Context-Aware Answer Extraction in Question Answering

Yeon Seonwoo[†], **Ji-Hoon Kim**[‡§], **Jung-Woo Ha**[‡§], **Alice Oh**[†]
[†]KAIST
[‡]NAVER AI LAB, [§]NAVER CLOVA
yeon.seonwoo@kaist.ac.kr
{genesis.kim,jungwoo.ha}@navercorp.com
alice.oh@kaist.edu

Motivation

- models are designed to select answer-spans in the relevant contexts from given passages, they sometimes result in predicting the **correct answer text** but in contexts that are **irrelevant** to the given questions.

Passage: Some of the most developmentally significant changes in the brain occur in the prefrontal cortex, which is involved in decision making and cognitive control, as well as other higher cognitive functions. ... pruning in the prefrontal cortex increases, improving the efficiency of information processing, and neural connections between the prefrontal cortex and other regions of the brain are strengthened. ... Specifically, developments in the dorsolateral prefrontal cortex are important for controlling impulses and planning ahead, while development in the ventromedial prefrontal cortex is important for decision making.

Question: Which part of the brain is involved in decision making and cognitive control?

Answer: prefrontal cortex

Figure 1: Example passage, question, and answer triple. This passage has multiple spans that are matched with the answer text. The first occurrence of "prefrontal cortex" is the only answer-span within the context of the question.

Motivation

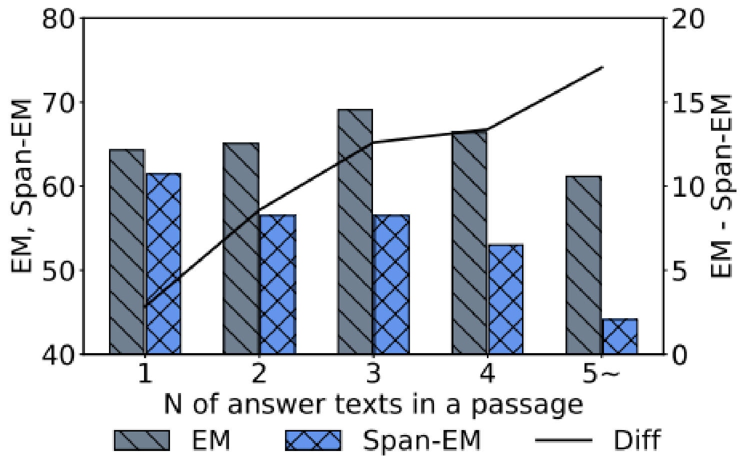


Figure 2: EM (text-matching) and Span-EM (span matching) of BERT on the groups divided by the number of answer text occurrences in a passage. Note: The difference for $N = 1$ results from post-processing steps (removing articles) in EM evaluation.

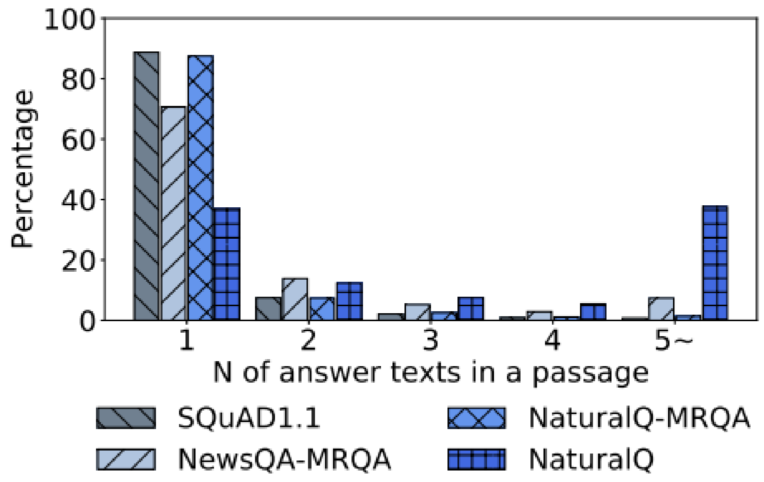


Figure 3: Proportions of questions with various numbers of the answer text in a passage. SQuAD has only a few examples for ($n \geq 5$), while NaturalQuestions has a large proportion.

Context

- We assume words near an answer-span are likely to be included in the context of a given question.

task. To achieve this, we hypothesize the words in an answer-span are included in the context and make the probability of adjacent words decrease with a specific ratio as the distance between answer-span and a word increases. The soft-label for the latent context is as follows:

$$p_{\text{soft}}(\mathrm{w}_i \in \mathcal{C}) = \begin{cases} 1.0 & \text{if } i \in [s_a, e_a] \\ q^{|i-s_a|} & \text{if } i < s_a \\ q^{|i-e_a|} & \text{if } i > e_a, \end{cases} \tag{1}$$

where $0 \leq q \leq 1$, and q is a hyper-parameter for the decreasing ratio as the distance from a given answer-span. For computational efficiency, we apply (1) to words bounded by certain window-size only, which is a hyper-parameter, on both sides of an answer-span. This results in assigning $p_{\text{soft}}(\mathrm{w}_i \in \mathcal{C})$ to 0 for the words outside the segment bounded by the window-size.

Model

- We propose BLANC based on two novel ideas:
  - 1. soft-labeling method for the context prediction
  - 2. a block attention method that predicts the soft-labels.

Model

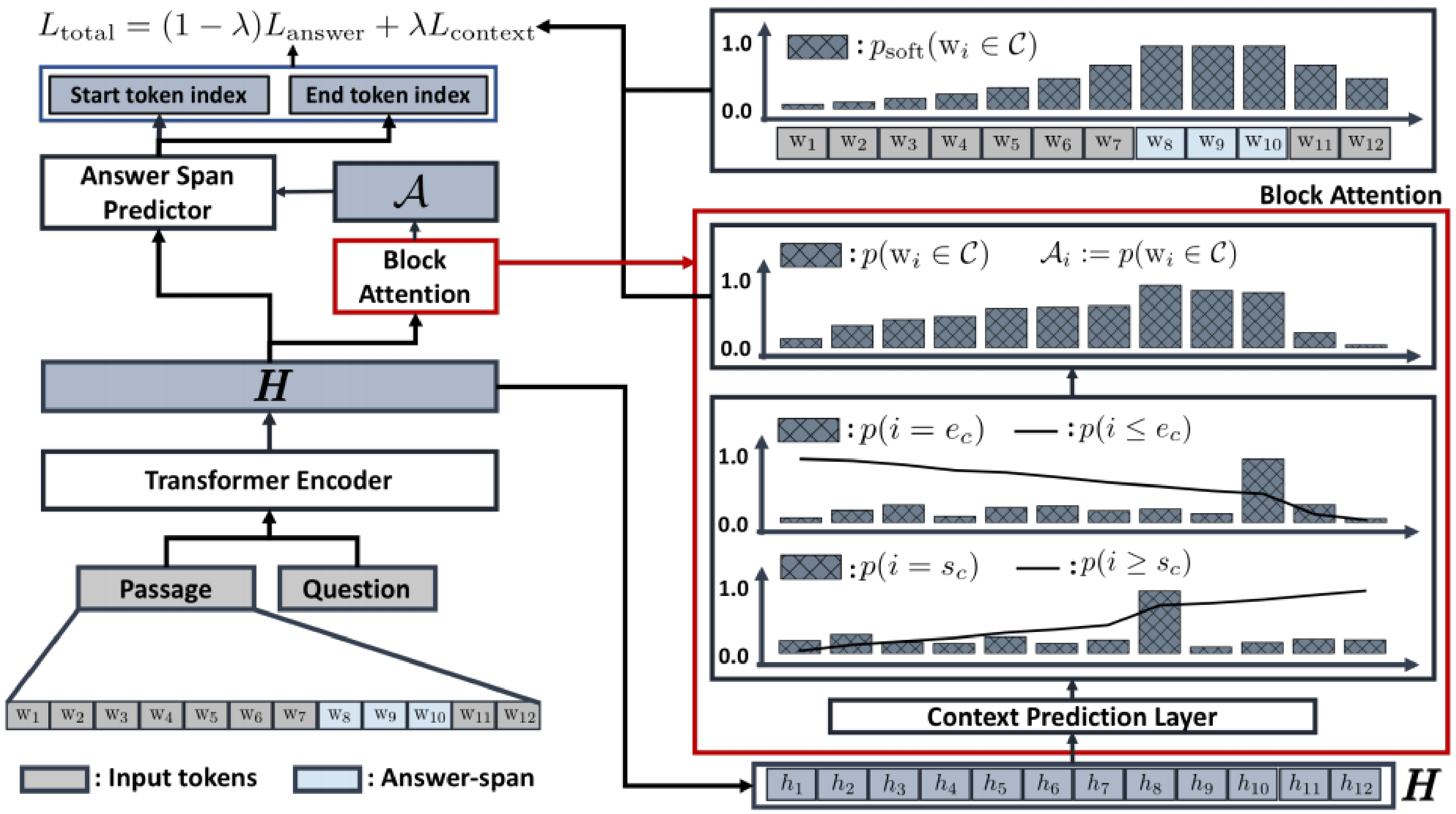


Figure 4: Schematic visualization of BLANC. Block attention model takes contextual vector representations from transformer encoder and predicts context words of an answer, $p(\text{w}_i \in \mathcal{C})$. We define loss function for context words with the prediction, $p(\text{w}_i \in \mathcal{C})$ and the self-generated soft-label $p_{\text{soft}}(\text{w}_i \in \mathcal{C})$ defined in (1). Answer-span predictor takes $p(\text{w}_i \in \mathcal{C}$ and $\boldsymbol{H}$ to predict an answer-span. We optimize our model in manner of multi-task learning of two tasks: answer-span prediction and context words prediction.

Block Attention

- In the first step, at predicting the start and end indices, all encoder models that produce vector representation of words in a passage are compatible with the block attention model.

$$\boldsymbol{H} = \text{Encoder}(\text{Passage}, \text{Question}) \qquad (2)$$

Here, we denote $\boldsymbol{H}$ as output vectors of transformer encoder and $\boldsymbol{H}_j$ as output vector of w_j. From $\boldsymbol{H}$, we predict the start and end indices of the context:

$$\begin{aligned} p(i = s_c) &= \frac{\exp(\boldsymbol{W_c}\boldsymbol{H}_i + b_s^c)}{\sum_j \exp(\boldsymbol{W_c}\boldsymbol{H}_j + b_s^c)}, \\ p(i = e_c) &= \frac{\exp(\boldsymbol{V_c}\boldsymbol{H}_i + b_e^c)}{\sum_j \exp(\boldsymbol{V_c}\boldsymbol{H}_j + b_e^c)}, \end{aligned} \qquad (3)$$

Block Attention

$$p(\mathrm{w}_i \in \mathcal{C}) = p(i \geq s_c) \times p(i \leq e_c). \tag{4}$$

Here, we assume the independence between s_c and e_c for computational conciseness. The cumulative distributions of $p(i \geq s_c)$ and $p(i \leq e_c)$ are calculated with the following equations:

$$\begin{aligned} p(i \geq s_c) &= \sum_{j \leq i} p(j = s_c) \\ p(i \leq e_c) &= \sum_{j \geq i} p(j = e_c). \end{aligned} \tag{5}$$

We explicitly force the block attention model to learn context words of a given question by minimizing the cross-entropy of the two probabilities, $p(\mathrm{w}_i \in \mathcal{C})$ and $p_{\mathrm{soft}}(\mathrm{w}_i \in \mathcal{C})$. The loss function for the latent context is defined by the following equation:

$$\begin{aligned} L_{\mathrm{context}} = &- \sum_{1 \leq i \leq l} p_{\mathrm{soft}}(\mathrm{w}_i \in \mathcal{C}) \log p(\mathrm{w}_i \in \mathcal{C}) \\ &- \sum_{1 \leq i \leq l} p_{\mathrm{soft}}(\mathrm{w}_i \notin \mathcal{C}) \log p(\mathrm{w}_i \notin \mathcal{C}), \end{aligned} \tag{6}$$

where l is the length of a passage. By averaging L_{context} across all train examples, we get the final context loss function.

Answer-span Prediction

BLANC predicts answer-span with the context probability, $p(\mathrm{w}_i \in \mathcal{C})$. We use the same answer-span prediction layer as BERT, but we multiply $p(\mathrm{w}_i \in \mathcal{C})$ to the output of the encoder, $\boldsymbol{H}$ to give attention at indices of answer-span within the context, $\mathcal{C}$.

$$
\begin{aligned}
p(i = s_a) &= \frac{\exp(\mathcal{A}_i \boldsymbol{W_a} \boldsymbol{H}_i + b_s^a)}{\sum_j \exp(\mathcal{A}_j \boldsymbol{W_a} \boldsymbol{H}_j + b_s^a)}, \\
p(i = e_a) &= \frac{\exp(\mathcal{A}_i \boldsymbol{V_a} \boldsymbol{H}_i + b_e^a)}{\sum_j \exp(\mathcal{A}_j \boldsymbol{V_a} \boldsymbol{H}_j + b_e^a)},
\end{aligned}
\tag{7}
$$

where $\boldsymbol{W_a}$, $\boldsymbol{V_a}$, b_s^a, and b_e^a represent weight and bias parameters for answer-span prediction layer, and $\mathcal{A}_i = p(\mathrm{w}_i \in \text{context})$. The loss function for answer-span prediction is defined by the following equation:

$$
\begin{aligned}
L_{\text{answer}} = -\frac{1}{2}\{ &\sum_{1 \le i \le l} \mathbb{1}(i = s_a) \log p(i = s_a) \\
&+ \sum_{1 \le i \le l} \mathbb{1}(i = e_a) \log p(i = e_a)\}.
\end{aligned}
\tag{8}
$$

$\mathbb{1}$(condition) represents an indicator function that returns 1 if the condition is true and returns 0 otherwise. By averaging L_{answer} across all train examples, we get the final answer-span loss function. We

Experiment

		#Param	Span-F1	Span-EM	F1	EM
NaturalQA	BERT	108M	72.92 ± 0.36	60.63 ± 0.39	76.39 ± 0.26	64.48 ± 0.28
	ALBERT	17M	72.66 ± 0.48	60.31 ± 0.49	75.89 ± 0.36	63.81 ± 0.37
	RoBERTa	124M	75.07 ± 0.17	62.59 ± 0.14	78.54 ± 0.20	66.33 ± 0.09
	SpanBERT	108M	75.16 ± 0.26	62.71 ± 0.37	78.31 ± 0.22	66.60 ± 0.31
	BLANC	108M	$\mathbf{76.99 \pm 0.09}$	$\mathbf{64.57 \pm 0.12}$	$\mathbf{80.04 \pm 0.06}$	$\mathbf{68.33 \pm 0.09}$
	$\text{SpanBERT}_{\text{large}}$	333M	77.62 ± 0.10	65.28 ± 0.41	80.66 ± 0.11	69.14 ± 0.18
	$\text{BLANC}_{\text{large}}$	333M	$\mathbf{79.04 \pm 0.27}$	$\mathbf{66.75 \pm 0.14}$	$\mathbf{81.99 \pm 0.16}$	$\mathbf{70.59 \pm 0.12}$
SQuAD1.1	BERT	108M	83.36 ± 0.25	70.74 ± 0.43	88.10 ± 0.14	80.49 ± 0.28
	ALBERT	17M	84.60 ± 0.13	72.04 ± 0.38	88.75 ± 0.20	81.05 ± 0.27
	RoBERTa	124M	85.21 ± 0.25	72.82 ± 0.56	89.91 ± 0.16	82.53 ± 0.44
	SpanBERT	108M	86.67 ± 0.16	74.08 ± 0.13	91.58 ± 0.09	84.97 ± 0.18
	BLANC	108M	$\mathbf{86.89 \pm 0.15}$	$\mathbf{74.69 \pm 0.37}$	$\mathbf{91.87 \pm 0.13}$	$\mathbf{85.30 \pm 0.32}$
	$\text{SpanBERT}_{\text{large}}$	333M	88.27 ± 0.14	75.96 ± 0.22	93.22 ± 0.08	87.14 ± 0.11
	$\text{BLANC}_{\text{large}}$	333M	$\mathbf{88.42 \pm 0.17}$	$\mathbf{76.26 \pm 0.31}$	$\mathbf{93.37 \pm 0.05}$	$\mathbf{87.30 \pm 0.10}$
NewsQA	BERT	108M	59.18 ± 0.57	45.53 ± 0.55	65.07 ± 0.52	50.11 ± 0.50
	ALBERT	17M	60.12 ± 0.36	46.54 ± 0.04	66.02 ± 0.35	51.18 ± 0.18
	RoBERTa	124M	61.36 ± 0.63	47.43 ± 0.54	67.28 ± 0.63	52.36 ± 0.64
	SpanBERT	108M	62.26 ± 0.22	48.04 ± 0.48	67.93 ± 0.26	52.85 ± 0.49
	BLANC	108M	$\mathbf{64.39 \pm 0.76}$	$\mathbf{50.60 \pm 0.50}$	$\mathbf{70.31 \pm 0.66}$	$\mathbf{55.52 \pm 0.43}$
	$\text{SpanBERT}_{\text{large}}$	333M	63.43 ± 0.42	49.03 ± 0.13	69.06 ± 0.55	53.84 ± 0.27
	$\text{BLANC}_{\text{large}}$	333M	$\mathbf{66.48 \pm 0.20}$	$\mathbf{52.39 \pm 0.08}$	$\mathbf{72.36 \pm 0.01}$	$\mathbf{57.40 \pm 0.21}$

Table 1: Reading comprehension performance of baseline models and BLANC. We conduct experiments on three QA datasets: NaturalQ, SQuAD1.1, and NewsQA. For all evaluation metrics, we report mean and standard deviation of three separate trials. The results show that BLANC outperforms baseline models.

Experiment

	Span-F1	Span-EM
RoBERTa	65.99 ± 0.92	60.12 ± 0.86
SpanBERT	63.47 ± 0.72	57.63 ± 0.79
BLANC	$\mathbf{67.07 \pm 0.36}$	$\mathbf{61.43 \pm 0.38}$

Table 2: Performance of BLANC on passages of NaturalQ that have answer texts two or more.

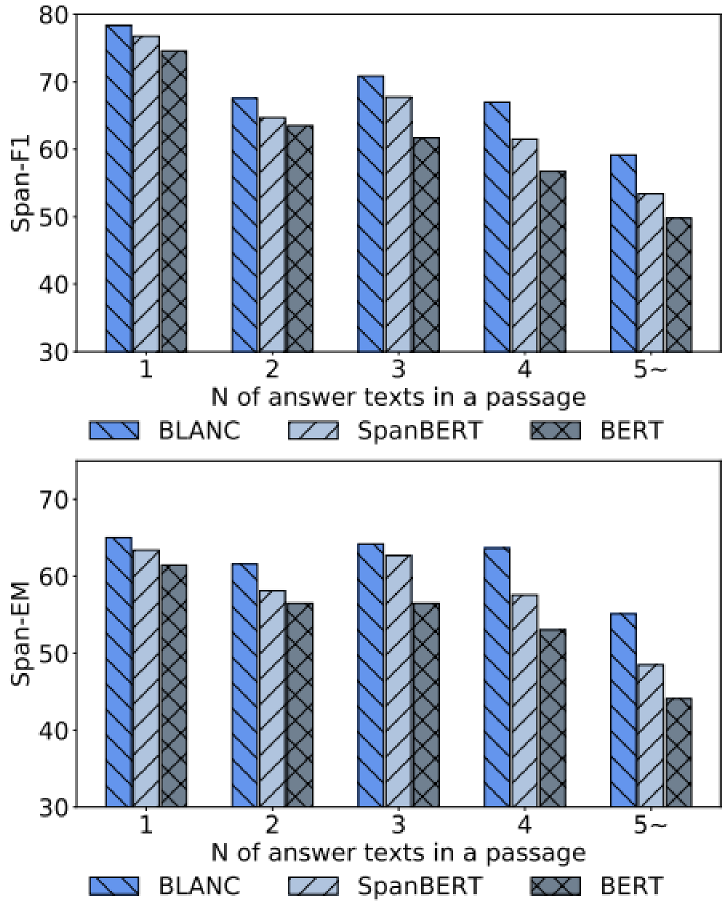
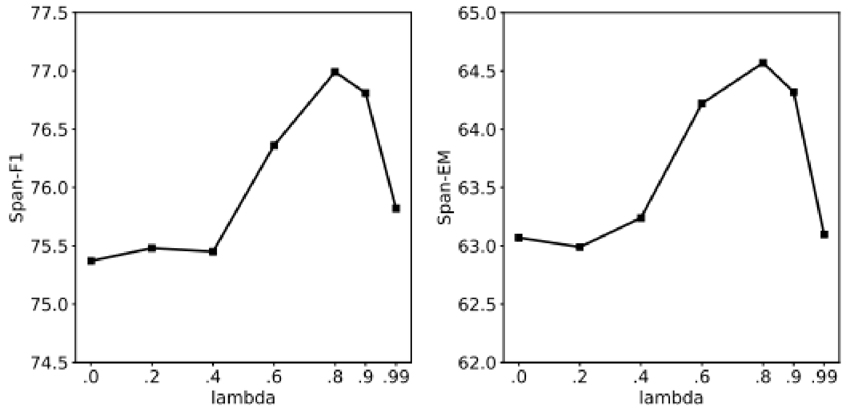


Figure 5: Span-F1 and Span-EM of baseline models and BLANC trained on NaturalQ. We categorize NaturalQ dataset into five groups by number answer texts appeared in a passage: $n = 1, 2, 3, 4$, and $n \geq 5$. BLANC outperforms baseline models on every groups and the performance gap increases as the number of answer texts in a passage increases.

Experiment



(a) Span-F1 (b) Span-EM

Figure 6: Analysis on λ for context word prediction for NaturalQ. We adjust λ, weight of (L_{context}), from 0.0 to 0.99 and report Span-F1 and Span-EM. Increasing λ improves answer-span prediction until $\lambda = 0.8$ and then decreases. This decrease is expected as the weight for (L_{answer}) becomes too small.

	Train	Inf
BERT	1.00x	1.00x
ALBERT$_{\text{large}}$	1.42x	1.89x
RoBERTa	1.01x	1.02x
SpanBERT	1.00x	1.00x
BLANC	1.04x	1.00x

Table 4: Training and inference time of each model measured on the same number of QA pairs.

Single-/Multi-Source Cross-Lingual NER via Teacher-Student Learning on Unlabeled Data in Target Language

Qianhui Wu[1], **Zijia Lin**[2], **Börje F. Karlsson**[2], **Jian-Guang Lou**[2], **and Biqing Huang**[1]
[1]Beijing National Research Center for Information Science and Technology (BNRist)
Department of Automation, Tsinghua University, Beijing 100084, China
wuqianhui@tsinghua.org.cn, hbq@tsinghua.edu.cn
[2]Microsoft Research, Beijing 100080, China
{zijlin,borje.karlsson,jlou}@microsoft.com

Motivation

- Previous works on cross-lingual NER are mostly based on <u>label projection</u> with pairwise texts or <u>direct model transfer</u>.

- However, such methods either are not applicable if the labeled data in the source languages is unavailable, or do not leverage information contained in unlabeled data in the target language.

model

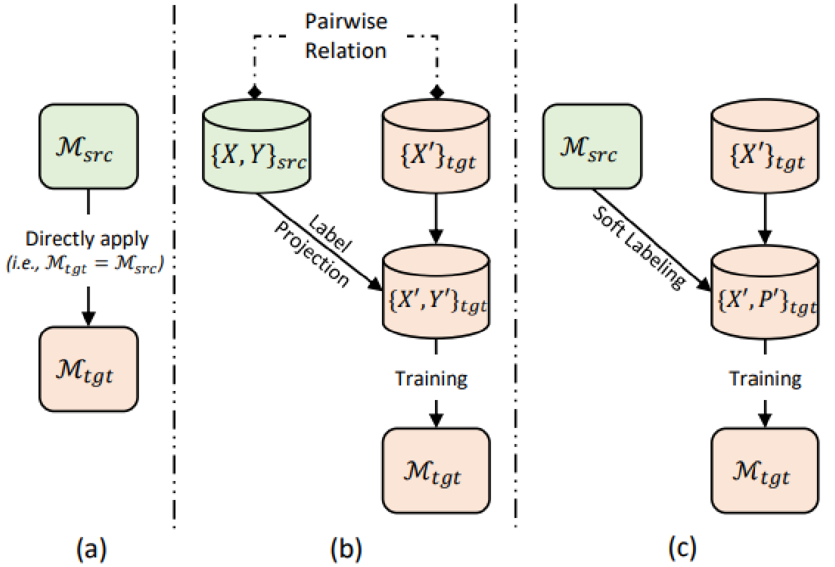


Figure 1: Comparison between previous cross-lingual NER methods (**a**/**b**) and the proposed method (**c**). (**a**): direct model transfer; (**b**): label projection with pairwise texts; (**c**): proposed teacher-student learning method. $\mathcal{M}_{src/tgt}$: learned NER model for source/target language; $\{X, Y\}_{src}$: labeled data in source language; $\{X'\}_{tgt}$: unlabeled data in target language; $\{X', Y'\}_{tgt}/\{X', P'\}_{tgt}$: pseudo-labeled data in target language with hard labels / soft labels.

model

1. We leverage multilingual BERT (Devlin et al., 2019) as the base model to produce language-independent features.
2. A previously trained NER model for the source language is then used as a teacher model to predict the probability distribution of entity labels (i.e., soft labels) for each token in the non-pairwise unlabeled data in the target language.
3. Finally, we train a student NER model for the target language using the pseudo-labeled data with such soft labels.

Model

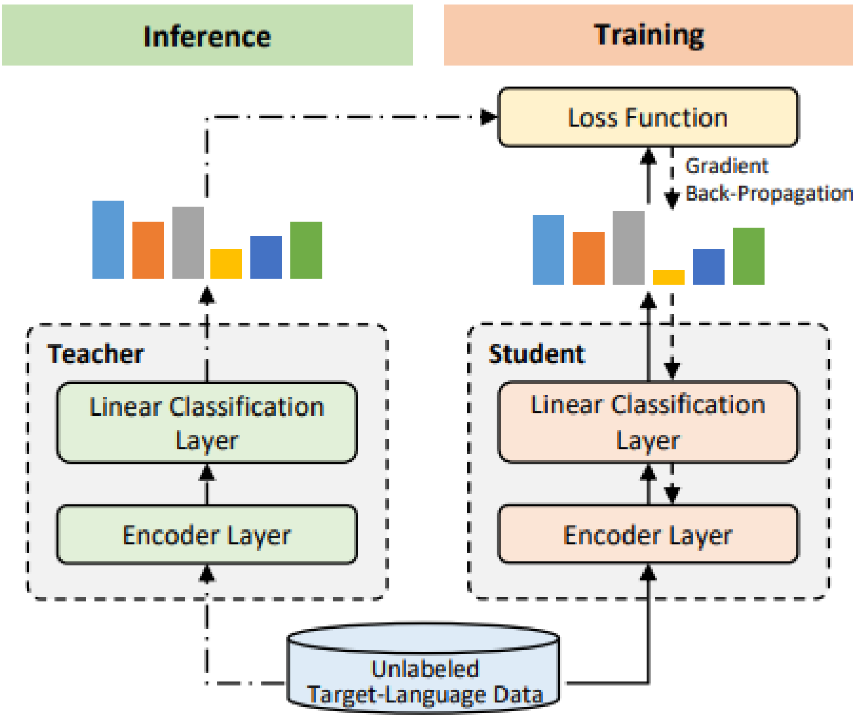


Figure 2: Framework of the proposed teacher-student learning method for **single-source** cross-lingual NER.

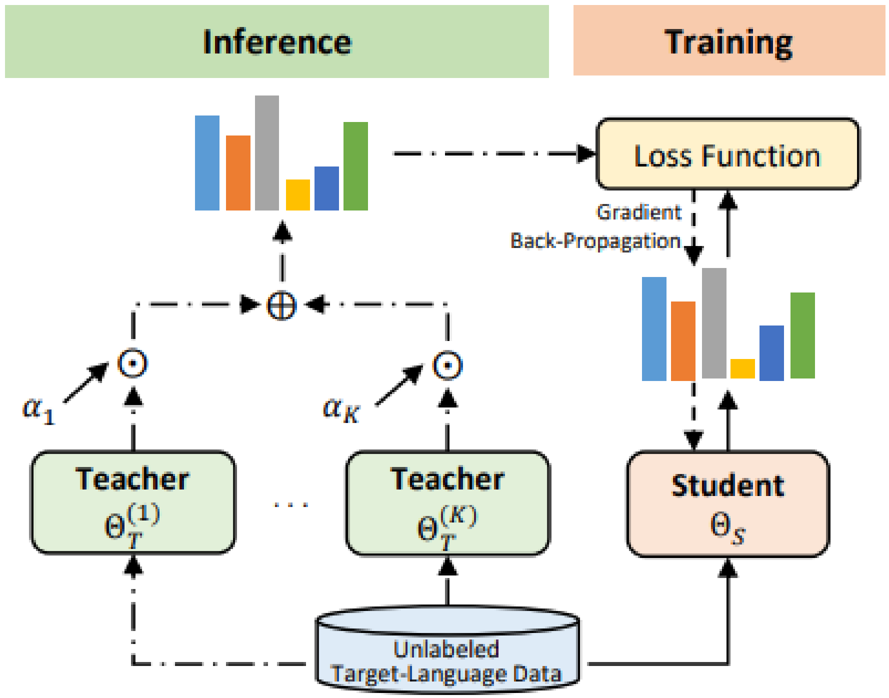


Figure 3: Framework of the proposed teacher-student learning method for **multi-source** cross-lingual NER.

Single-Source Cross-Lingual NER

With each h_i derived, the linear classification layer computes the probability distribution of entity labels for the corresponding token x_i, using a *softmax* function:

$$p(x_i, \Theta) = \text{softmax}(Wh_i + b) \qquad (2)$$

where $p(x_i, \Theta) \in \mathbb{R}^{|C|}$ with C being the entity label set, and $\Theta = \{f_\theta, W, b\}$ denotes the to-be-learned model parameters.

els. The teacher-student learning loss *w.r.t* $\boldsymbol{x'}$ is then defined as:

$$\mathcal{L}(\boldsymbol{x'}, \Theta_S) = \frac{1}{L} \sum_{i=1}^{L} \text{MSE}\left(\hat{p}(x'_i, \Theta_S), \tilde{p}(x'_i, \Theta_T)\right) \qquad (3)$$

And the whole training loss is the summation of losses *w.r.t* all sentences in D_{tgt}, as defined below.

$$\mathcal{L}(\Theta_S) = \sum_{\boldsymbol{x'} \in D_{tgt}} \mathcal{L}(\boldsymbol{x'}, \Theta_S) \qquad (4)$$

Minimizing $\mathcal{L}(\Theta_S)$ will derive the student model.

Multi-Source Cross-Lingual NE

as $\tilde{p}(x'_i, \Theta_T^{(k)})$. To combine all teacher models, we add up their output probability distributions with a group of weights $\{\alpha_k\}_{k=1}^K$ as follows.

$$\tilde{p}(x'_i, \Theta_T) = \sum_{k=1}^{K} \alpha_k \cdot \tilde{p}(x'_i, \Theta_T^{(k)}) \qquad (6)$$

where $\tilde{p}(x'_i, \Theta_T)$ is the combined probability distribution of entity labels, $\Theta_T = \{\Theta_T^{(k)}\}_{k=1}^K$ is the set of parameters of all teacher models, and α_k is the weight corresponding to the k-th teacher model, with $\sum_{k=1}^K \alpha_k = 1$ and $\alpha_k \geq 0, \forall k \in \{1, \ldots, K\}$.

Without Any Source-Language Data: It is straightforward to average over all teacher models:

$$\alpha_k = \frac{1}{K}, \forall k \in \{1, 2, \ldots, K\} \qquad (7)$$

Weighting Teacher Models

- we propose to introduce a language identification auxiliary task for calculating **similarities between source and target languages**, and then weight teacher models based on this metric

$\{(\boldsymbol{u}^{(k)}, k)\}$. We also assume that in the m-dimensional language-independent feature space, sentences from each source language should be clustered around the corresponding language embedding vector. We thus introduce a learnable language embedding vector $\mu^{(k)} \in \mathbb{R}^m$ for the k-th source language, and then utilize a *bilinear* operator to measure similarity between a given sentence $\boldsymbol{u}$ and the k-th source language:

$$s(\boldsymbol{u}, \mu^{(k)}) = g^T(\boldsymbol{u}) M \mu^{(k)} \tag{8}$$

where $g(\cdot)$ can be any language-independent model that outputs sentence embeddings, and $M \in \mathbb{R}^{m \times m}$ denotes the parameters of the *bilinear* operator.

With learned M and $P = [\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(K)}]$, we compute the weights $\{\alpha_k\}_{i=1}^{K}$ using the unlabeled data in the target language D_{tgt}:

$$\alpha_k = \frac{1}{|D_{tgt}|} \sum_{\boldsymbol{x}' \in D_{tgt}} \frac{\exp\left(s(\boldsymbol{x}', \mu^{(k)})/\tau\right)}{\sum_{i=1}^{K} \exp\left(s(\boldsymbol{x}', \mu^{(i)})/\tau\right)} \tag{11}$$

where τ is a temperature factor to smooth the output probability distribution. In our experiments, we set it as the variance of all values in $\{s(\boldsymbol{x}', \mu^{(k)})\}, \forall \boldsymbol{x}' \in D_{tgt}, \forall k \in \{1, ..., K\}$, so that α_k would not be too biased to either 0 or 1.

Weighting Teacher Models

By building a language embedding matrix $P \in \mathbb{R}^{m \times K}$ with each $\mu^{(k)}$ *column by column*, and applying a *softmax* function over the *bilinear* operator, we can derive language-specific probability distributions *w.r.t* $\boldsymbol{u}$ as below.

$$q(\boldsymbol{u}, M, P) = \text{softmax}\left(g^T(\boldsymbol{u}) M P\right) \tag{9}$$

Then the parameters M and P are trained to identify the language of each sentence in $\{D_{src}^{(k)}\}_{k=1}^{K}$, via minimizing the *cross-entropy* (CE) loss:

$$\mathcal{L}(P, M) = -\frac{1}{Z} \sum_{(\boldsymbol{u}^{(k)}, k) \in D_{src}} \text{CE}\left(q(\boldsymbol{u}^{(k)}, M, P), k\right) + \gamma \|PP^T - I\|_F^2 \tag{10}$$

where D_{src} is the union set of $\{D_{src}^{(k)}\}_{k=1}^{K}$, $Z = |D_{src}|$, $\|\cdot\|_F^2$ denotes the squared Frobenius norm, and I is an identity matrix. The regularizer in $\mathcal{L}(P, M)$ is to encourage different dimensions of the language embedding vectors to focus on different aspects, with $\gamma \geq 0$ being its weighting factor.

Experiments

	es	nl	de
Täckström et al. (2012)	59.30	58.40	40.40
Tsai et al. (2016)	60.55	61.56	48.12
Ni et al. (2017)	65.10	65.40	58.50
Mayhew et al. (2017)	65.95	66.50	59.11
Xie et al. (2018)	72.37	71.25	57.76
Wu and Dredze (2019)†	74.50	79.50	71.10
Moon et al. (2019)†	75.67	80.38	71.42
Wu et al. (2020)	76.75	80.44	73.16
Ours	**76.94**	**80.89**	**73.22**

Table 2: Performance comparisons of **single-source** cross-lingual NER. † denotes the reported results *w.r.t.* freezing the bottom three layers of $\text{BERT}_{\text{BASE}}$ as in this paper.

English as the source language

	es	nl	de
Täckström (2012)	61.90	59.90	36.40
Rahimi et al. (2019)	71.80	67.60	59.10
Chen et al. (2019)	73.50	72.40	56.00
Moon et al. (2019)†	76.53	**83.35**	72.44
Ours-avg	77.75	80.70	74.97
Ours-sim	**78.00**	81.33	**75.33**

Table 3: Performance comparisons of **multi-source** cross-lingual NER. **Ours-avg**: averaging teacher models (Eq. 7) . **Ours-sim**: weighting teacher models with learned language similarities (Eq. 11). † denotes the reported results *w.r.t.* freezing the bottom three layers of $\text{BERT}_{\text{BASE}}$.

Experiments

	es	nl	de
Single-source:			
Ours	**76.94**	**80.89**	**73.22**
HL	76.60 (-0.34)	80.43 (-0.46)	72.98 (-0.24)
MT	75.60 (-1.34)	79.99 (-0.90)	71.76 (-1.46)
Multi-source:			
Ours-avg	**77.75**	**80.70**	**74.97**
HL-avg	77.65 (-0.10)	80.39 (-0.31)	74.31 (-0.66)
MT-avg	77.25 (-0.50)	80.53 (-0.17)	74.18 (-0.79)
Ours-sim	**78.00**	**81.33**	**75.33**
HL-sim	77.81 (-0.19)	80.27 (-1.06)	74.63 (-0.70)
MT-sim	77.12 (-0.88)	80.24 (-1.09)	74.33 (-1.00)

Table 4: Ablation study of the proposed teacher-student learning method for cross-lingual NER. **HL**: Hard Label; **MT**: Direct Model Transfer; ***-avg**: averaging source-language models; ***-sim**: weighting source-language models with learned language similarities.

	es	nl	de
Ours	**78.00**	**81.33**	**75.33**
$cosine$	77.86 (-0.14)	79.94 (-1.39)	75.24 (-0.09)
ℓ_2	77.72 (-0.28)	79.74 (-1.59)	75.09 (-0.24)

Table 5: Comparison between the proposed language similarity measuring method and the commonly used $cosine/\ell_2$ metrics for multi-source cross-lingual NER.