Revisiting How to Focus: Triplet Attention for Joint Entity and Relation Extraction

Debraj Basu
dbasu@adobe.com
Adobe
San Jose, CA, USA

Meghanath Macha
yadagiri@adobe.com
Adobe
San Jose, CA, USA

Deepak Pai
dpai@adobe.com
Adobe
San Jose, CA, USA

ABSTRACT

We propose a method for extracting entities and relations from natural language. When put together, this results in fact-triplets of the form *subject*, *predicate* and *object* as knowledge units. Our method benefits from *triplet attention* in addition to conventional self-attention as a feature refinement mechanism. We do this by explicitly facilitating contextual cues for every candidate entity span and every *subject* and *object* pair, which are allowed to attend to each token of the sentence besides attention between any two tokens. In conjunction with sharing information between the two tasks and the benefits of transfer learning, our method exhibits competitive performance in strict evaluation, compared to the previous state-of-the-art, with improvements up to 2.6% and 3.4% in micro and macro-F1 for entity recognition, as well as 6.9% and 5.9% in micro and macro-F1 respectively for relation extraction.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**.

KEYWORDS

information extraction, multi-task learning

1 INTRODUCTION

The ability to continuously scrutinize, infer, store and retrieve knowledge about observations from the real world characterizes human intelligence. It enables us to build and leverage priors to make well-informed decisions.

A similar characterization of business intelligence relies on its ability to do the same programmatically. The diversity and volume of natural language text are rapidly increasing. As a result, extracting knowledge from it efficiently and at scale has become the modern workhorse of business intelligence.

The characterization of an elementary unit of knowledge is a fact triplet, comprising ⟨*subject, predicate, object*⟩. The *subject* and the *object* are typically entities, often named and typed, and represented as nodes. The *predicate* is represented as a directed edge and describes how the *subject* is related to the *object*. An example would be ⟨*Barack Obama:Person, born in, Honolulu:Location*⟩

These triplets, when appropriately combined, become elementary units of a much larger data structure, known as a *knowledge graph* [1]. The *knowledge graph* serves as a significantly rich source of knowledge for many downstream applications, as discussed in [13]. Every new entity and fact triplet extracted from reading natural language refines our understanding of the world by updating the current state of the *knowledge graph*.

Machine learning systems have made significant strides in the problem of joint entity recognition and relation extraction, benefiting from advances in contextual representations and transfer learning. Creative design has resulted in model architectures for learning sufficiently discriminative representations for recognizing typed entities and extracting the typed relations between them.

Related Work. The general framework of solutions has rapidly evolved from pipelined sequential systems lacking in learned interactions between entity recognition and relation extraction tasks [2, 7, 25], often with dated recurrent methods [10, 12, 20]. Often, the task performed early on in the pipeline did not benefit from the learning process of the latter tasks, as discussed in [26, Section 2].

Recently, more nuanced transformer-based methods leverage beneficial two-way intricate interactions between the tasks through joint training for discriminative labeling [22, 26, 27] and auto-regressive triplet generation [5]. As done recently in [26], our method generates predictions for discriminatively labeling every span as a possible entity and for labeling every pair of spans as a possible relation.

Our work is most closely related to recent state-of-the-art [26], which discusses the observed merits of the relation signal for recognizing entities. We demonstrate improvements in the strict evaluation F1 score from incorporating triplet attention in Section 3.2.1.

Contributions. Motivated by the conventional attention mechanism from [23], our proposal enables learning where to focus for extracting a given fact triplet or a candidate entity span (see Figure 1b), in its entirety as opposed to where to focus for the tokens individually (see Figure 1a). We call it *triplet attention*, and to our best knowledge, our method is the first to study its merits when combined with information sharing between tasks in the joint learning paradigm.

Our findings in Section 4, particularly in Table 1 and Table 2 suggest that enabling candidate entity spans and pairs of *subject* and *object* to attend to every token in a sentence facilitates more explicit contextual cues for both discriminative labeling tasks. The possible memory cost incurred can be conveniently handled using the simple trick from [18] while preserving equivalence.

Results of our ablation study in Table 2 demonstrate the incremental improvements of the triplet attention mechanism over information sharing and its base variant.

2 PROBLEM

In our problem setup, we extract the set of all entities $\mathcal{E}$, and the set of all fact triplets $\mathcal{R}$ present in a sentence $\mathcal{S}$. The sentence $\mathcal{S} = \{w_1, \ldots, w_L\}$ is an ordered sequence of L tokens.

An entity is extracted by classifying each triplet $\langle w_i, e, w_j \rangle$ into $\{0, 1\}$, where w_i and w_j are the start and end tokens of a span, and $e \in C_{\mathcal{E}}$ is an entity type, $C_{\mathcal{E}}$ is the set of all possible entity types.

Similarly, we also extract a triplet by extracting elements belonging to two sets, denoted by $\mathcal{R}_h$ and $\mathcal{R}_t$. We extract each element of $\mathcal{R}_h$ by classifying each triplet $\langle w_i, r, w_j \rangle$ into mutually exclusive classes 0 or 1, where w_i and w_j are the head tokens of the *subject* and the object, respectively, $r \in C_{\mathcal{R}}$ is a predicate or relation type, and $C_{\mathcal{R}}$ is the set of all possible relation types. When $\langle w_i, r, w_j \rangle$ is classified as 1, it implies that two entities exist, beginning with w_i and w_j respectively, which are related by r.

The definition of $\mathcal{R}_t$ follows for the tail tokens of the *subject* and *object* along the same lines. As done previously in [5, 7, 26], we also assume that both the *subject* and *object* in any fact triplet present in the ground truth, always belong to $\mathcal{E}$.

A strict inference protocol is used for constructing the predicted $\mathcal{E}$ and $\mathcal{R}$. We first construct $\mathcal{E}$, $\mathcal{R}_h$, and $\mathcal{R}_t$ as maximum likelihood estimates from the model's output for $\mathcal{S}$. This is followed by constructing $\mathcal{R}$, by including only those triplets $\langle$*subject, predicate, object*$\rangle$ that agree with all of the predicted $\mathcal{E}$, $\mathcal{R}_h$ and $\mathcal{R}_t$.

Notation. Deferred to Appendix A.

3 MODEL ARCHITECTURE

Along the lines of [7, 24, 26], we designed a system that generates sufficiently discriminative features for every $\langle w_i, e, w_j \rangle$ and $\langle w_i, r, w_j \rangle$, from a sentence $\mathcal{S}$, used for linearly separating them into the classes $\{0, 1\}$, as described in Section 2.

Our methodology involves three key stages, which are performed upon the token level embeddings from a well-known language model's encoder[1] such as BERT, ALBERT, SCIBERT, etc. sequentially. The three stages are given by (1), (2) and (3).

These embeddings are first transformed into three features per token, for entity recognition, relation extraction, and shared.

A refinement module transforms these features into highly discriminative features for each element of the form $\langle w_i, e, w_j \rangle$ or $\langle w_i, r, w_j \rangle$, by having the feature corresponding to each pair $(i, j) \in [L]^2$ attend to every token $i \in [L]$ in the sentence. Our experiments demonstrate the advantages of incorporating this module.

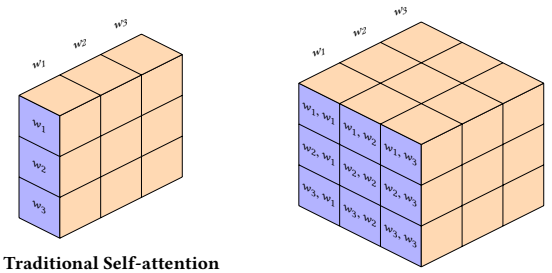
In the final stage, we utilize two task heads, one for each of the tasks corresponding to entity recognition and relation extraction, which is nothing but a linear layer followed by a sigmoid function trying to predict the presence or absence of $\langle w_i, e, w_j \rangle$ or $\langle w_i, r, w_j \rangle$.

3.1 Task Separation

Let $f_{\mathcal{S}} \coloneqq \texttt{BERT}(\mathcal{S}) \in \mathbb{R}^{L \times d}$ where L denotes the number of tokens in $\mathcal{S}$, and d is the dimensionality of the per token embeddings obtained from BERT [6]. We compute the following features as a transformation of $f_{\mathcal{S}}$,

$$\begin{aligned} f_{\texttt{ner}} &= g_{\texttt{ner}}(f_{\mathcal{S}}) \\ f_{\texttt{re}} &= g_{\texttt{re}}(f_{\mathcal{S}}) \\ f_{\texttt{shared}} &= g_{\texttt{shared}}(f_{\mathcal{S}}) \end{aligned} \tag{1}$$

[1]For the rest of this paper, we will refer to this model as BERT for convenience and ease of understanding. However, note that it is interchangeable, as shown in Section 4.



(a) Traditional Self-attention **(b) Triplet attention**

Figure 1 For a sentence with three tokens, blue corresponds to *query* and orange corresponds to attention entries for the *key*. Each cube in the grid can be denoted by Att$[w_i, w_k]$ in Figure 1a and Att$[w_i, w_j, w_k]$ in Figure 1b represents an attention value. Consequently we have $\sum_{k=1}^{3}$ Att$[\ldots, w_k] = 1$. Note the differences in the query between the two figures, where in Figure 1b each query is associated with a pair of tokens, unlike in Figure 1a. A pair of tokens can be a pair of either head or tail tokens of the subject and object corresponding to the candidate fact triplet, for example (w_h = *Barack*, w_h = *Honolulu*) or (w_t = *Obama*, w_t = *Honolulu*). Or it could even correspond to a pair of head and tail tokens of a candidate entity span, for example (w_h = *Barack*, w_t = *Obama*).

Where the parametrized functional form of g is as follows,

$$g : x \in \mathbb{R}^d \rightarrow \texttt{tanh}[\texttt{Linear}(\texttt{TransformerEncoderLayer}(x))] \in \mathbb{R}^{d'}$$

Here the TransformerEncoderLayer is as defined in [23, Section 3.1], tanh activation is as defined in [17, Section III-B] and d' is the feature dimensionality, tuned for performance. Each g is responsible for inducing a separation of the representation $f_{\mathcal{S}}$ into task specific features $f_{\texttt{ner}} \in \mathbb{R}^{L \times d'}$ and $f_{\texttt{re}} \in \mathbb{R}^{L \times d'}$ along with features $f_{\texttt{shared}} \in \mathbb{R}^{L \times d'}$ shared by the two.

[26] uses a recurrent encoder network for inducing dynamic partitioning of neurons between entity recognition, relation extraction and shared features. On the other hand, our proposal in (1) involves learning three parallel encoders of $f_{\mathcal{S}}$ instead. $f_{\mathcal{S}}$ is the common representation of $\mathcal{S}$ which is subsequently processed by $g_{\texttt{ner}}$ and $g_{\texttt{re}}$ for distilling the respective task specific discriminative features.

By providing $f_{\texttt{shared}}$ to both tasks, we are allowing the possibility of another common representation which by itself is a distilled version of $f_{\mathcal{S}}$ through a non-linear mechanism $g_{\texttt{shared}}$. Table 2 demonstrates improvement in performance for the joint task.

3.2 Feature Refinement

Here our goal is to design a feature refinement module which generates discriminative embeddings as a function of $f_{\texttt{ner}}$ and $f_{\texttt{shared}}$ for entity recognition, and as a function of $f_{\texttt{re}}$ and $f_{\texttt{shared}}$ for relation extraction.

We first construct per token features $c \in \mathbb{R}^{L \times d'}$ and per token pair features $\widetilde{c} \in \mathbb{R}^{L \times L \times d'}$ using mechanisms discussed in Appendix B and use them as inputs to our triplet attention module.

3.2.1 **Triplet attention**. Traditionally, attention involves each token as a query, for which we attend to all keys and obtain a weighted average over their values. In this paper, since our focus is on classifying elements of the form $\langle w_i, e, w_j \rangle$ or $\langle w_i, r, w_j \rangle$, a natural extension is to have each of them serve as a query, attending to all tokens present in the sentence as denoted in $\widehat{c}$ =

`TransformerEncoderLayer`$(\widetilde{c}, c)$. Here $\widetilde{c}$ governs the query, and c governs the key and value for the attention mechanism. The `TransformerEncoderLayer` module here implements cross-attention instead of self-attention. This enables the representations derived for $\langle w_i, e, w_j \rangle$ and $\langle w_i, r, w_j \rangle$ to directly benefit by attending to the contextual representation of every token in the sentence[2].

This is followed by a fully connected layer with ELU activation [17, Section III-G], $\widehat{c} \leftarrow \texttt{ELU}(\tanh(\widehat{c}))$.

The composition of the modules described in this section, as a function of a and b are denoted by the feature refinement module $G : (a, b) \in \mathbb{R}^{L \times d'^2} \rightarrow G(a, b) \in \mathbb{R}^{L \times L \times d'}$. Using this parametrized functional form, we obtain the following

$$\begin{aligned} \widehat{f}_{\texttt{ner}} &= G_{\texttt{ner}}(f_{\texttt{ner}}, f_{\texttt{shared}}) \\ \widehat{f}^{h}_{\texttt{re}} &= G^{h}_{\texttt{re}}(f_{\texttt{re}}, f_{\texttt{shared}}) \\ \widehat{f}^{t}_{\texttt{re}} &= G^{t}_{\texttt{re}}(f_{\texttt{re}}, f_{\texttt{shared}}) \end{aligned} \tag{2}$$

where $\widehat{f}_{\texttt{ner}}$, $\widehat{f}^{h}_{\texttt{re}}$ and $\widehat{f}^{t}_{\texttt{re}}$ are the discriminative features required for constructing $\mathcal{E}$, $\mathcal{R}_h$ and $\mathcal{R}_t$, as defined in Section 2, respectively.

3.3 Task Head

At this stage, we learn linear separations between features of those $\langle w_i, e, w_j \rangle$ and $\langle w_i, r, w_j \rangle$ which agree with the ground truth $\mathcal{E}$, $\mathcal{R}_h$ and $\mathcal{R}_t$, and those which do not. We first define the following events $X(i, e, j) \coloneqq \mathbf{1}\{\langle w_i, e, w_j \rangle \in \mathcal{E}\}$, $Y(i, r, j) \coloneqq \mathbf{1}\{\langle w_i, r, w_j \rangle \in \mathcal{R}_h\}$ and $Z(i, r, j) \coloneqq \mathbf{1}\{\langle w_i, r, w_j \rangle \in \mathcal{R}_t\}$

$$\begin{aligned} \mathbb{P}_m(X(i, e, j) = 1 | \mathcal{S}) &\coloneqq \widehat{G}_{\texttt{ner}}\left(\widehat{f}_{\texttt{ner}}\right)_{ije} \\ \mathbb{P}_m(Y(i, r, j) = 1 | \mathcal{S}) &\coloneqq \widehat{G}^{h}_{\texttt{re}}\left(\widehat{f}^{h}_{\texttt{re}}\right)_{ijr} \\ \mathbb{P}_m(Z(i, r, j) = 1 | \mathcal{S}) &\coloneqq \widehat{G}^{t}_{\texttt{re}}\left(\widehat{f}^{t}_{\texttt{re}}\right)_{ijr} \end{aligned} \tag{3}$$

where $\widehat{G} : x \in \mathbb{R}^{L \times L \times d'} \rightarrow \sigma(\texttt{Linear}(x)) \in [0, 1]^{L \times L \times C}$, C is $C_{\mathcal{R}}$ for $\widehat{G}^{h}_{\texttt{re}}$ and $\widehat{G}^{t}_{\texttt{re}}$, and $C_{\mathcal{E}}$ for $\widehat{G}_{\texttt{ner}}$, σ is the sigmoid function [17, Section III-A]. $\mathbb{P}_m$ is used to denote predicted distribution emitted by the model.

4 EXPERIMENTS

Similar to [5, 7, 26, 27], we evaluated our method on multiple public datasets. We do not train our model using a comprehensive pre-training strategy for relation extraction. As also discussed in Section 2, our method receives raw text $\mathcal{S}$ and emits both $\mathcal{E}$ and $\mathcal{R}$.

4.1 Datasets

We conduct experiments on the WebNLG[8], SciERC[16], NYT[19] and ADE[11] datasets. WebNLG was created for verbalizing a set of fact triplets into freeform text. The SciERC dataset was collected from 500 abstracts of academic publications associated with artificial intelligence and used for constructing scientific knowledge graphs. The NYT dataset was annotated using distant supervision with FreeBase [4], over New York Times articles. The ADE dataset has pairs of drugs and their adverse effects marked as fact triplets.

We have the training, validation, and test splits for WebNLG, SciERC, and NYT datasets as proposed by their respective authors. For ADE, we have ten training and test sets, of which 15% of the training set is used to construct the validation set for each fold.

4.2 Training

To summarize, the `BERT` features $f_{\mathcal{S}}$ are processed in sequence as per (1), (2) and (3). For a sentence $\mathcal{S}$, the binary cross entropy loss[3] for both entity recognition and relation extraction, is optimized.

$$\begin{aligned} Loss = \sum_{i \in [L]} \sum_{j \in [L]} \Big[& \sum_{e \in C_{\mathcal{E}}} l\left(\mathbb{P}_m\left(X(i, e, j) | \mathcal{S}\right), X(i, e, j)\right) \\ & + \sum_{r \in C_{\mathcal{R}}} l\left(\mathbb{P}_m\left(Y(i, r, j) | \mathcal{S}\right), Y(i, r, j)\right) \\ & + \sum_{r \in C_{\mathcal{R}}} l\left(\mathbb{P}_m\left(Z(i, r, j) | \mathcal{S}\right), Z(i, r, j)\right) \Big] \end{aligned} \tag{4}$$

Here l corresponds to the binary cross-entropy loss. To prevent overfitting, we employ dropout [21] of 0.1, for $f_{\mathcal{S}}$ in Section 3.1, as well as the outputs of G in (2).

The model is trained for 100 epochs on the training split, using the `Adam` optimizer [14]. Four Tesla P40 GPUs on a single node were used to train our models in a data-parallel setting. To conform with the experiment setup in [26], we have also leveraged the following pre-trained encoders as candidate backbone encoders for generating $f_{\texttt{ner}}$, $f_{\texttt{re}}$ and $f_{\texttt{shared}}$; `BERT` (`bert-base-cased`, [6]) for experiments on the WebNLG and NYT datasets, `ALBERT` (`albert-xxlarge-v1`, [15]) for the ADE dataset, and `SCIBERT` (`scibert-scivocab-uncased`, [3]) for SciERC dataset.

4.3 Evaluation

As also done in [26], the sum of the micro-F1 score [9] of both entity recognition, and relation extraction is monitored for all three splits after every epoch. Finally, the monitored criterion on the test split, corresponding to the best criterion value on the validation split, is reported as the model performance.

Furthermore, we do this for the ten-fold cross-validation on the ten splits available in the ADE dataset. In addition we also monitor the macro-F1 score [9] for ADE, as done in [26]. The average micro and macro-F1 on the test sets, is reported for ADE. Note that the macro-F1 for relation extraction is the same as the micro-F1, as expected when only one relation type exists. For consistency, our evaluation scripts are borrowed from the implementation of [26].

Our baselines were selected to represent the recent state-of-the-art in joint entity recognition and relation extraction. [5] is an autoregressive relation extraction method involving large-scale pre-training on the distantly supervised REBEL dataset for relation extraction, systematically cleansed using a state-of-the-art natural language inference model. Table 1 provides comparisons with two versions, of which REBEL does not involve the pre-training.

[22] argues against the ordering requirement for *seq2seq* methods and proposes an encoder-decoder framework for predicting all fact triplets at once, demonstrating competitive performance. [27] incorporated relations between two spans by modeling the

[2]The space complexity here is $O(L^3)$, which can be handled through a simple trick discussed in [18], which considerably reduces the memory bottleneck. Although the focus of [18] is on self-attention, the method is also applicable to cross-attention.

[3]$-y \log g(y|x) - (1 - y) \log (1 - g(y|x))$ where $y \in \{0, 1\}$ is the ground truth label and $g(y|x)$ is the model's predicted probability for the class y.

Method	WebNLG		SciERC		NYT		ADE	
	NER	RE	NER	RE	NER	RE	NER	RE
REBEL [5]	-	-	-	-	-	91.8	-	81.7
REBEL$_{pre-training}$ [5]	-	-	-	-	-	92.0	-	<u>82.2</u>
PFN [26]	<u>98.0</u>	<u>93.6</u>	66.8	38.4	**95.8**	<u>92.4</u>	(<u>91.3</u>)	(<u>83.2</u>)
SPN [22]	-	93.4	-	-	-	**92.5**	-	-
PL-Marker [27]	-	-	<u>69.9</u>	<u>41.6</u>	-	-	-	-
Our Method	**98.2**	**94.2**	**72.5**	**41.8**	<u>95.3</u>	91.3	**94.5** (**94.7**)	**89.1** (**89.1**)
Our Method - best baseline	+0.2%	+0.6%	+2.6%	+0.2%	-0.5%	-1.2%	_ (+3.4%)	+6.9% (+5.9%)

Table 1 Our method demonstrates competitive performance in terms of micro-F1 for both entity recognition (NER) and relation extraction (RE) against all other baselines representing recent state-of-the-art on different public datasets. The scores enclosed within brackets for the ADE dataset correspond to the macro-F1 scores, along the lines of [26]. The first and second best methods in each column are in bold and <u>underlined</u> respectively. This reinforces the benefit of information sharing between the two tasks of NER and RE combined with triplet attention in Section 3.2.1.

relationships between the *subject* and *object* pairs for every *subject* with significant improvement in micro-F1.

Our method is most closely related to [26] in motivation and setup of the task, which proposes mechanisms for sharing information through dynamic neuron partitions in the multi-task setting. We propose an alternate information-sharing mechanism as described in Section 3.1, which in conjunction with triplet attention, demonstrates benefits in both entity and relation extraction tasks.

The predicted $\mathcal{E}$ and $\mathcal{R}$ are constructed from the model, as described in Section 2. For this purpose, the maximum likelihood estimates from the model are used as predictions for every $\langle w_i, e, w_j \rangle$ and $\langle w_i, r, w_j \rangle$.

Method	WebNLG		SciERC	
	NER	RE	NER	RE
Base	97.6	93.1	71.3	35.0
Base + 1	97.9	93.5	71.2	36.5
Base + 1 + 2 (Our Method)	98.2	94.2	72.5	41.8

Table 2 Ablations over different variations of the architecture discussed in Section 3 are presented. Base denotes Section 3 with the exclusion of both the f_{shared} in (1) as well as the use of triplet attention Section 3.2.1. Base + 1 incorporates f_{shared} in (1) and Base + 1 + 2 also incorporates the triplet attention. There is evidence of merit in incorporating both 1 and 2.

4.4 Results

Comparison with baselines. As seen in Table 1, our method outperforms the baseline approaches in micro-F1 for both entity recognition and relation extraction on the WebNLG, SciERC, and ADE datasets. The bottom row provides the improvement in F1 scores by comparing our method with the best baseline. The performance is marginally inferior, for the distantly supervised NYT dataset, warranting further investigation.

The baselines [5, 22, 26, 27] have been selected to cover the recent state-of-the-art methods for entity recognition and relation extraction for every dataset. Our method also demonstrates significant improvement in macro-F1 for entity recognition for ADE.

Ablation study. We evaluate the merits of the additional feature sharing f_{share} in Section 3.1 as well as the triplet attention mechanism in Section 3.2.1 by incrementally including them into the architecture, denoted by Base + 1 and Base + 1 + 2 respectively.

The inclusion of f_{share} denoted by Base + 1 in Table 2, shows merit in both relation extraction for both WebNLG and SciERC, as well as in entity recognition for WebNLG. Note that, the absence of this feature does not imply that there is no information sharing. In fact the generated $f_{\mathcal{S}}$ is also a shared representation that gets refined into f_{ner}, f_{re} and through the subsequent modules. We also find it noteworthy, that the results for both Base and Base + 1 are not widely away from the recent state-of-the-art enlisted in Table 1.

This in conjunction with the triplet attention, denoted by Base + 1 + 2, demonstrates significant improvements in the micro-F1 score for both entity recognition and relation extraction.

5 CONCLUSION

We propose a new task unit by enabling the explicit recognition of contextual cues for every candidate entity span and pairs of candidate entities over all tokens in a sentence. When applied generically for joint entity recognition and relation extraction, our method exhibits competitive performance against recent state-of-the-art methods for different public datasets. It also remains to be seen if our method benefits from large-scale pre-training as done in [5].

REFERENCES

[1] [n.d.]. Introducing the Knowledge Graph: things, not strings. https://blog.google/products/search/introducing-knowledge-graph-things-not/. Accessed: 2022-05-26.

[2] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *CoRR* abs/1804.07847 (2018). arXiv:1804.07847 http://arxiv.org/abs/1804.07847

[3] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3613–3618. https://doi.org/10.18653/v1/D19-1371

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*. Association for Computing Machinery, New York, NY, USA, 1247–1250. https://doi.org/10.1145/1376616.1376746

[5] Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 2370–2381. https://doi.org/10.18653/v1/2021.findings-emnlp.204

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[7] Markus Eberts and Adrian Ulges. 2020. Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020) (Frontiers in Artificial Intelligence and Applications)*, Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang (Eds.), Vol. 325. IOS Press, 2006–2013. https://doi.org/10.3233/FAIA200321

[8] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating Training Corpora for NLG Micro-Planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, 179–188. https://doi.org/10.18653/v1/P17-1017

[9] Margherita Grandini, Enrico Bagli, and Giorgio Visani. 2020. Metrics for Multi-Class Classification: an Overview. https://doi.org/10.48550/ARXIV.2008.05756

[10] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table Filling Multi-Task Recurrent Neural Network for Joint Entity and Relation Extraction. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad (Eds.). ACL, 2537–2547. https://aclanthology.org/C16-1239/

[11] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. 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 (2012), 885–892. https://doi.org/10.1016/j.jbi.2012.04.008 Text Mining and Natural Language Processing in Pharmacogenomics.

[12] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR* abs/1508.01991 (2015). arXiv:1508.01991 http://arxiv.org/abs/1508.01991

[13] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2020. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *CoRR* abs/2002.00388 (2020). arXiv:2002.00388 https://arxiv.org/abs/2002.00388

[14] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[15] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=H1eA7AEtvS

[16] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3219–3232. https://doi.org/10.18653/v1/D18-1360

[17] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. 2018. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *CoRR* abs/1811.03378 (2018). arXiv:1811.03378 http://arxiv.org/abs/1811.03378

[18] Markus N. Rabe and Charles Staats. 2021. Self-attention Does Not Need O(n^2) Memory. *CoRR* abs/2112.05682 (2021). arXiv:2112.05682 https://arxiv.org/abs/2112.05682

[19] Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling Relations and Their Mentions without Labeled Text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III (Lecture Notes in Computer Science)*, José L. Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag (Eds.), Vol. 6323. Springer, 148–163. https://doi.org/10.1007/978-3-642-15939-8_10

[20] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. 2019. Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=B1l6qiR5F7

[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15, 56 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

[22] Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. 2020. Joint Entity and Relation Extraction with Set Prediction Networks. *CoRR* abs/2011.01675 (2020). arXiv:2011.01675 https://arxiv.org/abs/2011.01675

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[24] Jue Wang and Wei Lu. 2020. Two are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 1706–1721. https://doi.org/10.18653/v1/2020.emnlp-main.133

[25] Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 1476–1488. https://doi.org/10.18653/v1/2020.acl-main.136

[26] Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A Partition Filter Network for Joint Entity and Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 185–197. https://doi.org/10.18653/v1/2021.emnlp-main.17

[27] Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed Levitated Marker for Entity and Relation Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 4904–4917. https://aclanthology.org/2022.acl-long.337

A NOTATION

We denote the statement, x is defined to be equal to y, by the expression $x \coloneqq y$. Furthermore, for any $n \in \mathbb{Z}^+$, $[n] \coloneqq \{1, 2, \ldots, n\}$. For $d \in \mathbb{Z}^+, k \in \mathbb{Z}^+, \mathbb{R}^{d^k} \coloneqq \mathbb{R}^d \times \ldots \texttt{k times} \ldots \times \mathbb{R}^d$. For two tensors A and B, with identical shape until the penultimate dimension, $[A; B]$ is the concatenation of B after A along the last dimension. The expression $x \leftarrow y$ implies that the result of the expression y on the R.H.S. is assigned to the variable x on the L.H.S., and is used for reducing excessive variables by enabling their reuse. $\mathbf{1}$ represents the indicator function. $\mathbb{P}$ is used to denote probability.

B FEATURE CONSTRUCTION MECHANISMS FOR SECTION 3.2

Motivated by ideas from [26], we first propose and evaluate a naive and generic mechanism as follows:

$$c \leftarrow \widetilde{g}([a; b]) \tag{5}$$

where $\widetilde{g} : (x, y) \in \mathbb{R}^{d'^2} \rightarrow \texttt{tanh}\left(\texttt{Linear}\left([x; y]\right)\right) \in \mathbb{R}^{d'}$, $a \in \mathbb{R}^{L \times d'}$ and $b \in \mathbb{R}^{L \times d'}$.

Motivated by the strategy for generating overall features in [26], we then implement a maxpooling layer for generating the overall features $c^{\texttt{global}} \in \mathbb{R}^{d'}$ as

$$c^{\texttt{global}} = \texttt{maxpool}\left(c^{(1)}, c^{(2)}, \ldots, c^{(L)}\right). \tag{6}$$

Finally the representations $\widetilde{c}(i,j) \in \mathbb{R}^{d'}$ which could generically apply to either $\langle w_i, e, w_j \rangle$ or $\langle w_i, r, w_j \rangle$ are given by

$$\widetilde{c}(i,j) = \widehat{g}\left(c^{(i)}, c^{(j)}, c^{\texttt{global}}\right) \tag{7}$$

where $\widehat{g} : (x,y,z) \in \mathbb{R}^{d'^3} \rightarrow \texttt{tanh}\left(\texttt{Linear}\left([x;y;z]\right)\right) \in \mathbb{R}^{d'}$.