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

Qin Dai[*1], **Benjamin Heinzerling**[*1,2], **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 are designed to make use of the complementary information found in text and knowledge graphs (KG). However, current architectures suffer from two drawbacks. They either do not allow any sharing between the text encoder and the KG encoder 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 full interaction between the text encoder and the KG encoder via a cross-stitch mechanism. The cross-stitch mechanism allows sharing and updating representations between the two encoders at any layer, with the amount of sharing being dynamically controlled via cross-attention-based gates. Experimental results on two relation extraction benchmarks from two different domains show that enabling full interaction between the two encoders yields strong improvements.

https://github.com/cl-tohoku/xbe

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, Mintz et al., 2009): 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) The tumor was remarkably large in size, and **pain** unrelieved by **aspirin**.

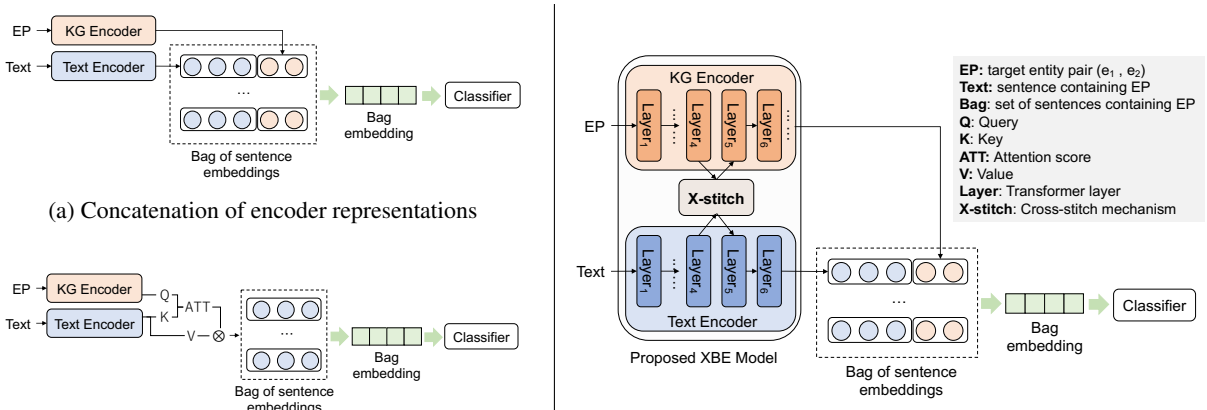
(3) **Elon Musk** fired some **SpaceX** employees who were talking smack about ...

Sentence (2) is a false positive example of a `may_treat` relation since it describes a failed treatment. Sentence (3) is, strictly speaking, a false positive of a `founded_by` relation since this sentence is not about founding companies, but can also be seen as indirect evidence, since founders are often in a position that allows them to fire employees. We refer to false positive and indirectly relevant 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 (Weston et al., 2013; Han et al., 2018a; Zhang et al., 2019; Hu et al., 2019; Dai et al., 2019, 2021; Hu et al., 2021). 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 Hu et al. (2021), 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 Han et al. (2018a); Hu et al. (2019); Zhang et al. (2019), i.a., do allow information to flow from the KG encoder to the text encoder, but not in the opposite direction.

∗ Equal contribution

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6947–6958
December 7-11, 2022



(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.

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. Intuitively, such a "full interaction" between the two encoders is desirable because it is not *a priori* clear at which point in the encoding process an encoder's representation is best-suited for sharing with the other encoder. Concretely, we equip a bi-encoder with a cross-stitch component (Misra et al., 2016) to enable bi-directional information sharing and employ a gating mechanism based on cross-attention (Bahdanau et al., 2015; Vaswani et al., 2017) 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.

In summary, our contributions are:

- A bi-encoder architecture that enables full interaction between its encoders: both encoders can share and update information at any layer (§3);
- An implementation of the proposed architecture for distantly-supervised relation extraction (§4);
- Improvement of performance on two relation extraction benchmarks covering two different domains and achievement of state of the art results on a widely used dataset.(§5.4);
- Ablations showing the importance of the components of the proposed architecture (§5.5).

2 Terminology and Notation

Throughout this work we use terminology and notation as follows. We assume access to a domain-specific knowledge graph (KG) which contains fact triples $\mathcal{O} = \{(e_1, r, e_2), ...\}$ consisting of entities $e_1, e_2 \in \mathcal{E}$ and a relation $r \in \mathcal{R}$ that holds between them. The set of entities $\mathcal{E}$ and the inventory of relations $\mathcal{R}$ are closed and finite.

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 Cross-stitch Bi-Encoder (XBE)

The cross-stitch bi-encoder 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 (Vaswani et al., 2017) for both text and KG.

The **Text Encoder** takes a sentence s_i^k containing a sequence of N tokens $(tok_1, ..., tok_N)$ as input and produces L_T layers of d_T-dimensional contextualized representations $S_i \in \mathbb{R}^{N \times d_T}$, $1 \leqslant i \leqslant L_T$. We construct a fixed-length representation of the sentence s_i^k mentioning the entity pair (e_1^k, e_2^k) by concatenating the embeddings of the head and tail entities h_{e1} and h_{e2} obtained from the last layer S_{L_T} via the method described in Peng et al. (2020), as well as the mean- and max-pooled token representations h_{mean} and h_{max} obtained from pooling over the last encoder layer S_{L_T}. That is, the final representation of the input sentence s_i^k is $\mathbf{s}_i^k = [h_{e1}; h_{e2}; h_{mean}; h_{max}]$, where ; denotes vector concatenation.

The **KG Encoder** takes a KG triple (e_1, r, e_2) as input and generates T_K layers of d_K-dimensional contextualized representations $T_i \in \mathbb{R}^{3 \times d_K}$, $1 \leqslant i \leqslant L_K$. Then $x_{e_1} \in \mathbb{R}^{d_K}$, $x_r \in \mathbb{R}^{d_K}$ and $x_{e_2} \in \mathbb{R}^{d_K}$ from the last layer T_{L_K} are used as the embeddings of the head entity e_1, relation r and tail entity e_2 respectively. The KG encoder's vocabulary $\mathcal{V}$ is formed by the union of all entities $\mathcal{E}$ and relations $\mathcal{R}$, as well as a mask token [M], i.e., $\mathcal{V} = \mathcal{R} \cup \mathcal{E} \cup \{[\text{M}]\}$. For simplicity we assume that the text and KG encoder representations have the same dimensionality d, that is, we set $d = d_K = d_L$, although this is not required by the model architecture.

3.2 Cross-stitch (X-stitch)

To enable bi-directional information sharing between the two encoders, we employ a cross-stitch[1] mechanism based on Misra et al. (2016). 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 sentence $s = (tok_1, ..., tok_N)$ 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,

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

[1]For brevity, we use *X-stitch* in tables and figures.

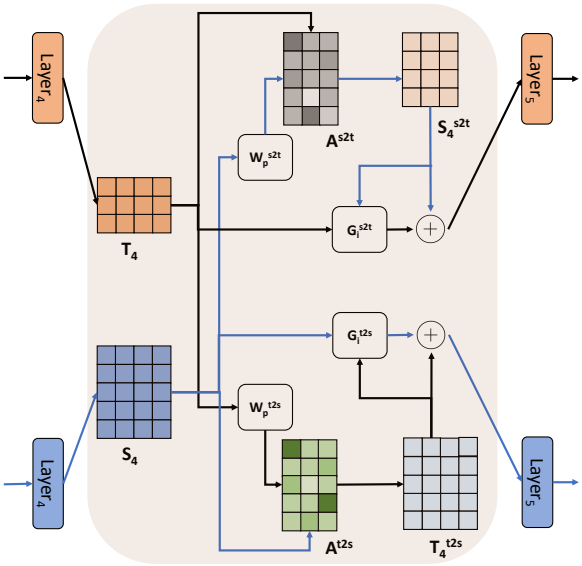


Figure 2: Illustration of the cross-stitch mechanism in combination with cross-attention. See §3.2 for notation.

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

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:

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

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:

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

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}:

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

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

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

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

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

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,

$$L = L_{RE} + w \cdot L_{KG} \quad (9)$$

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

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

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, $\mathbf{r}_{ht}$ is an additional KG feature vector obtained from a pre-trained KG completion model such as TransE (Bordes et al., 2013), L_{KG} is the loss of KG relation prediction and $g(x)$ outputs the predicted probability of the masked token over the vocabulary $\mathcal{V}$ based on the embedding $x_{[\text{M}]}$ from the KG encoder.

During inference, we follow Hu et al. (2021) and use the mean of sentence embeddings as the bag embedding:

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

As our bi-encoder consists of two transformer-based encoders, we make use of pre-training for each modality. For the text encoder, we employ an off-the-shelf model, as detailed in the next section. The KG encoder is pre-trained on a set of KG triples via a relation prediction task. Specifically, given a relation masked triple $(e_1, [\text{M}], e_2)$, the KG encoder is asked to predict the masked symbolic token and pre-trained via the loss given by Equation 11.

5 Experiments

5.1 Data

We evaluate our model on the biomedical dataset introduced by Dai et al. (2021) (hereafter: Medline21) and the NYT10 dataset (Riedel et al., 2010) Statistics for both datasets are summarized in Table 1.

Medline21. This dataset was created by aligning the biomedical knowledge graph UMLS[2] with the Medline corpus, a collection of biomedical abstracts. Both resources are published by the U.S. National Library of Medicine[3]. A state-of-the-art UMLS Named Entity Recognizer, ScispaCy (Neumann et al., 2019), is applied to identify UMLS entity mentions in the Medline corpus. The sentences until the year 2008 are used for training and the ones from the year 2009 $\sim$ 2018 are used for testing. Following (Han et al., 2018a), Dai et al. (2021) also provided a subset of UMLS in the dataset, which consists of $582,686$ KG triples. We use the set of triples to train the KG encoder.

NYT10. This dataset was created by aligning Freebase relational facts with the New York Times Corpus. Sentences from the year 2005 $\sim$ 2006 are used for training and the sentences from 2007 are used for testing. The NYT10 dataset has been used widely for relation extraction (Lin et al., 2016; Ji et al., 2017; Du et al., 2018; Jat et al., 2018; Han et al., 2018a,b; Vashishth et al., 2018; Ye and Ling, 2019; Hu et al., 2019; Alt et al., 2019; Sun et al., 2019; Li et al., 2020; Hu et al., 2021; Dai et al., 2021). In order to leverage a KG for DS-RE on NYT10, Han et al. (2018a) extended the dataset with FB60K, which is a KG containing $335,350$ triples. Following (Hu et al., 2019; Han et al., 2018a; Hu et al., 2021), we use FB60K to train the KG encoder for DS-RE.

[2] https://www.nlm.nih.gov/research/umls/
[3] https://www.nlm.nih.gov/

	#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 1: 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.

5.2 Settings

Following the conventional settings of DS-RE (see, e.g., Lin et al., 2016), we conduct a held-out evaluation, in which models are evaluated by comparing the fact triple identified from a bag of sentences S_r with the corresponding KG triple. Further following evaluation methods of previous work, we draw Precision-Recall curves and report the Area Under Curve (AUC), as well as Precision@N (P@N) scores, which give the percentage of correct triples among the top N ranked predictions. In addition, as done by Hu et al. (2021), the text encoder (§3) for experiments on NYT10 is initialized with the pre-trained weights from the `bert-base-uncased` variant of BERT (Devlin et al., 2018). The text encoder for Medline21 is initialized with BioBERT (Lee et al., 2020) and the KG encoder (§3) is pre-trained using each dataset's corresponding KG, as mentioned above.

5.3 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** (Han et al., 2018a): 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** (Hu et al., 2019): A multi-layer attention-based model, which makes use of KG embeddings and entity descriptions for DS-RE.
- **BRE+KA** (Hu et al., 2021): A version of the JointE model that integrates BERT.
- **BRE+CE** (Hu et al., 2021): 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.

To collect AUC results and draw Precision-Recall curves, we use pre-trained models where possible or carefully run published implementations using suggested hyperparameters[4] from the original papers if no pre-trained model is publicly available. See the supplementary material for training details.

In addition to the models above, we select the following baselines for further comparison.

- **PCNN+ATT** (Lin et al., 2016) A CNN-based model with a relation embedding attention mechanism.
- **PCNN+HATT** (Han et al., 2018b) A CNN-based model with a relation hierarchy attention mechanism.
- **RESIDE** (Vashishth et al., 2018) A Bi-GRU-based model which makes use of relevant side information (e.g., syntactic information), which is encoded via a Graph Convolution Network.
- **DISTRE** (Alt et al., 2019) A Generative Pre-trained Transformer model with a relation embedding attention mechanism.

5.4 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 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 2 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.

[4]For hyperparameter selection on Medline21, we use a 30% random split of the training set.

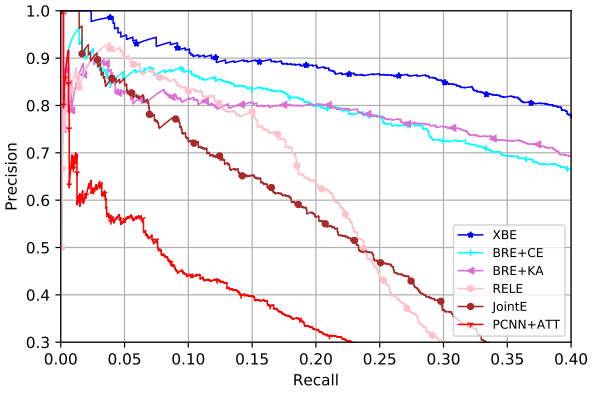


Figure 3: PR curves on Medline21.

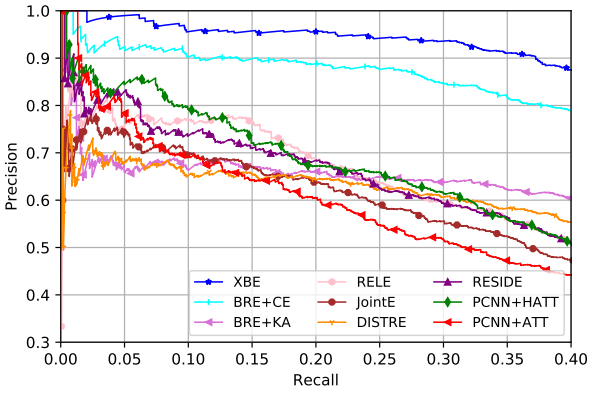


Figure 4: PR curves on NYT10.

	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 2: P@N and AUC on Medline21 and NYT10 datasets (k=1000), where †represents that these results are quoted from (Alt et al., 2019), ‡indicates the results using the pre-trained model, ⋆ indicates the results are obtained by re-running corresponding codes and ∗ indicates using the OpenNRE (Han et al., 2019) implementation.

Since the underlying resources, namely the pre-trained language model and the KG are the same as those used by the best baseline models, we take this strong performance as evidence that the proposed model can make better use of the combination of KG and text. This in turn, we hypothesize, is due to the fact that our proposed model can realize encoder layer level communication between KG and text representations. In the next section we conduct an ablation study to verify this hypothesis.

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

5.5 Ablation Study

We first ablate the three main model components in order to assess the contribution to overall performance. Results are shown in Table 3, 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.

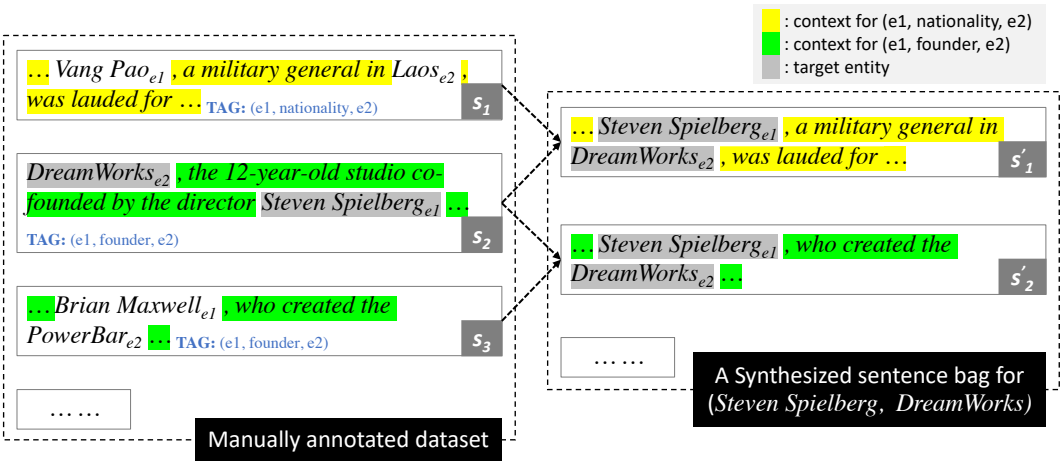


Figure 5: Process of creating a synthesized sentence bag, in which a valid sentence (e.g., s'_2) is created by the combination of a target entity pair (e.g., (Steven Spielberg, DreamWorks)) and the context representing their relation (e.g., *founder*), while a noisy one (e.g., s'_1) is done by the combination of the target entity pair and a random context representing different relation (e.g., *nationality*).

	Medline21		NYT10	
Model	AUC	P@2k	AUC	P@2k
XBE	**61.9**	**56.1**	**70.5**	**63.2**
- Pre-KG enc.	55.8	52.8	63.7	60.3
- Joint-KG enc.	49.7	49.6	63.0	59.1

Table 4: Performance comparison of XBE trained under different conditions (non-cumulative) on Medline21 and NYT10 datasets (k=1000).

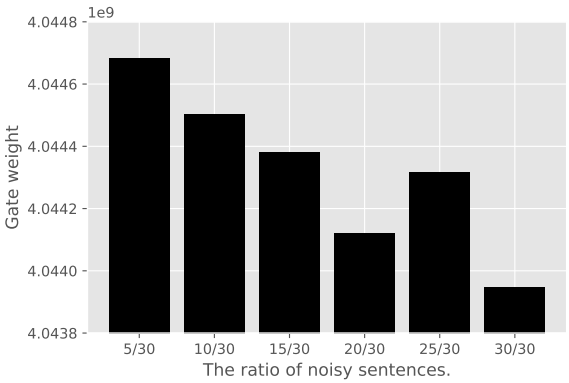


Figure 6: The sum of gate weights w.r.t different noise ratio. The x axis denotes the noise ratio of a set of synthesized data, where $\#_1/\#_2$ represents the number of noisy and all sentences in a synthesized sentence bag respectively. The y axis denotes the sum of gate weights over an entire set.

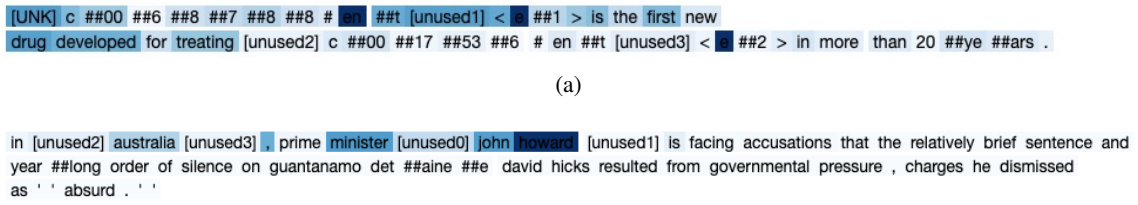
As described in §4, we pre-train the KG encoder via a relation prediction task before fine-tuning the XBE model end-to-end on a DS-RE dataset. In order to measure the effect of KG encoder pre-training, we compare with a setup in which the KG encoder is not pre-trained but initialized randomly instead. In addition, since our proposed XBE model facilitates joint training of the KG encoder and text encoder, we also compare to a setting in which the pre-trained KG encoder is frozen, i.e., not updated during training on the two DS-RE datasets. The results of these KG-encoder ablations are shown in Table 4, where "- Pre-KG enc." denotes the random initialization of the KG encoder and "- Joint-KG enc." is the model with a pre-trained, frozen KG encoder. We observe that performance decreases both without pre-training of the KG encoder and when we freeze the KG encoder while fine-tuning XBE. That is, performance gains not only stem from employing a pre-trained KG encoder but also from the effective joint training of both the KG encoder and the text encoder.

5.6 Cross-stitch Gate Weights vs. Noise

In order to analyze how the XBE model dynamically controls information flow between the encoders, we construct several sets of synthesized sentence bags differing in the proportion of noisy sentences they contain, similarly to Hu et al. (2021). Specifically, given a target entity pair (e_1, e_2) we create a synthesized sentence bag in which each valid sentence is created by the combination of the entity pair and the context that expresses their relation, and each noisy one by randomly selecting a context representing a different relation. This process is illustrated in Figure 5. We use the NYT10m dataset (Gao et al., 2021), which is a

Bag	Sentence	Target Relation	XBE	XBE - X-stitch	BRE+CE
B1	... is released from prison into an unrecognizable 1980s **Paris**, and ... in which the New Wave aesthetic reaches some sort of terminal point, as **Jean-Pierre** ...	/people/person/ place_lived	✓	✗	✗
B2	... an online advertising technology company in **san francisco** called **zedo**, ...	/business/company/ place_founded	✓	✗	✗
B3	... CagA promoted the underglycosylation of IgA1 , which at least partly attributed to the downregulation of **C3812673#ent** (C1GALT1) and its **C1332924#ent** ...	gene_product_ encoded_by_gene	✓	✗	✗
B4	Although existing recommendations for the care of patients with **C0017205#ent** in effect , unique characteristics of **C3272698#ent** require additional investigation and monitoring .	may_be_treated_by	✓	✗	✗

Table 5: Qualitative results. Each bag contains one sentence, ✓(or ✗) represents the correct (or incorrect) prediction of the target relation.



(a)

(b)

Figure 7: Examples of A^{s2t} in the cross-stitch component, the comparative contribution of each token is visualized by the blue level, where the higher the blue the bigger the contribution. Figure 7a (above) shows the A^{s2t} visualization for predicting the masked token in $(e_1, [\text{M}], e_2)$, where the ground truth relation is *may_treat*, and Figure 7b (below) does A^{s2t} visualization for predicting the KG relation */people/person/nationality*.

manually annotated version of the NYT10 test set, as data source and create 6 sets of synthesized sentence bags with noise settings varying from $5/30$ to $30/30$, where $5/30$ ($30/30$) denotes that in the set, each bag has 30 sentences and contains 5 (30) noisy sentences. Each set contains about 4k entity pairs and each entity pair has 30 sentences, for a total of about 130k sentences.

We train one XBE model on each of the six sets with varying noise proportions and observe the gate weights of the cross-stitch mechanism, $\mathbf{G}_i^{t2s}$ in Equation 4, which control the amount of information that flows into the next layer of text encoder. We show the weights with respect to different noise ratios in Figure 6. From the Figure 6, we can observe that the gate weights (i.e., $\mathbf{G}_i^{t2s}$) tend to decrease as the noise ratio increases, indicating that the proposed cross-stitch mechanism of XBE effectively filters out noisy sentences and thereby aids the text encoder in extracting effective features. This observation is a possible explanation for the performance gain from the cross-stitch mechanism found in the ablation study (Table 3).

5.7 Qualitative Examples

We provide a few qualitative examples intended to demonstrate how the proposed cross-stitch mechanism can impact the performance of DS-RE, which are shown in Table 5. We can observe that the cross-stitch mechanism appears to facilitate DS-RE especially when a sentence bag is noisy. For instance, although the bag B1 fails to describe the */people/person/place_lived* relation, the proposed model can utilize useful information from KG through cross-stitch and thus correctly predicts the relation. Similarly, the the model can correctly a identify *may_be_treated_by* relation from the bag B4, which does not explicitly describe the target relation. Please see the supplementary material for further results.

We also visualize cross-attention weights A^{s2t}, which indicate the attention values over textual tokens used by the KG encoder to construct hidden representations. As shown in Figure 7a, for the representation of the KG relation token *may_treat*, the cross-stitch mechanism assigns higher atten-

tion score on informative tokens such as "drug" and "treating" than the irrelevant ones from"more than 20 years". Similarly, as shown in Figure 7b, in order to encode */people/person/nationality*, the cross-stitch mechanism focuses on the token "minister", which implicitly conveys the meaning of nationality, than irrelevant tokens such as "facing".

6 Additional Related Work

In this section we discuss related work besides the approaches already mentioned in the introduction. To improve the performance of a DS-RE model, recently, researchers introduce various attention mechanisms. Lin et al. (2016) propose a relation vector based attention mechanism. Jat et al. (2018); Du et al. (2018) propose multi-level (e.g., word-level and sentence-level) structured attention mechanism. Ye and Ling (2019) apply both intra-bag and inter-bag attention for DS-RE. Han et al. (2018b) propose a relation hierarchy based attention mechanism. Jia et al. (2019) propose an attention regularization framework for DS-RE. To handle the one-instance sentence bags, Li et al. (2020) propose a new selective gated mechanism.

Ji et al. (2017) apply entity descriptions generated from Freebase and Wikipedia as extra evidences, Lin et al. (2017) utilize multilingual text as extra evidences and Vashishth et al. (2018) use multiple side information including entity types, dependency and relation alias information for DS-RE. Alt et al. (2019) utilize pre-trained language model for DS-RE. Sun et al. (2019) apply relational table extracted from Web as extra evidences for DS-RE. Zeng et al. (2017) apply two-hop KG paths for DS-RE. Dai et al. (2021) introduce multi-hop paths over a KG-text joint graph for DS-RE.

KG has been proved to be effective for DS-RE. Han et al. (2018a) propose a joint model that adopts a KG embeddings based attention mechanism. Dai et al. (2019) extend the framework of Han et al. (2018a) by introducing multiple KG paths as extra evidences for DS-RE. Hu et al. (2019) propose a multi-layer attention-based framework to utilize both KG and textual signals for DS-RE. Based on the extensive analysis about the effect of KG and attention mechanism on DS-RE, Hu et al. (2021) proposed a straightforward but strong model and achieve a significant performance gain. However these methods mostly employ shallow integration of KG and text such as representations concatenation and KG embedding based attention mechanism. To fully take advantage of KG for DS-RE, in this paper, we propose a novel model to realize deep encoder level integration of KG and text.

7 Limitations

We focus only on one particular NLP task (i.e., DS-RE) to explore the effective way to jointly encoding KG and text, and thus further work is required to determine to what extend the proposed XBE can be generalized into multiple NLP tasks. Therefore, our work carries the limitation that the performance gain in DS-RE does not guarantee that it is effective in other NLP tasks such as Knowledge Graph Completion and Question Answering, where the combination of KG and text is needed. For this reason, we empathize the importance of multi-tasking for exploring such research question. In addition, we only utilize monolingual datasets to conduct evaluation and thus further work is required to investigate the effectiveness of the proposed model on multi-lingual datasets.

8 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. We are grateful to the anonymous reviewers for their constructive comments.

References

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

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

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

Qin Dai, Naoya Inoue, Paul Reisert, Takahashi Ryo, and Kentaro Inui. 2019. 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)*, pages 19–28, Hakodate, Japan. Waseda Institute for the Study of Language and Information.

Qin Dai, Naoya Inoue, Ryo Takahashi, and Kentaro Inui. 2021. 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*, pages 3673–3684.

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

Jinhua Du, Jingguang Han, Andy Way, and Dadong Wan. 2018. Multi-level structured self-attentions for distantly supervised relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2216–2225.

Tianyu Gao, Xu Han, Keyue Qiu, Yuzhuo Bai, Zhiyu Xie, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. Manual evaluation matters: reviewing test protocols of distantly supervised relation extraction. *arXiv preprint arXiv:2105.09543*.

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

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

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

Linmei Hu, Luhao Zhang, Chuan Shi, Liqiang Nie, Weili Guan, and Cheng Yang. 2019. 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)*, pages 3821–3829.

Zikun Hu, Yixin Cao, Lifu Huang, and Tat-Seng Chua. 2021. 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)*, pages 4662–4671.

Sharmistha Jat, Siddhesh Khandelwal, and Partha Talukdar. 2018. Improving distantly supervised relation extraction using word and entity based attention. *arXiv preprint arXiv:1804.06987*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. Arnor: attention regularization based noise reduction for distant supervision relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1399–1408.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Yang Li, Guodong Long, Tao Shen, Tianyi Zhou, Lina Yao, Huan Huo, and Jing Jiang. 2020. Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8269–8276.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Neural relation extraction with multi-lingual attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34–43.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. 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)*, volume 1, pages 2124–2133.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. 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*, pages 1003–1011. Association for Computational Linguistics.

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

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672.

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

Huan Sun et al. 2019. Leveraging 2-hop distant supervision from table entity pairs for relation extraction. 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)*, pages 410–420.

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

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

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371. Association for Computational Linguistics.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Distant supervision relation extraction with intra-bag and inter-bag attentions. 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)*, pages 2810–2819.

Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Incorporating relation paths in neural relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1768–1777.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. 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)*, pages 3016–3025.

A Appendix

A.1 Cross-stitch Layer Selection

Since the first few layers of BERT are the basis for the high level semantic task (Jawahar et al., 2019), we place the cross-stitch in the layers $1 \sim 6$, and conduct a layer by layer analysis to find the best fitting layers in a development set. The development set is obtained by 30% random selection from the training set of the Medline21. The layer-pair wise performance is shown in Figure 8, which indicates that setting cross-stitch between Layer4 and Layer5 achieves better AUC than the others, which might be because the information encoded by Layer4 is complementary. In addition, "all" fails to outperform the others, which might be because not all the layers are complementary, for instance the KG encoder provides very little syntactic information for the text encoder.

A.2 Dynamic Gate vs. Fixed Gate

Two strategies can be applied to calculate the weights for $\mathbf{G}_i^{t2s}$ and $\mathbf{G}_i^{s2t}$ in Figure 2: one is using fixed weights of gate throughout the entire training process; another is the proposed dynamic control of weights evaluated via Equation 4 and Equation 7 respectively. Table 6 and Table 7 show the performance comparison between the fixed gate and the proposed dynamic gate. We set the value of the fixed gate as 0.5 in this work. The results show that our proposed dynamic gate achieves better performance than the fixed gate, indicating the effectiveness of the proposed XBE model on dynamically controlling information flow from one layer to the next.

System	AUC	P@0.1k	P@0.2k	P@0.3k	P@0.5k	P@1k	P@2k
XBE (Fixed Gate)	68.82	98.0	**96.5**	94.0	92.4	84.5	62.2
XBE (Dynamic Gate)	**70.50**	**99.0**	96.0	**95.6**	**94.4**	**85.8**	**63.2**

Table 6: P@N and AUC from XBE with Fixed and Dynamic Gate on NYT10 dataset.

System	AUC	P@0.1k	P@0.2k	P@0.3k	P@0.5k	P@1k	P@2k
XBE (Fixed Gate)	61.87	93.0	89.5	88.7	86.2	**76.6**	**56.4**
XBE (Dynamic Gate)	**61.88**	**94.0**	**91.0**	**89.3**	**86.4**	76.1	56.1

Table 7: P@N and AUC from XBE with Fixed and Dynamic Gate on Medline21.

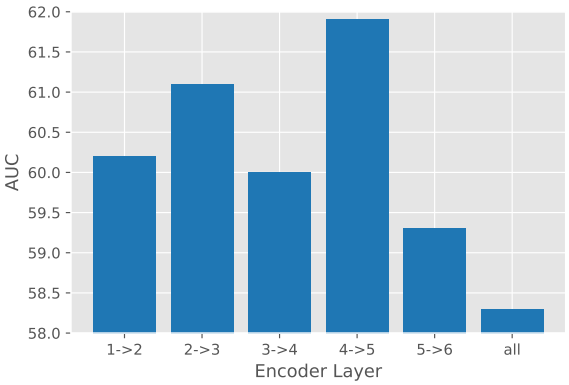


Figure 8: Interacting layer wise performance on a development set of Medline21, where, for instance "4->5" means locating cross-stitch between Layer4 and Layer5, as shown in Figure 1c, "all" means setting cross-stitch between all adjacent layers.

	Medline21		NYT10	
Model	AUC	P@2k	AUC	P@2k
BRE+CE	55.3	53.8	63.2	58.7
XBE	**61.9**	56.1	**70.5**	**63.2**
- $\mathbf{r}_{ht}$	59.02	**56.9**	68.64	62.1

Table 8: P@N and AUC from XBE with removed $\mathbf{r}_{ht}$.

A.3 Impact of $\mathbf{r}_{ht}$

We conduct ablation study to detect the impact of $\mathbf{r}_{ht}$ in (§4) on the overall performance. The results are shown in Table 8, where "- $\mathbf{r}_{ht}$" denotes the XBE model without $\mathbf{r}_{ht}$. We can observe that the performance slightly degrades without $\mathbf{r}_{ht}$, indicating that $\mathbf{r}_{ht}$ has limited contribution to the performance gain comparing with other components of the XBE model.