACL-HLT 2019*, volume 1, pp. 3036–3046, Minneapolis, Minnesota, (June 2019). ACL.

[22] Makoto Miwa and Mohit Bansal, 'End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures', in *Proc. of ACL 2016*, pp. 1105–1116, Berlin, Germany, (August 2016). ACL.

[23] Makoto Miwa and Yutaka Sasaki, 'Modeling Joint Entity and Relation Extraction with Table Representation', in *Proc. of EMNLP 2014*, pp. 1858–1869, Doha, Qatar, (2014). ACL.

[24] Dat Quoc Nguyen and Karin Verspoor, 'End-to-end neural relation extraction using deep biaffine attention', in *Proc. of ECIR 2019*, (2019).

[25] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto, 'A span selection model for semantic role labeling', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1630–1642, Brussels, Belgium, (October-November 2018). Association for Computational Linguistics.

[26] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever, 'Improving Language Understanding by Generative Pre-Training', (2018).

[27] Dan Roth and Wen-tau Yih, 'A Linear Programming Formulation for Global Inference in Natural Language Tasks', in *Proc. of CoNLL 2004 at HLT-NAACL 2004*, pp. 1–8, Boston, Massachusetts, USA, (May 6 - May 7 2004). ACL.

[28] Rico Sennrich, Barry Haddow, and Alexandra Birch, 'Neural machine translation of rare words with subword units', in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, (August 2016). Association for Computational Linguistics.

[29] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng, 'Semantic Compositionality Through Recursive Matrix-vector Spaces', in *Proc. of EMNLP-CoNLL 2012*, pp. 1201–1211, Stroudsburg, PA, USA, (2012). ACL.

[30] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu, 'MASS: Masked Sequence to Sequence Pre-training for Language Generation', *CoRR*, **abs/1905.02450**, (2019).

[31] Tung Tran and Ramakanth Kavuluru, 'Neural metric learning for fast end-to-end relation extraction', *CoRR*, **abs/1905.07458**, (2019).

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin, 'Attention is All you Need', in *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., (2017).

[33] Patrick Verga, Emma Strubell, and Andrew Mccallum, 'Simultaneously Self-Attending to All Mentions for Full-Abstract Biological Relation Extraction', in *Proc. of ACL-HLT 2018*, pp. 872–884, (01 2018).

[34] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi, 'Entity, Relation, and Event Extraction with Contextualized Span Representations', *ArXiv*, **abs/1909.03546**, (2019).

[35] Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar, 'Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers', in *Proc. of ACL 2019*, pp. 1371–1377, Florence, Italy, (July 2019). ACL.

[36] Vikas Yadav and Steven Bethard, 'A Survey on Recent Advances in Named Entity Recognition from Deep Learning models', in *Proc. of the 27th International Conference on Computational Linguistics*, pp. 2145–2158, Santa Fe, New Mexico, USA, (August 2018). ACL.

[37] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin, 'End-to-End Open-Domain Question Answering with BERTserini', in *Proc. of NAACL 2019 (Demonstrations)*, pp. 72–77, Minneapolis, Minnesota, (June 2019). ACL.

[38] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao, 'Relation Classification via Convolutional Deep Neural Network', in *Proc. of COLING 2014*, pp. 2335–2344, Dublin, Ireland, (August 2014). Dublin City University and ACL.

[39] Dongxu Zhang and Dong Wang, 'Relation Classification via Recurrent Neural Network', *CoRR*, **abs/1508.01006**, (2015).

[40] Meishan Zhang, Yue Zhang, and Guohong Fu, 'End-to-End Neural Relation Extraction with Global Optimization', in *Proc. of EMNLP 2017*, pp. 1730–1740, Copenhagen, Denmark, (September 2017). ACL.

[41] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu, 'Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme', in *Proc. of ACL 2017*, pp. 1227–1236, Vancouver, Canada, (July 2017). ACL.

[42] Peng Zhou, Suncong Zheng, Jiaming Xu, Zhenyu Qi, Hongyun Bao, and Bo Xu, 'Joint Extraction of Multiple Relations and Entities by Using a Hybrid Neural Network', in *Proc. of CCL 2017*, pp. 135–146, (10 2017).