Cross-stitching Text and Knowledge Graph Encoders for Distantly Supervised Relation Extraction

Qin Dai[*1] **Benjamin Heinzerling**[*2,1] **Kentaro Inui**[1,2]
[1]Tohoku University [2]RIKEN AIP
qin.dai.b8@tohoku.ac.jp benjamin.heinzerling@riken.jp
kentaro.inui@tohoku.ac.jp

Abstract

Bi-encoder architectures for distantly-supervised relation extraction aim to use complementary information from texts and knowledge graphs (KG). However, current architectures suffer drawbacks. They either do not share information between text and KG encoders at all, or, in case of models with KG-to-text attention, only share information in one direction. Here, we introduce cross-stitch bi-encoders, which allow bi-directional information sharing between text and KG encoders via a cross-stitch mechanism. Cross-stitching enables sharing and updating representations between the two encoders, with the degree of sharing controlled by cross-attention gates. Experiments on relation extraction benchmarks show that bi-directional sharing between encoders yields strong improvements.[1)]

1 Introduction

Identifying semantic relations between textual mentions of entities is a key task for information extraction systems. For example, consider the sentence:

(1) **Aspirin** is widely used for short-term treatment of **pain**, fever or colds.

Assuming an inventory of relations such as `may_treat` or `founded_by`, a relation extraction (RE) system should recognize the predicate in (1) as an instance of a `may_treat` relation and extract a knowledge graph (KG) triple like (ASPIRIN, `may_treat`, PAIN). RE systems are commonly trained on data obtained via Distant Supervision (DS) [1]: Given a KG triple, i.e., a pair of entities and a relation, one assumes that all sentences mentioning both entities express the relation and collects all such sentences as positive examples. DS allows collecting large amounts of training data, but its assumption is often violated:

(2) Nursing diagnoses acute **pain** related to **aspirin** use and variants in the radiotherapy group ...

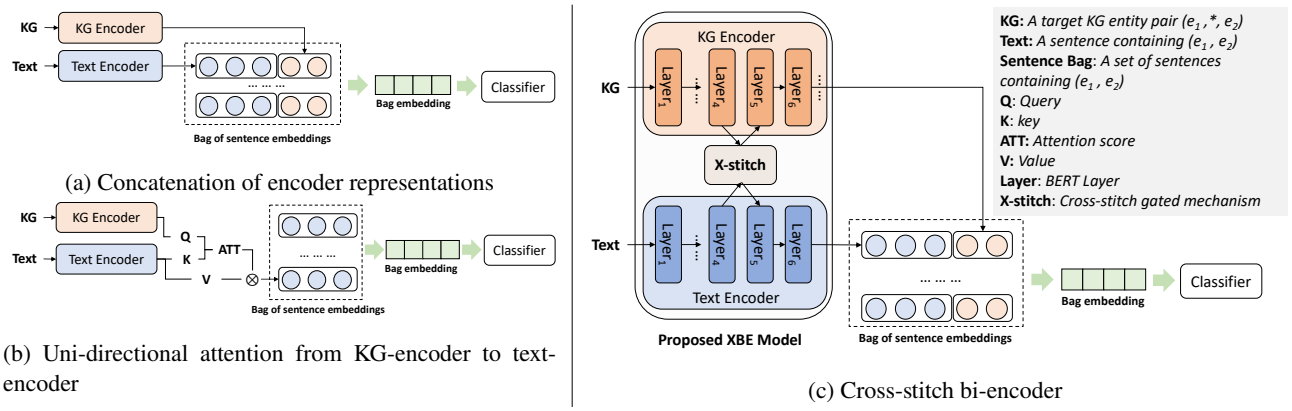
(3) **Elon Musk** and **SpaceX** engineers embark on a historic mission to return NASA astronauts to ...

Sentence (2) and (3) are false positive examples for `may_treat` and `founded_by` relation respectively, since they are not about a treatment and founding a company. We refer to false positive examples like (2) and (3) as **noisy** sentences.

A common approach for dealing with noisy sentences is to use the KG as a complementary source of information. Models taking this approach are typically implemented as bi-encoders, with one encoder for textual input and one encoder for KG input. They are trained to rely more on the text encoder when given informative sentences and more on the KG encoder when faced with noisy ones [2, 3, 4, 5, 6, 7]. However, current bi-encoder models suffer from drawbacks. Bi-encoders that encode text and KG separately and then concatenate each encoder's output, as illustrated in Figure 1a and proposed by [7], i.a., cannot share information between the text encoder and the KG encoder during encoding. In contrast, Bi-encoders whose text encoder can attend to the KG encoder's hidden states, as illustrated in Figure 1b and proposed by [2, 4, 3], i.a., do allow information to flow from the KG encoder to the text encoder, but not in the opposite direction.

Here, we propose a cross-stitch bi-encoder (XBE, Figure 1c) that addresses both of these drawbacks by enabling information sharing between the text encoder and KG encoder at arbitrary layers in both directions. Concretely, we equip a bi-encoder with a cross-stitch component [8]

1) Code and data: www.github.com/cl-tohoku/xbe
∗ Equal contribution



(a) Concatenation of encoder representations

(b) Uni-directional attention from KG-encoder to text-encoder

(c) Cross-stitch bi-encoder

Figure 1: Illustration of existing and proposed bi-encoder architectures for distantly-supervised relation extraction. Simple concatenation of representations (a) does not allow information sharing between text and KG encoders, while KG-to-text attention (b) only allows sharing in one direction. In contrast, our model (c) allows bi-directional information sharing between encoders during the encoding process.

to enable bi-directional information sharing and employ a gating mechanism based on cross-attention [9, 10] to dynamically control the amount of information shared between the text encoder and KG encoder. As we will show, allowing bi-directional information sharing during the encoding process, i.e., at intermediate layers, yields considerable performance improvements on two relation extraction benchmarks covering two different domains and achieves state of the art results on a widely used dataset.(§5.1)

2 Task Formulation

Given a corpus of entity-linked sentences and KG triples (e_1^k, r^k, e_2^k), distant supervision (DS) yields a bag of sentences $B^k = \{s_1^k, ..., s_n^k\}$ where each sentence s_i^k mentions both entities in the pair (e_1^k, e_2^k). Given the entity pair (e_1^k, e_2^k) and the sentence bag B^k, a DS-RE model is trained to predict the KG relation r^k.

3 Proposed Model

The cross-stitch bi-encoder (XBE) model is designed to enable bidirectional information sharing among its two encoders. As illustrated in Figure 1c, it consists of a text encoder, a KG encoder, and a cross-stitch component controlled by cross-attention. The following subsections describe these components.

3.1 Bi-Encoder

To obtain representations of inputs belonging to the two different modalities in DS-RE, we employ a bi-encoder architecture consisting of one encoder for textual inputs and one encoder for KG triples. While the cross-stitch component is agnostic to the type of encoder, we use pre-trained Transformer models [10] for both text and KG.

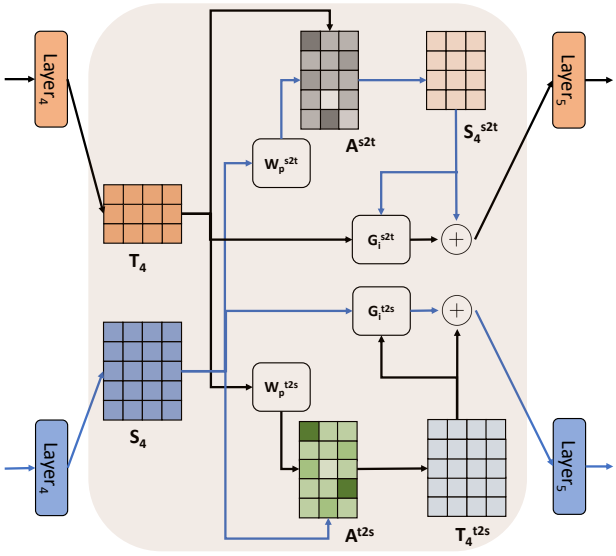


Figure 2: Illustration of the cross-stitch mechanism in combination with cross-attention. See §3.2 for notation.

3.2 Cross-stitch (X-stitch)

To enable bi-directional information sharing between the two encoders, we employ a cross-stitch[2)] mechanism based on [8]. The mechanism operates by mixing and updating intermediate representations of the bi-encoder. We dynamically control the amount of mixing via gates based on cross-attention (Figure 2). More formally, our cross-stitch variant operates as follows. Given a sen-

2) For brevity, we use **X-stitch** in tables and figures.

tence $s = (tok_1, ..., tok_N)$ and corresponding KG triple $t = (e_1, r, e_2)$, the text encoder generates sentence representations $S_i \in \mathbb{R}^{N \times d}$and the KG encoder triple representations $T_i \in \mathbb{R}^{3 \times d}$. We then compute cross-attentions A in two directions, triple-to-sentence ($t2s$) and sentence-to-triple ($s2t$), via Equations 1 and 2,

$$A^{t2s} = \text{softmax}_{\text{column}}((W_p^{t2s} \cdot T_i) \cdot S_i) \tag{1}$$

$$A^{s2t} = \text{softmax}_{\text{row}}(S_i \cdot (W_p^{s2t} \cdot T_i)^T) \tag{2}$$

where, $W_p^{s2t} \in \mathbb{R}^{d \times d}$ and $W_p^{t2s} \in \mathbb{R}^{d \times d}$ denote trainable linear transformations. The triple-to-sentence attention A^{t2s} represents the weight of the embedding of each token in triple t that will be used to update the sentence representation S_i:

$$T_i^{t2s} = W_{g2}^{t2s} \cdot ReLU(W_{g1}^{t2s} \cdot (A^{t2s} \cdot T_i^T)) \tag{3}$$

where $W_{g1}^{t2s} \in \mathbb{R}^{d' \times d}$ and $W_{g2}^{t2s} \in \mathbb{R}^{d \times d'}$ are trainable parameters. Next, a gating mechanism determines the degree to which the original textual representation S_i will contribute to the new hidden state of the text encoder:

$$\mathbf{G}_i^{t2s} = \sigma(T_i^{t2s}) \tag{4}$$

where, σ denotes the logistic sigmoid function. We then update the hidden state of the text encoder at layer i by interpolating its original hidden state S_i with the triple representation T_i^{t2s}:

$$S_i' = \mathbf{G}_i^{t2s} \cdot S_i + \lambda_t \cdot T_i^{t2s} \tag{5}$$

Information sharing in the sentence-to-triple direction is performed analogously:

$$S_i^{s2t} = W_{g2}^{s2t} \cdot ReLU(W_{g1}^{s2t} \cdot ((A^{s2t})^T \cdot S_i)) \tag{6}$$

$$\mathbf{G}_i^{s2t} = \sigma(S_i^{s2t}) \tag{7}$$

$$T_i' = \mathbf{G}_i^{s2t} \cdot T_i + \lambda_s \cdot S_i^{s2t} \tag{8}$$

where λ_t and λ_s are weight hyperparameters. Having devised a general architecture for text-KG bi-encoders, we now turn to implementing this architecture for distantly supervised relation extraction.

4 XBE for Relation Extraction

In distantly supervised relation extraction, the automatically collected data consists of a set of sentence bags $\{B^1, ..., B^n\}$ and set of corresponding KG triples $\{(e_1^1, r^1, e_2^1), ..., (e_1^n, r^n, e_2^n)\}$. To create training instances, we mask the relation in the KG triples $\{(e_1^1, [\text{M}], e_2^1), ..., (e_1^n, [\text{M}], e_2^n)\}$ and provide these masked triples as input to the KG encoder, while the text encoder receives one sentence from the corresponding sentence bag. If the sentence bag contains k sentences, we pair each sentence with the same KG triple and run the bi-encoder for each pairing, i.e., k times, to obtain a sentence bag representation. During training, the loss of the model is calculated via Equations 9, 10 and 11,

$$L = L_{RE} + w \cdot L_{KG} \tag{9}$$

$$L_{RE} = -\sum_{k=1}^{n} \sum_{i=1}^{|B^k|} \log P(r^k | [\mathbf{s}_i^k; \mathbf{r}_{ht}; x_{e_1^k}; x_{e_2^k}]) \tag{10}$$

$$L_{KG} = -\sum_{k=1}^{n} \log g((e_1^k, [\text{M}], e_2^k)) \tag{11}$$

where $w \in (0, 1]$ is a weight hyperparameter, $P(x)$ is the predicted probability of the target relation over a set of predefined relations, ; denotes vector concatenation, $\mathbf{r}_{ht}$ is an additional KG feature vector obtained from a pre-trained KG completion model such as TransE [11], x_{e^k} is entity embedding from the KG encoder, L_{KG} is the loss of KG relation prediction and $g(x)$ outputs the predicted probability of the masked KG token based on its embedding from the KG encoder. During inference, we follow [7] and use the mean of sentence embeddings as the bag embedding:

$$P(r^k | B^k) = (\sum_{i=1}^{|B^k|} P(r^k | [\mathbf{s}_i^k; \mathbf{r}_{ht}; x_{e_1^k}; x_{e_2^k}]))/|B^k| \tag{12}$$

5 Experiments

Please see Appendix (§A.1) for details on Data and Settings and (§A.2) on Baseline Models.

5.1 Results

The Precision-Recall (PR) curves of each model on Medline21 and NYT10 datasets are shown in Figure 3 and Figure 4, respectively. We make two main observations: (1) Among the compared models, BRE+KA and BRE+CE, are strong baselines because they significantly outperform

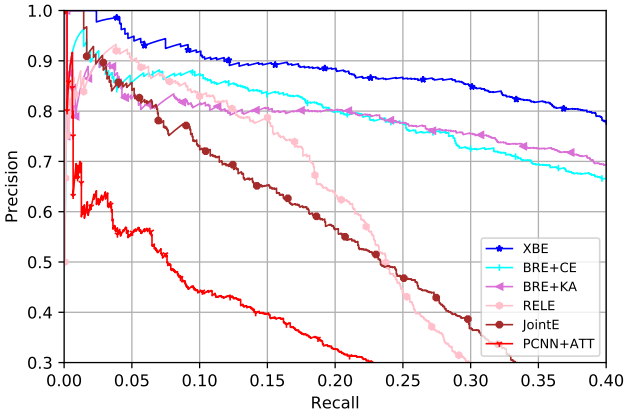


Figure 3: PR curves on Medline21.

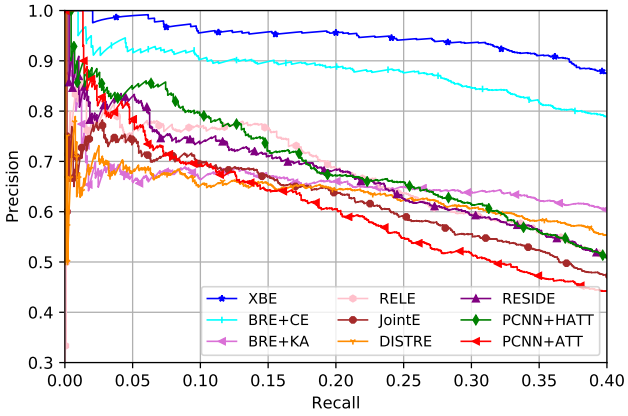


Figure 4: PR curves on NYT10.

	Medline21					NYT10						
Model	AUC	P@0.3k	P@0.5k	P@1k	P@2k	AUC	P@0.1k	P@0.2k	P@0.3k	P@0.5k	P@1k	P@2k
PCNN+ATT	17.8*	48.3*	43.2*	34.3*	25.2*	34.1†	73.0†	68.0†	67.0†	63.6†	53.3†	40.0†
PCNN+HATT	-	-	-	-	-	42.0‡	81.0‡	79.5‡	75.7‡	68.0‡	58.6‡	42.1‡
RESIDE	-	-	-	-	-	41.5†	81.8†	75.4†	74.3†	69.7†	59.3†	45.0†
DISTRE	-	-	-	-	-	42.2†	68.0†	67.0†	65.3†	65.0†	60.2†	47.9†
JointE	26.3*	70.0*	61.4*	46.4*	30.0*	38.5*	74.0*	71.5*	69.0*	65.4*	55.9*	43.6*
RELE	25.6*	78.7*	66.8*	44.7*	27.5*	40.5*	79.0*	77.0*	77.0*	71.2*	59.3*	44.7*
BRE+KA	50.3*	79.7*	79.2*	70.3*	51.2*	48.8*	68.0*	68.0*	67.0*	66.0*	63.7*	52.4*
BRE+CE	55.3*	84.0*	79.4*	67.7*	53.8*	63.2‡	92.0‡	92.0‡	90.0‡	88.0‡	78.7‡	58.7‡
XBE	**61.9**	**89.3**	**86.4**	**76.1**	**56.1**	**70.5**	**99.0**	**96.0**	**95.6**	**94.4**	**85.8**	**63.2**

Table 1: P@N and AUC on Medline21 and NYT10 datasets (k=1000), where †represents that these results are quoted from [12], ‡indicates the results using the pre-trained model, ⋆ indicates the results are obtained by re-running corresponding codes and ∗ indicates using the OpenNRE [13] implementation.

other state-of-the-art models especially when the recall is greater than 0.25, demonstrating the benefit of combining a pre-trained language model (here: BERT) and a KG for DS-RE. (2) The proposed XBE model outperforms all baselines and achieves the highest precision over the entire recall range on both datasets. Table 1 further presents more detailed results in terms of AUC and P@N, which shows improved performance of XBE in all testing metrics. In particular, XBE achieves a new state-of-the-art on the commonly used NYT10 dataset, indicating that the proposed model can make better use of the combination of KG and text for DS-RE. Please see Appendix for Ablation Study(§A.3).

6 Conclusions and Future Work

We proposed a cross-stitch bi-encoder architecture, XBE, to leverage the complementary relation between KG and text for distantly supervised relation extraction. Experimental results on both Medline21 and NYT10 datasets prove the robustness of our model because the proposed model achieves significant and consistent improvement as compared with strong baselines and achieve a new state-of-the-art result on the widely used NYT10 dataset. Possible future work includes a more thorough investigation of how communication between KG encoder and text encoder influences the performance, as well as a more complex KG encoder that can not only handle relation triples, but arbitrary KG subgraphs, which could have applications in, e.g., multi-hop relation extraction.

Acknowledgements

This work was supported by JST CREST Grant Number JPMJCR20D2 and JSPS KAKENHI Grant Number 21K17814.

References

[1] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In **Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2**, pp. 1003–1011. Association for Computational Linguistics, 2009.

[2] Xu Han, Zhiyuan Liu, and Maosong Sun. Neural knowledge acquisition via mutual attention between knowledge graph and text. In **Thirty-Second AAAI Conference on Artificial Intelligence**, 2018.

[3] Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. 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)**, pp. 3016–3025, 2019.

[4] Linmei Hu, Luhao Zhang, Chuan Shi, Liqiang Nie, Weili Guan, and Cheng Yang. Improving distantly-supervised relation extraction with joint label embedding. 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)**, pp. 3821–3829, 2019.

[5] Qin Dai, Naoya Inoue, Paul Reisert, Takahashi Ryo, and Kentaro Inui. Incorporating chains of reasoning over knowledge graph for distantly supervised biomedical knowledge acquisition. In **Proceedings of the 33nd Pacific Asia Conference on Language, Information and Computation (PACLIC33)**, pp. 19–28, Hakodate, Japan, 2019. Waseda Institute for the Study of Language and Information.

[6] Qin Dai, Naoya Inoue, Ryo Takahashi, and Kentaro Inui. Two training strategies for improving relation extraction over universal graph. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 3673–3684, 2021.

[7] Zikun Hu, Yixin Cao, Lifu Huang, and Tat-Seng Chua. How knowledge graph and attention help? a qualitative analysis into bag-level relation extraction. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 4662–4671, 2021.

[8] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In **Proceedings of the IEEE conference on computer vision and pattern recognition**, pp. 3994–4003, 2016.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**, 2015.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. **Advances in neural information processing systems**, Vol. 30, , 2017.

[11] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In **Advances in neural information processing systems**, pp. 2787–2795, 2013.

[12] Christoph Alt, Marc Hübner, and Leonhard Hennig. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. **arXiv preprint arXiv:1906.08646**, 2019.

[13] Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. OpenNRE: An open and extensible toolkit for neural relation extraction. In **Proceedings of EMNLP-IJCNLP: System Demonstrations**, pp. 169–174, 2019.

[14] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In **Joint European Conference on Machine Learning and Knowledge Discovery in Databases**, pp. 148–163. Springer, 2010.

[15] 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.

[16] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. **Bioinformatics**, Vol. 36, No. 4, pp. 1234–1240, 2020.

[17] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, Vol. 1, pp. 2124–2133, 2016.

[18] Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. Hierarchical relation extraction with coarse-to-fine grained attention. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 2236–2245, 2018.

[19] Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. Reside: Improving distantly-supervised neural relation extraction using side information. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 1257–1266, 2018.

	#R	#EP	#Related EP	#Sentence
Medline21	40	100,549 / 21,081	10,936 / 1,804	165,692 / 28,912
NYT10	53	281,270 / 96,678	18,252 / 1,950	522,611 / 172,448

Table 2: Statistics of datasets in this work, where **R** and **EP** stand for the target Relation and Entity Pair, $\#_1/\#_2$ represent the number of training and testing data respectively.

Hyperparameter	Medline21	NYT10
learning rate	3e-5	3e-5
weight decay rate	1e-5	1e-5
Adam epsilon	1-e8	1-e8
warmup steps	500	500
batch size	100	80
w	1.0	0.6
λ_t	1.0	1.0
λ_s	1-e4	1-e4
maximum epochs	15	10

Table 3: Hyperparameters used in our proposed XBE model. The experiments (Medline21) are conducted on Nvidia Titan X(Pascal) GPU, and the experiments (NYT10) are conducted on a NVIDIA GeForce GTX 1080 TI GPU.

A Appendix

A.1 Data and Settings

We evaluate our model on the biomedical dataset introduced by [6] (hereafter: Medline21) and the NYT10 dataset [14] Statistics for both datasets are summarized in Table 2. Medline21 dataset contains 582, 686 KG triples and NYT10 does 335, 350 triples.

As done by [7], the text encoder (§3) for experiments on NYT10 is initialized with the pre-trained weights from the `bert-base-uncased` variant of BERT [15]. The text encoder for Medline21 is initialized with BioBERT [16] and the KG encoder (§3) is pre-trained using each dataset's corresponding KG triples, as mentioned above. Hyperparameters used in our Model are listed in Table 3.

A.2 Baseline Models

To demonstrate the effectiveness of the proposed model, we compare to the following baselines. Baselines were selected because they are the closest models in terms of integrating KG with text for DS-RE and/or because they achieve competitive or state-of-the-art performance on the datasets used in our evaluation: **JointE** [2], which is a joint model for KG embedding and RE, where the KG embedding is utilized for attention calculation over a sentence bag, as shown in Figure 1b, **RELE** [4], which extends the JointE via entity definitions, **BRE+KA** [7], which is a version of the JointE model that integrates BERT, and **BRE+CE** [7], which is a BERT and KG embedding based model, where BERT output and the KG triple embedding are concatenated as a feature vector for DS-RE, as shown in Figure 1a.

In addition to the models above, we select the following baselines for further comparison: **PCNN+ATT** [17], **PCNN+HATT** [18], **RESIDE** [19] and **DISTRE** [12].

	Medline21		NYT10	
Model	AUC	P@2k	AUC	P@2k
XBE	**61.9**	**56.1**	**70.5**	**63.2**
- X-stitch	58.7	53.3	68.3	61.3
- KG enc.	55.7	53.8	61.5	56.9
- text enc.	39.8	41.1	55.9	55.1

Table 4: Performance comparison of XBE with different ablated components (non-cumulative) on Medline21 and NYT10 datasets (k=1000).

A.3 Ablation Study

We ablate the three main model components in order to assess the contribution to overall performance. Results are shown in Table 4, where "- X-stitch" is the model without the cross-stitch mechanism, "- KG enc." denotes removing the KG encoder, and "- text enc." removing the text encoder. We observe that performance drops for all ablations, indicating that each component is important for the model when performing DS-RE. While the impact of ablating the text encoder is by far the largest, removing the cross-stitch component or the KG encoder results in performance that is comparable to the performance of the strongest baseline, BRE+CE, on both datasets. This suggests that these two components, i.e., the KG encoder and the cross-stitch mechanism allowing sharing of information between the text and KG encoder, are what enables our model to improve over BRE+CE.