END-TO-END NAMED ENTITY RECOGNITION AND RELATION EXTRACTION USING PRE-TRAINED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Named entity recognition (NER) and relation extraction (RE) are two important tasks in information extraction and retrieval (IE & IR). Recent work has demonstrated that it is beneficial to learn these tasks jointly, which avoids the propagation of error inherent in pipeline-based systems and improves performance. However, state-of-the-art joint models typically rely on external natural language processing (NLP) tools, such as dependency parsers, limiting their usefulness to domains (e.g. news) where those tools perform well. The few neural, end-to-end models that have been proposed are trained almost completely from scratch. In this paper, we propose a neural, end-to-end model for jointly extracting entities and their relations which does not rely on external NLP tools and which integrates a large, pre-trained language model. Because the bulk of our model's parameters are pre-trained and we eschew recurrence for self-attention, our model is fast to train. On 5 datasets across 3 domains, our model matches or exceeds state-of-the-art performance, sometimes by a large margin.

1 INTRODUCTION

The extraction of named entities (named entity recognition, NER) and their semantic relations (relation extraction, RE) are key tasks in information extraction and retrieval (IE & IR). Given a sequence of text (usually a sentence), the objective is to identify both the named entities and the relations between them. This information is useful in a variety of NLP tasks such as document summarization, question answering, knowledge base population, and semantic search (Jiang, 2012). In the biomedical domain, NER and RE facilitate large-scale biomedical data analysis, such as network biology (Zhou et al., 2014), gene prioritization (Aerts et al., 2006), drug repositioning (Wang & Zhang, 2013) and the creation of curated databases (Li et al., 2015). In the clinical domain, NER and RE can aid in disease and treatment prediction, readmission prediction, de-identification, and patient cohort identification (Miotto et al., 2017).

Most commonly, the tasks of NER and RE are approached as a pipeline, with NER preceding RE. There are two main drawbacks to this approach: (1) Pipeline systems are prone to error propagation between the NER and RE systems. (2) One task is not able to exploit useful information from the other (e.g. the *type* of relation identified by the RE system may be useful to the NER system for determining the *type* of entities involved in the relation, and vice versa). More recently, joint models which simultaneously learn to extract entities and relations have been proposed, alleviating the aforementioned issues and achieving state-of-the-art performance (Miwa & Sasaki, 2014; Miwa & Bansal, 2016; Li et al., 2016; 2017; Zhang et al., 2017; Bekoulis et al., 2018a;b; Nguyen & Verspoor, 2019; Li et al., 2019).

Many of the proposed joint models for entity and relation extraction rely heavily on external NLP tools such as dependency parsers. For instance, Miwa & Bansal (2016) propose a recurrent neural network (RNN)-based joint model that uses a bidirectional long-short term memory network (BiLSTM) to model the entities and a tree-LSTM to model the relations between entities; Li et al. (2017) propose a similar model for biomedical text. The tree-LSTM uses dependency tree information extracted using an external dependency parser to model relations between entities. The use of these external NLP tools limits the effectiveness of a model to domains (e.g. news) where those

NLP tools perform well. As a remedy to this problem, Bekoulis et al. (2018a) propose a neural, end-to-end system which jointly learns to extract entities and relations without relying on external NLP tools. In Bekoulis et al. (2018b), they augment this model with adversarial training. Nguyen & Verspoor (2019) propose a different, albeit similar end-to-end neural model which makes use of deep biaffine attention (Dozat & Manning, 2016). Li et al. (2019) approach the problem with multi-turn question answering, posing templated queries to a BERT-based QA model (Devlin et al., 2018) whose answers constitute extracted entities and their relations and achieve state-of-the-art results on three popular benchmark datasets.

While demonstrating strong performance, end-to-end systems like Bekoulis et al. (2018a;b) and Nguyen & Verspoor (2019) suffer from two main drawbacks. The first is that most of the models parameters are trained from scratch. For large datasets, this can lead to long training times. For small datasets, which are common in the biomedical and clinical domains where it is particularly challenging to acquire labelled data, this can lead to poor performance and/or overfitting. The second is that these systems typically contain RNNs, which are sequential in nature and cannot be parallelized within training examples. The multi-pass QA model proposed in Li et al. (2019) alleviates these issues by incorporating a pre-trained language model, BERT (Devlin et al., 2018), which eschews recurrence for self-attention. The main limitation of their approach is that it relies on hand-crafted question templates to achieve maximum performance. This may become a limiting factor where domain expertise is required to craft such questions (e.g., for biomedical or clinical corpora). Additionally, one has to create a question template for each entity and relation type of interest.

In this study, we propose an end-to-end model for joint NER and RE which addresses all of these issues. Similar to past work, our model can be viewed as a mixture of a NER module and a RE module (Figure 1). Unlike most previous works, we include a pre-trained, transformer-based language model, specifically BERT (Devlin et al., 2018), which achieved state-of-the-art performance across many NLP tasks. The weights of the BERT model are fine-tuned during training, and the entire model is trained in an end-to-end fashion.

Our main contributions are as follows: (1) Our solution is truly end-to-end, relying on no hand-crafted features (e.g. templated questions) or external NLP tools (e.g. dependency parsers). (2) Our model is fast to train (e.g. 12 minutes on a single GPU for the CoNLL04 corpus), as most of its parameters are pre-trained and we avoid recurrence. (3) We match or exceed state-of-the-art performance for joint NER and RE on 5 datasets across 3 domains.

2 The Model

Figure 1 illustrates the architecture of our approach. Our model is composed of an NER module and an RE module. The NER module is identical to the one proposed by Devlin et al. (2018). In short, for a given input sequence s of N word tokens w_1, w_2, …, w_N, the pre-trained BERT$_{\text{BASE}}$ model first produces a sequence of vectors, $\boldsymbol{x}_1^{(\text{NER})}, \boldsymbol{x}_2^{(\text{NER})}, \ldots, \boldsymbol{x}_N^{(\text{NER})}$ which are then fed to a feed-forward neural network (FFNN) for classification.

$$\boldsymbol{s}_i^{(\text{NER})} = \text{FFNN}_{\text{NER}}(\boldsymbol{x}_i^{(\text{NER})}) \tag{1}$$

The output size of this layer is the number of BIOES-based NER labels in the training data, $|C^{(\text{NER})}|$. In the BIOES tag scheme, each token is assigned a label, where the B- tag indicates the beginning of an entity span, I- the inside, E- the end and S- is used for any single-token entity. All other tokens are assigned the label O.

During training, a cross-entropy loss is computed for the NER objective,

$$\mathcal{L}_{\text{NER}} = -\sum_{n=1}^{N} \log \left(\frac{e^{\boldsymbol{s}_n^{(\text{NER})}}}{\sum_c^{C^{(\text{NER})}} e^{\boldsymbol{s}_{n,c}^{(\text{NER})}}} \right) \tag{2}$$

where $\boldsymbol{s}_n^{(\text{NER})}$ is the predicted score that token $n \in N$ belongs to the ground-truth entity class and $\boldsymbol{s}_{n,c}^{(\text{NER})}$ is the predicted score for token n belonging to the entity class $c \in C^{(\text{NER})}$.

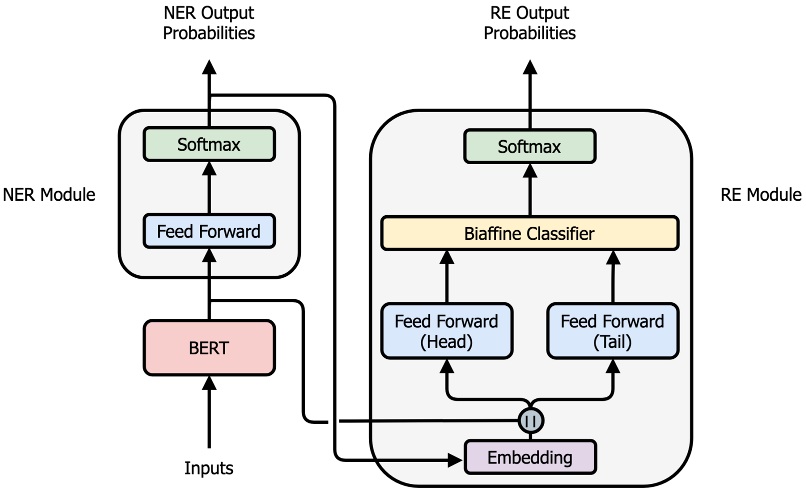

Figure 1: Architecture of our model for joint named entity recognition (NER) and relation extraction (RE).

In the RE module, the predicted entity labels are obtained by taking the argmax of each score vector $\boldsymbol{s}_1^{(\text{NER})}$, $\boldsymbol{s}_2^{(\text{NER})}$, . . ., $\boldsymbol{s}_N^{(\text{NER})}$. The predicted entity labels are then embedded to produce a sequence of vectors $\boldsymbol{e}_1^{(\text{NER})}$, $\boldsymbol{e}_2^{(\text{NER})}$, . . ., $\boldsymbol{e}_N^{(\text{NER})}$ and concatenated with the hidden states from the final layer in the BERT model.

$$\boldsymbol{x}_i^{(\text{RE})} = \boldsymbol{x}_i^{(\text{NER})} \parallel \boldsymbol{e}_i^{(\text{NER})} \tag{3}$$

Following Miwa & Bansal (2016) and Nguyen & Verspoor (2019), we incrementally construct the set of relation candidates, R, using all possible combinations of the last word tokens of predicted entities, i.e. words with E- or S- labels. An entity pair is assigned to a negative relation class (NEG) when the pair has no relation or when the predicted entities are not correct. Once relation candidates are constructed, classification is performed with a deep bilinear attention mechanism (Dozat & Manning, 2016), as proposed by Nguyen & Verspoor (2019).

First, to encode directionality, the mechanism uses two FFNNs to project each $\boldsymbol{x}_i^{(\text{RE})}$ into *head* and *tail* vector representations, which correspond to whether the i^{th} word serves as the head or tail argument of the relation.

$$\boldsymbol{h}_i^{(\text{head})} = \text{FFNN}_{\text{head}}(\boldsymbol{x}_i^{(\text{RE})}) \tag{4}$$

$$\boldsymbol{h}_i^{(\text{tail})} = \text{FFNN}_{\text{tail}}(\boldsymbol{x}_i^{(\text{RE})}) \tag{5}$$

These projections are then fed to a biaffine classifier,

$$\boldsymbol{s}_{j,k}^{(\text{RE})} = \text{Biaffine}(\boldsymbol{h}_j^{(\text{head})}, \boldsymbol{h}_k^{(\text{tail})}) \tag{6}$$

$$\text{Biaffine}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \mathbf{U} \boldsymbol{x}_2 + \boldsymbol{W}(\boldsymbol{x}_1 \parallel \boldsymbol{x}_2) + \boldsymbol{b} \tag{7}$$

where $\mathbf{U}$ is an $m \times |C^{(\text{RE})}| \times m$ tensor, $\boldsymbol{W}$ is a $|C^{(\text{RE})}| \times (2 * m)$ matrix, and $\boldsymbol{b}$ is a bias vector. Here, m is the size of the output layers of $\text{FFNN}_{\text{head}}$ and $\text{FFNN}_{\text{tail}}$ and $C^{(\text{RE})}$ is the set of all relation classes (including NEG). During training, a second cross-entropy loss is computed for the RE objective

$$\mathcal{L}_{\text{RE}} = -\sum_{r=1}^{R} \log \left(\frac{e^{\boldsymbol{s}_r^{(\text{RE})}}}{\sum_c^{C^{(\text{RE})}} e^{\boldsymbol{s}_{r,c}^{(\text{RE})}}} \right) \tag{8}$$

where $\boldsymbol{s}_r^{(\text{RE})}$ is the predicted score that relation candidate $r \in R$ belongs to the ground-truth relation class and $\boldsymbol{s}_{r,c}^{(\text{RE})}$ is the predicted score for relation r belonging to the relation class $c \in C^{(\text{RE})}$.

The model is trained in an end-to-end fashion to minimize the sum of the NER and RE losses.

$$\mathcal{L} = \mathcal{L}_{\text{NER}} + \mathcal{L}_{\text{RE}} \tag{9}$$

2.1 Entity Pretraining

In Miwa & Bansal (2016), entity pre-training is proposed as a solution to the problem of low-performance entity detection in the early stages of training. It is implemented by delaying the training of the RE module by some number of epochs, before training the entire model jointly.

Our implementation of entity pretraining is slightly different. Instead of delaying training of the RE module by some number of epochs, we weight the contribution of $\mathcal{L}_{\text{RE}}$ to the total loss during the first epoch of training,

$$\mathcal{L} = \mathcal{L}_{\text{NER}} + \lambda\, \mathcal{L}_{\text{RE}} \tag{10}$$

where λ is increased linearly from 0 to 1 during the first epoch and set to 1 for the remaining epochs. We chose this scheme because the NER module quickly achieves good performance for all datasets (i.e., within one epoch). In early experiments, we found this scheme to outperform a delay of a full epoch.

2.2 Implementation

We implemented our model in PyTorch (Paszke et al., 2017) using the $\text{BERT}_{\text{BASE}}$ model from the PyTorch Transformers library[1]. Our model is available at our GitHub repository[2]. Furthermore, we use NVIDIAs automatic mixed precision (AMP) library Apex[3] to speed up training and reduce memory usage without affecting task-specific performance.

3 Experimental Setup

3.1 Datasets and evaluation

To demonstrate the generalizability of our model, we evaluate it on 5 commonly used benchmark corpora across 3 domains. All corpora are in English. Detailed corpus statistics are presented in Table A.1 of the appendix.

3.1.1 ACE04/05

The Automatic Content Extraction (ACE04) corpus was introduced by Doddington et al. (2004), and is commonly used to benchmark NER and RE methods. There are 7 entity types and 7 relation types. ACE05 builds on ACE04, splitting the *Physical* relation into two classes (*Physical* and *Part-Whole*), removing the *Discourse* relation class and merging *Employment-Membership-Subsidiary* and *Person-Organization-Affiliation* into one class (*Employment-Membership-Subsidiary*).

For ACE04, we follow Miwa & Bansal (2016) by removing the *Discourse* relation and evaluating our model using 5-fold cross-validation on the *bnews* and *nwire* subsets, where 10% of the data was held out within each fold as a validation set. For ACE05, we use the same test split as Miwa & Bansal (2016). We use 5-fold cross-validation on the remaining data to choose the hyperparameters. Once hyperparameters are chosen, we train on the combined data from all the folds and evaluate on the test set. For both corpora, we report the micro-averaged F_1 score. We obtained the pre-processing scripts from Miwa & Bansal (2016)[4].

[1] https://github.com/huggingface/pytorch-transformers

[2] to-add-to-camera-ready-version

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

[4] https://github.com/tticoin/LSTM-ER/tree/master/data

3.1.2 CoNLL04

The CoNLL04 corpus was introduced in Roth & Yih (2004) for learning the problem of simultaneously recognizing entities and relations, and consists of articles from the Wall Street Journal (WSJ) and Associated Press (AP). There are 4 entity types and 5 relation types.

As there are no official splits, we use the same test set split as Miwa & Sasaki (2014)[5]. We use 5-fold cross-validation on the remaining data to choose hyperparameters. Once hyperparameters are chosen, we train on the combined data from all folds and evaluate on the test set, reporting the micro-averaged F_1 score.

3.1.3 ADE

The adverse drug event corpus was introduced by Gurulingappa et al. (2012) to serve as a benchmark for systems which aim to identify adverse drug events from free-text. It consists of the abstracts of medical case reports retrieved from PubMed[6]. There are two entity types, *Drug* and *Adverse effect* and one relation type, *Adverse drug event*.

Similar to previous work (Li et al., 2016; 2017; Bekoulis et al., 2018b), we remove ∼130 relations with overlapping entities and evaluate our model using 10-fold cross-validation, where 10% of the data within each fold was used as a validation set, 10% as a test set and the remaining data is used as a train set. We report the macro F_1 score averaged across all folds.

3.1.4 i2b2

The 2010 i2b2/VA dataset was introduced by Uzuner et al. (2011) for the 2010 i2b2/Va Workshop on Natural Processing Challenges for Clinical Records. The workshop contained an NER task focused on the extraction of 3 medical entity types (*Problem*, *Treatment*, *Test*) and an RE task for 8 relation types.

In the official splits, the test set contains roughly twice as many examples as the train set. To increase the number of training examples while maintaining a rigorous evaluation, we elected to perform 5-fold cross-validation on the combined data from both partitions. We used 10% of the data within each fold as a validation set, 20% as a test set and the remaining data was used as a train set. We report the micro F_1 score averaged across all folds.

To the best of our knowledge, we are the first to evaluate a joint NER and RE model on the 2010 i2b2/VA dataset. Therefore, we decided to compare to scores obtained by independent NER and RE systems. We note, however, that the scores of independent RE systems are not directly comparable to the scores we report in this paper. This is because RE is traditionally framed as a sentence-level classification problem. During pre-processing, each example is permutated into processed examples containing two "blinded" entities and labelled for one relation class. E.g. the example: "His PCP had recently started ciprofloxacin$_{\text{TREATMENT}}$ for a UTI$_{\text{PROBLEM}}$" becomes "His PCP had recently started @TREATMENT$ for a @PROBLEM$", where the model is trained to predict the target relation type, "Treatment is administered for medical problem" (TrAP).

We argue that this task is inherently easier than the joint setup, for two reasons: relation predictions are made on ground-truth entities, as opposed to predicted entities (which are noisy) and the model is only required to make one classification decision per pre-processed sentence. In the joint setup, a model must identify any number of relations (or the lack thereof) between all unique pairs of *predicted* entities in a given input sentence. To control for the first of these differences, we report scores from our model in two settings, once when predicted entities are used as input to the RE module, and once when ground-truth entities are used.

[5] `http://cistern.cis.lmu.de/globalNormalization/globalNormalization_all.zip`

[6] `https://www.ncbi.nlm.nih.gov/pubmed`

3.2 Hyperparameters

Besides batch size, learning rate and number of training epochs, we used the same hyperparameters across all experiments (see Table A.2). Similar to Devlin et al. (2018), learning rate and batch size were selected for each dataset using a minimal grid search (see See Table A.3).

One hyperparameter selected by hand was the choice of the pre-trained weights used to initialize the $\text{BERT}_{\text{BASE}}$ model. For general domain corpora, we found the cased $\text{BERT}_{\text{BASE}}$ weights from Devlin et al. (2018) to work well. For biomedical corpora, we used the weights from BioBERT (Lee et al., 2019), which recently demonstrated state-of-the-art performance for biomedical NER, RE and QA. Similarly, for clinical corpora we use the weights provided by Peng et al. (2019), who pre-trained $\text{BERT}_{\text{BASE}}$ on PubMed abstracts and clinical notes from MIMIC-III[7].

4 Results

4.1 Jointly learning NER and RE

Table 1 shows our results in comparison to previously published results, grouped by the domain of the evaluated corpus. We find that on every dataset besides i2b2, our model improves NER performance, for an average improvement of $\sim$2%. This improvement is particularly large on the ACE04 and ACE05 corpora (3.98% and 2.41% respectively). On i2b2, our joint model performs within 0.29% of the best independent NER solution. For relation extraction, we outperform previous methods on 2 datasets and come within $\sim$2% on both ACE05 and CoNLL04. In two cases, our performance improvement is substantial, with improvements of 4.59% and 10.25% on the ACE04 and ADE corpora respectively. For i2b2, our score is not directly comparable to previous systems (as discussed in section 3.1.4) but will facilitate future comparisons of joint NER and RE methods on this dataset. By comparing overall performance, we find that our approach achieves new state-of-the-art performance for 3 popular benchmark datasets (ACE04, ACE05, ADE) and comes within 0.2% for CoNLL04.

4.2 Ablation Analysis

To determine which training strategies and components are responsible for our models performance, we conduct an ablation analysis on the CoNLL04 corpus (Table 2). We perform five different ablations: (a) Without entity pre-training (see section 2.1), i.e. the loss function is given by equation 9. (b) Without entity embeddings, i.e. equation 3 becomes $\boldsymbol{x}_i^{(\text{RE})} = \boldsymbol{x}_i^{(\text{NER})}$. (c) Replacing the two feed-forward neural networks, $\text{FFNN}_{\text{head}}$ and $\text{FFNN}_{\text{tail}}$ with a single FFNN (see equation 4 and 5). (d) Removing $\text{FFNN}_{\text{head}}$ and $\text{FFNN}_{\text{tail}}$ entirely. (e) Without the bilinear operation, i.e. equation 7 becomes a simple linear transformation.

Removing $\text{FFNN}_{\text{head}}$ and $\text{FFNN}_{\text{tail}}$ (see equation 4 and 5) has, by far, the largest negative impact on performance. Interestingly, however, replacing $\text{FFNN}_{\text{head}}$ and $\text{FFNN}_{\text{tail}}$ with a single FFNN has only a small negative impact. This suggests that while these layers are very important for model performance, using distinct FFNNs for the projection of head and tail entities (as opposed to the same FFNN) is relatively much less important. The next most impactful ablation was entity pre-training, suggesting that low-performance entity detection during the early stages of training is detrimental to learning (see section 2.1). Finally, we note that the importance of entity embeddings is surprising, as a previous study has found that entity embeddings did not help performance on the CoNLL04 corpus specifically (Bekoulis et al., 2018a), although their architecture was markedly different. We conclude that each of our ablated components is necessary to achieve maximum performance.

4.3 Analysis of the word-level attention weights

One advantage of including a transformer-based language model is that we can easily visualize the attention weights with respect to some input. This visualization is useful, for example, in detecting model bias and locating relevant attention heads (Vig, 2019). Previous works have used such visualizations to demonstrate that specific attention heads mark syntactic dependency relations and

[7] `https://mimic.physionet.org/`

Table 1: Comparison to previously published F_1 scores for joint named entity recognition (NER) and relation extraction (RE). Ours: our full, end-to-end model (see section 2). Ours (gold): our model, when gold entity labels are used as input to the RE module. Bold: best scores. Subscripts denote standard deviation across three runs. Δ: difference to our overall score.

	Dataset	Method	Entity	Relation	Overall	Δ
General	ACE04	Miwa & Bansal (2016)	81.80	48.40	65.10	-5.69
		Bekoulis et al. (2018b)	81.64	47.45	64.54	-6.25
		Li et al. (2019)	83.60	49.40	66.50	-4.29
		Ours	$\mathbf{87.58_{0.2}}$	$\mathbf{53.99_{0.1}}$	**70.79**	–
	ACE05	Miwa & Bansal (2016)	83.40	55.60	69.50	-3.42
		Zhang et al. (2017)	83.50	57.50	70.50	-2.42
		Li et al. (2019)	84.80	**60.20**	72.50	-0.42
		Ours	$\mathbf{87.21_{0.3}}$	$58.63_{0.0}$	**72.92**	–
	CoNLL04	Miwa & Sasaki (2014)	80.70	61.00	70.85	-7.30
		Bekoulis et al. (2018b)	83.61	61.95	72.78	-5.37
		Li et al. (2019)	87.80	**68.90**	**78.35**	0.20
		Ours	$\mathbf{89.46_{1.0}}$	$66.83_{0.4}$	78.15	–
Biomedical	ADE	Li et al. (2016)	79.50	63.40	71.45	-16.21
		Li et al. (2017)	84.60	71.40	78.00	-9.66
		Bekoulis et al. (2018b)	86.73	75.52	81.13	-6.53
		Ours	$\mathbf{89.56_{0.1}}$	$\mathbf{85.77_{0.4}}$	**87.66**	–
Clinical	i2b2*	Si et al. (2019)**	**89.55**	–	–	–
		Peng et al. (2019)	–	76.40	–	–
		Ours (gold)	–	$72.03_{0.1}$	–	–
		Ours	$89.26_{0.1}$	$63.02_{0.3}$	76.14	–

* To the best of our knowledge, there are no published joint NER and RE models that evaluate on the i2b2 2010 corpus. We compare our model to the state-of-the-art for each individual task (see section 3.1.4).

** We compare to the scores achieved by their $\mathrm{BERT}_{\mathrm{BASE}}$ model.

Table 2: Ablation experiment results on the CoNLL04 corpus. Scores are reported as a micro-averaged F_1 score on the validation set, averaged across three runs of 5-fold cross-validation. (a) Without entity pre-training (section 2.1). (b) Without entity embeddings (Eq. 3). (c) Using a single FFNN in place of $\mathrm{FFNN}_{\mathrm{head}}$ and $\mathrm{FFNN}_{\mathrm{tail}}$ (Eq. 4 and 5) (d) Without $\mathrm{FFNN}_{\mathrm{head}}$ and $\mathrm{FFNN}_{\mathrm{tail}}$ (e) Without the bilinear operation (Eq. 7). Bold: best scores. Subscripts denote standard deviation across three runs. Δ: difference to the full models score.

Model	Entity	Relation	Overall	Δ
Full model	$86.32_{0.2}$	$\mathbf{60.29_{0.2}}$	**73.30**	–
(a) w/o Entity pre-training	$85.91_{0.1}$	$59.44_{0.2}$	72.67	-0.63
(b) w/o Entity embeddings	$86.07_{0.1}$	$59.52_{0.2}$	72.80	-0.51
(c) Single FFNN	$86.02_{0.1}$	$60.13_{0.0}$	73.07	-0.23
(d) w/o Head/Tail	$83.93_{0.2}$	$54.67_{0.8}$	69.30	-4.00
(e) w/o Bilinear	$\mathbf{86.35_{0.1}}$	$59.60_{0.0}$	72.98	-0.33

that lower layers tend to learn more about syntax while higher layers tend to encode more semantics (Raganato & Tiedemann, 2018).

In Figure 2 we visualize the attention weights of select layers and attention heads from an instance of BERT fine-tuned within our model on the CoNLL04 corpus. We display four patterns that are easily interpreted: paying attention to the next and previous word, paying attention to the word itself, and paying attention to the end of the sentence. These same patterns have been found in pre-trained

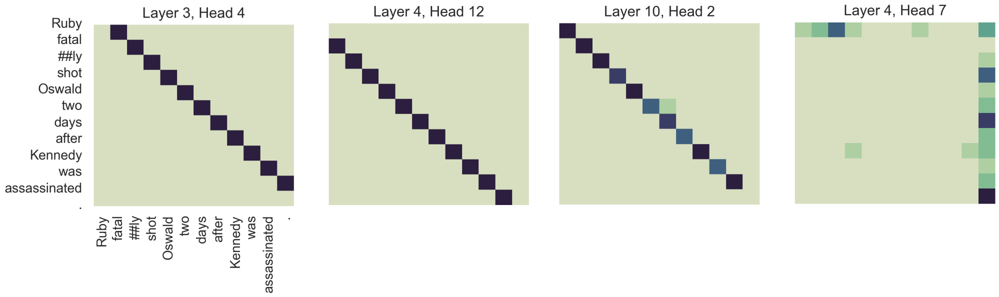

Figure 2: Visualization of the attention weights from select layers and heads of BERT after it was fine-tuned within our model on the CoNLL04 corpus. Darker squares indicate larger attention weights. Attention weights are shown for the input sentence: ”Ruby fatally shot Oswald two days after Kennedy was assassinated.”. The CLS and SEP tokens have been removed. Four major patterns are displayed: paying attention to the next word (first image from the left) and previous word (second from the left), paying attention to the word itself (third from the left) and to the end of the sentence (fourth from the left).

BERT models that have not been fine-tuned on a specific, supervised task (Vig, 2019; Raganato & Tiedemann, 2018), and therefore, are retained after our fine-tuning procedure.

To facilitate further analysis of our learned model, we make available Jupyter and Google Colaboratory notebooks on our GitHub repository[8], where users can use multiple views to explore the learned attention weights of our models. We use the BertViz library (Vig, 2019) to render the interactive, HTML-based views and to access the attention weights used to plot the heat maps.

5 Discussion and Conclusion

In this paper, we introduced an end-to-end model for entity and relation extraction. Our key contributions are: (1) No reliance on any hand-crafted features (e.g. templated questions) or external NLP tools (e.g. dependency parsers). (2) Integration of a pre-trained, transformer-based language model. (3) State-of-the-art performance on 5 datasets across 3 domains. Furthermore, our model is inherently modular. One can easily initialize the language model with pre-trained weights better suited for a domain of interest (e.g. BioBERT for biomedical corpora) or swap BERT for a comparable language model (e.g. XLNet (Yang et al., 2019)). Finally, because of (2), our model is fast to train, converging in approximately 1 hour or less on a single GPU for all datasets used in this study.

Our model out-performed previous state-of-the-art performance on ADE by the largest margin (6.53%). While exciting, we believe this corpus was particularly easy to learn. The majority of sentences ($\sim$68%) are annotated for two entities (*drug* and *adverse effect*, and one relation (*adverse drug event*). Ostensibly, a model should be able to exploit this pattern to get near-perfect performance on the majority of sentences in the corpus. As a test, we ran our model again, this time using ground-truth entities in the RE module (as opposed to predicted entities) and found that the model very quickly reached almost perfect performance for RE on the test set ($\sim$98%). As such, high performance on the ADE corpus is not likely to transfer to real-world scenarios involving the large-scale annotation of diverse biomedical articles.

In our experiments, we consider only intra-sentence relations. However, the multiple entities within a document generally exhibit complex, inter-sentence relations. Our model is not currently capable of extracting such inter-sentence relations and therefore our restriction to intra-sentence relations will limit its usefulness for certain downstream tasks, such as knowledge base creation. We also ignore the problem of nested entities, which are common in biomedical corpora. In the future, we would like to extend our model to handle both nested entities and inter-sentence relations. Finally, given that multilingual, pre-trained weights for BERT exist, we would also expect our model's performance to hold across multiple languages. We leave this question to future work.

[8] `to-add-to-camera-ready-version`

REFERENCES

Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Van Loo, Bert Coessens, Frederik De Smet, Leon-Charles Tranchevent, Bart De Moor, Peter Marynen, Bassem Hassan, Peter Carmeliet, and Yves Moreau. Gene prioritization through genomic data fusion. *Nature Biotechnology*, 24(5): 537–544, may 2006. doi: 10.1038/nbt1203.

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, 2018a.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. Adversarial training for multi-context joint entity and relation extraction. *arXiv preprint arXiv:1808.06876*, 2018b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf`.

Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*, 2016.

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. *Journal of Biomedical Informatics*, 45(5):885 – 892, 2012. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2012.04.008. URL `http://www.sciencedirect.com/science/article/pii/S1532046412000615`. Text Mining and Natural Language Processing in Pharmacogenomics.

Jing Jiang. *Information Extraction from Text*, pp. 11–41. Springer US, Boston, MA, 2012. ISBN 978-1-4614-3223-4. doi: 10.1007/978-1-4614-3223-4_2. URL `https://doi.org/10.1007/978-1-4614-3223-4_2`.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, 2019.

Fei Li, Yue Zhang, Meishan Zhang, and Donghong Ji. Joint models for extracting adverse drug events from biomedical text. In *IJCAI*, volume 2016, pp. 2838–2844, 2016.

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.

Gang Li, Karen E. Ross, Cecilia N. Arighi, Yifan Peng, Cathy H. Wu, and K. Vijay-Shanker. miRTex: A text mining system for miRNA-gene relation extraction. *PLOS Computational Biology*, 11(9):e1004391, sep 2015. doi: 10.1371/journal.pcbi.1004391.

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. *arXiv preprint arXiv:1905.05529*, 2019.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. URL `http://arxiv.org/abs/1711.05101`.

Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2017.

Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*, 2016.

Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1858–1869, 2014.

Dat Quoc Nguyen and Karin Verspoor. End-to-end neural relation extraction using deep biaffine attention. In Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra (eds.), *Advances in Information Retrieval*, pp. 729–738, Cham, 2019. Springer International Publishing. ISBN 978-3-030-15712-8.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pp. 1310–1318, 2013.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and elmo on ten benchmarking datasets. *CoRR*, abs/1906.05474, 2019. URL http://arxiv.org/abs/1906.05474.

Alessandro Raganato and Jörg Tiedemann. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 287–297, 2018.

Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pp. 1–8, 2004.

Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. Enhancing clinical concept extraction with contextual embedding. *CoRR*, abs/1902.08691, 2019. URL http://arxiv.org/abs/1902.08691.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011.

Jesse Vig. A multiscale visualization of attention in the transformer model. *CoRR*, abs/1906.05714, 2019. URL http://arxiv.org/abs/1906.05714.

Zhong-Yi Wang and Hong-Yu Zhang. Rational drug repositioning by medical genetics. *Nature Biotechnology*, 31(12):1080–1082, dec 2013. doi: 10.1038/nbt.2758.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

Meishan Zhang, Yue Zhang, and Guohong Fu. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1730–1740, 2017.

XueZhong Zhou, Jrg Menche, Albert-László Barabási, and Amitabh Sharma. Human symptoms–disease network. *Nature Communications*, 5, jun 2014. doi: 10.1038/ncomms5212.

A Appendix

Table A.1: Detailed entity and relation counts for each corpus used in this study.

	Dataset	Entity classes (count)	Relation classes (count)
General	ACE04	Person (12508), Organization (4405), Geographical Entities (4425), Location (614), Facility (688), Weapon (119), and Vehicle (209)	Physical (1202), Person-Social (362), Employment-Membership-Subsidiary (1591), Agent-Artifact (212), Person-Organization-Affiliation (141), Geopolitical Entity-Affiliation (517)
	ACE05	Person (20891), Organization (5627), Geographical Entities (7455), Location(1119), Facility (1461), Weapon (911), and Vehicle (919)	Physical (1612), Part-Whole (1060), Person-Social (615), Agent-Artifact (703), Employment-Membership-Subsidiary (1922), Geopolitical Entity-Affiliation (730)
	CoNLL04	Location (4765), Organization (2499), People (3918), Other (3011)	Kill (268), Live in (521), Located in (406), OrgBased in (452), Work for (401)
Biomedical	ADE	Drug (4979), Adverse effect (5669)	Adverse drug event (6682)
Clinical	i2b2	Problem (19664), Test (13831), Treatment (14186)	PIP (2203), TeCP (504), TeRP (3053), TrAP (2617), TrCP (526), TrIP (203), TrNAP (174), TrWP (133)

Table A.2: Hyperparameter values and model details used across all experiments.

Hyperparameter	Value	Comment
Tagging scheme	BIOES	Single token entities are tagged with an S- tag, the beginning of an entity span with a B- tag, the last token of an entity span with an E- tag, and tokens inside an entity span with an I- tag.
Dropout rate	0.1	Dropout rate applied to the output of all FFNNs and the attention heads of the BERT model.
Entity embeddings	128	Output dimension of the entity embedding layer.
$\text{FFNN}_{\text{head}}$ / $\text{FFNN}_{\text{tail}}$	512	Output dimension of the $\text{FFNN}_{\text{head}}$ and $\text{FFNN}_{\text{tail}}$ layers
No. layers (NER module)	1	Number of layers used in the FFNN of the NER module.
No. layers (RE module)	2	Number of layers used in the FFNNs of the RE module.
Optimizer	AdamW	Adam with fixed weight decay regulatization (Loshchilov & Hutter, 2017).
Gradient normalization	$\Gamma = 1$	Rescales the gradient whenever the norm goes over some threshold Γ (Pascanu et al., 2013).
Weight decay	0.1	L2 weight decay.

Table A.3: Hyperparameter values specific to individual datasets. Similar to Devlin et al. (2018), a minimal grid search was performed over the values 16, 32 for batch size and 2e-5, 3e-5, and 5e-5 for learning rate.

Dataset	No. Epochs	Batch size	Learning rate	Initial BERT weights
ACE04	15	16	2e-5	BERT-Base (cased) (Devlin et al., 2018)
ACE05	15	32	3e-5	BERT-Base (cased)
CoNLL04	10	16	3e-5	BERT-Base (cased)
ADE	7	16	2e-5	BioBERT (cased) (Lee et al., 2019)
i2b2	12	16	2e-5	NCBI-BERT (uncased) (Peng et al., 2019)