Paper Sharing

- *LRC-BERT*: Latent-representation Contrastive Knowledge Distillation for Natural Language Understanding

- Logic-guided Semantic Representation Learning for Zero-Shot Relation Classification

- *LayoutLMv2*: Multi-modal Pre-training for Visually-Rich Document Understanding

——2021/01/21 朱静丹

LRC-BERT: Latent-representation Contrastive Knowledge Distillation for Natural Language Understanding

Hao Fu,[1*] Shaojun Zhou,[2*] Qihong Yang,[2] Junjie Tang,[2†] Guiquan Liu,[1]
Kaikui Liu,[2] Xiaolong Li[2]
[1]School of Computer Science and Technology, University of Science and Technology of China
[2]Alibaba Group
hfu@mail.ustc.edu.cn, {zsj148798, xiaokui.yqh, lixi.tjj}@alibaba-inc.com, gqliu@ustc.edu.cn,
{damon, xl.li}@alibaba-inc.com

——高德(AAAI2021)

The contributions of this paper:

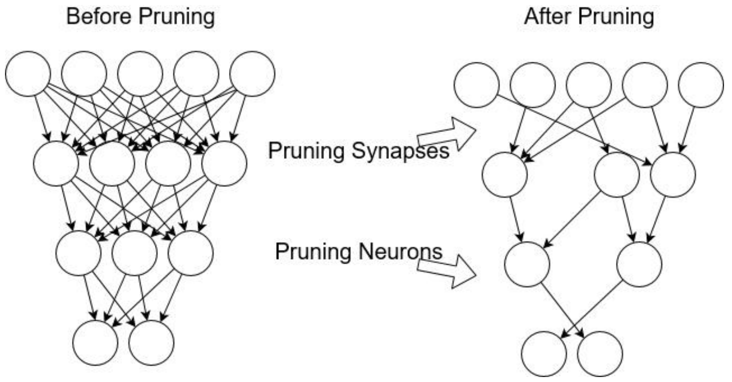
- They propose a knowledge distillation method LRC-BERT based on contrastive learning .
- A new contrastive loss COS-NCE is proposed to effectively capture the structural characteristics between different samples.
- They introduce a gradient perturbation-based training architecture in the training phase to increase the robustness of LRC-BERT.
- They design a two stage training method for the total distillation loss.

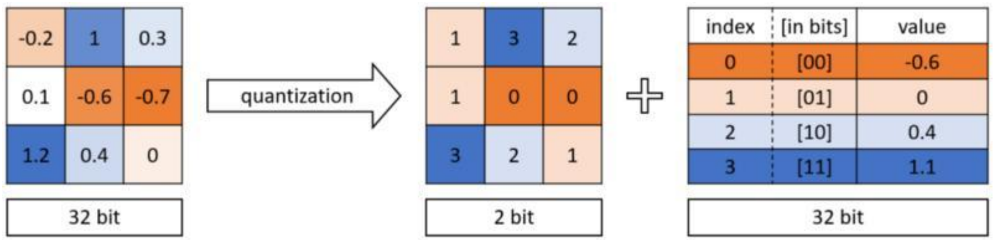
Background

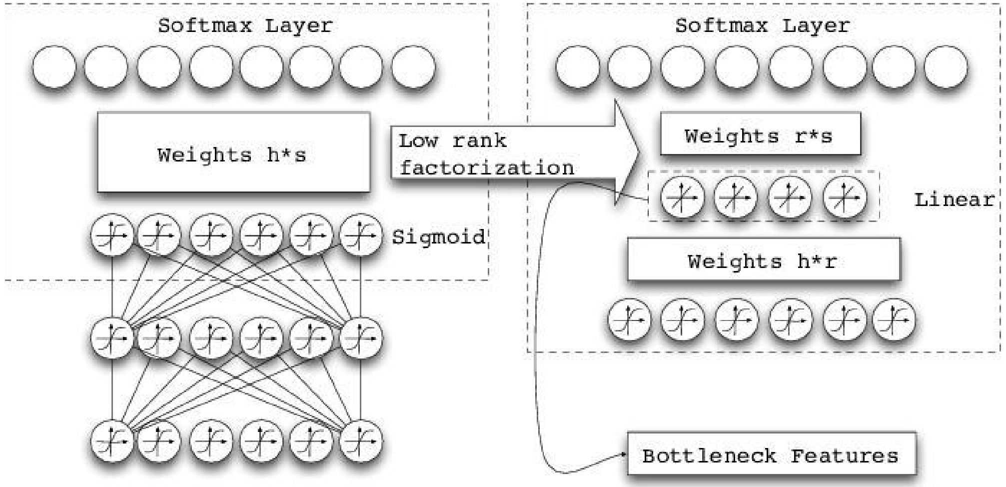
模型压缩

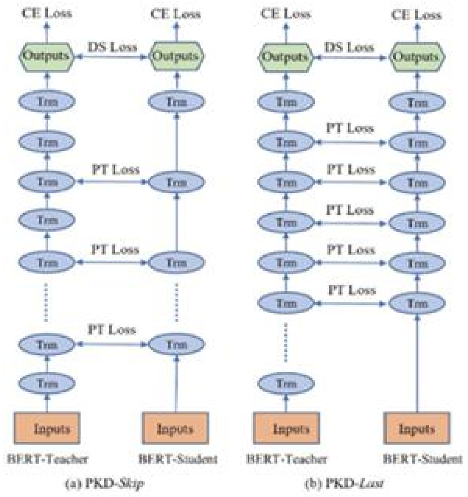
Pruning裁剪 Weight Sharing权重共享 Quantization量化

Factorization因子分解 Knowledge distillation知识蒸馏

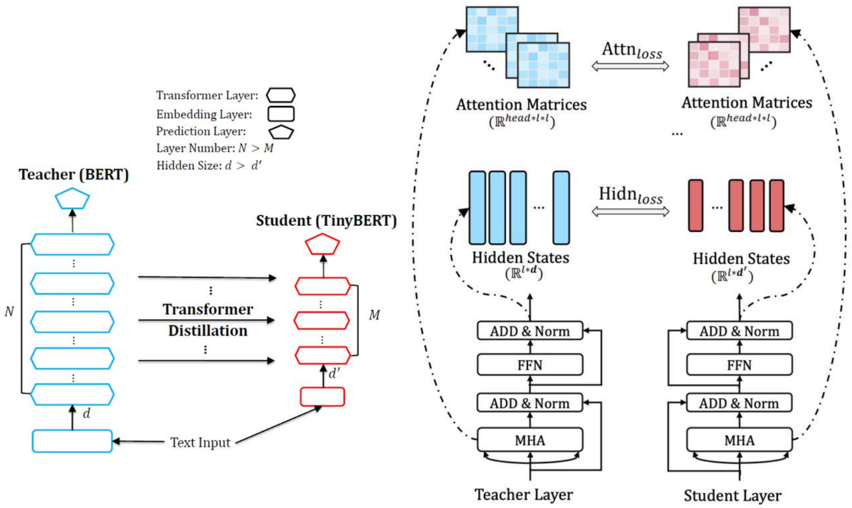




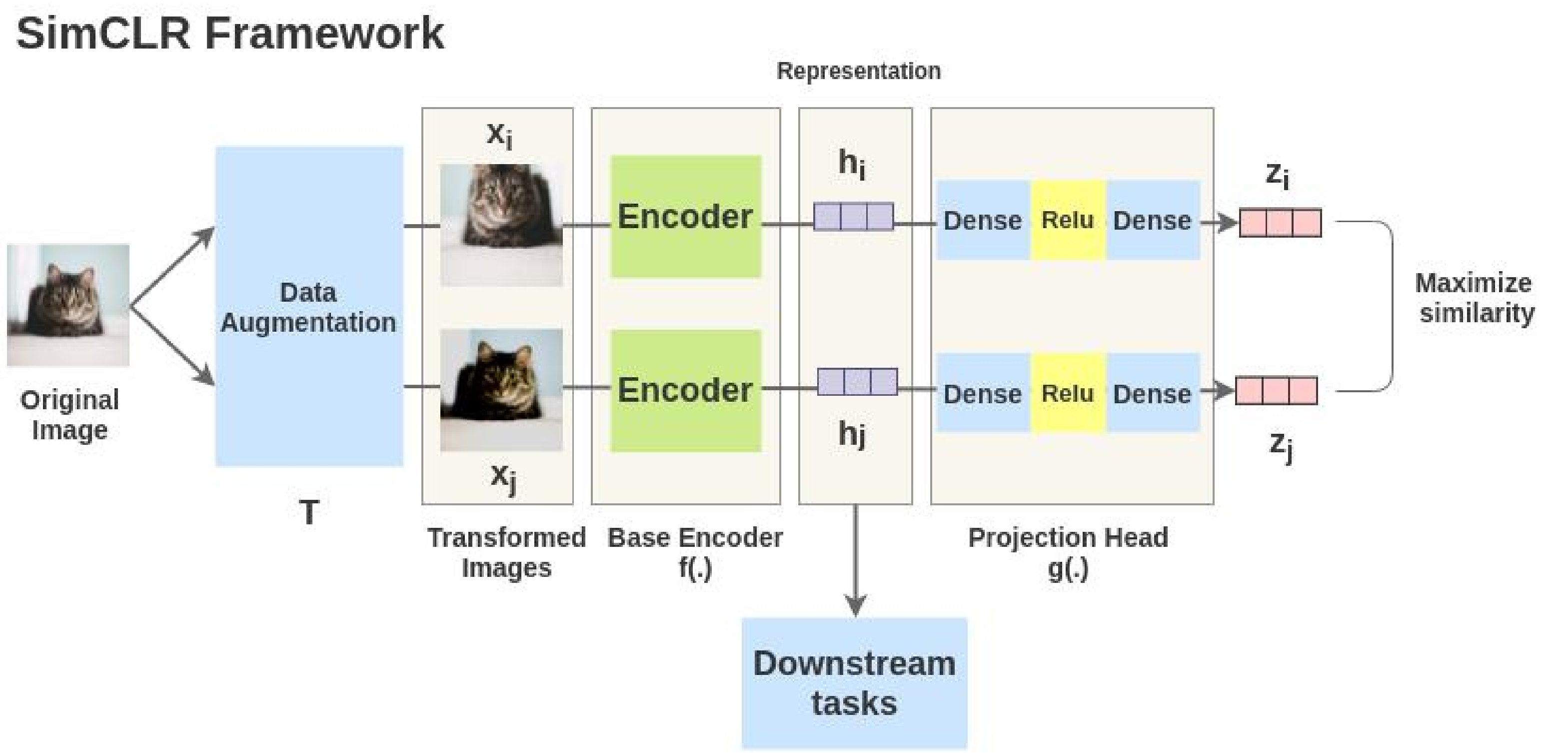




BERT-PKD

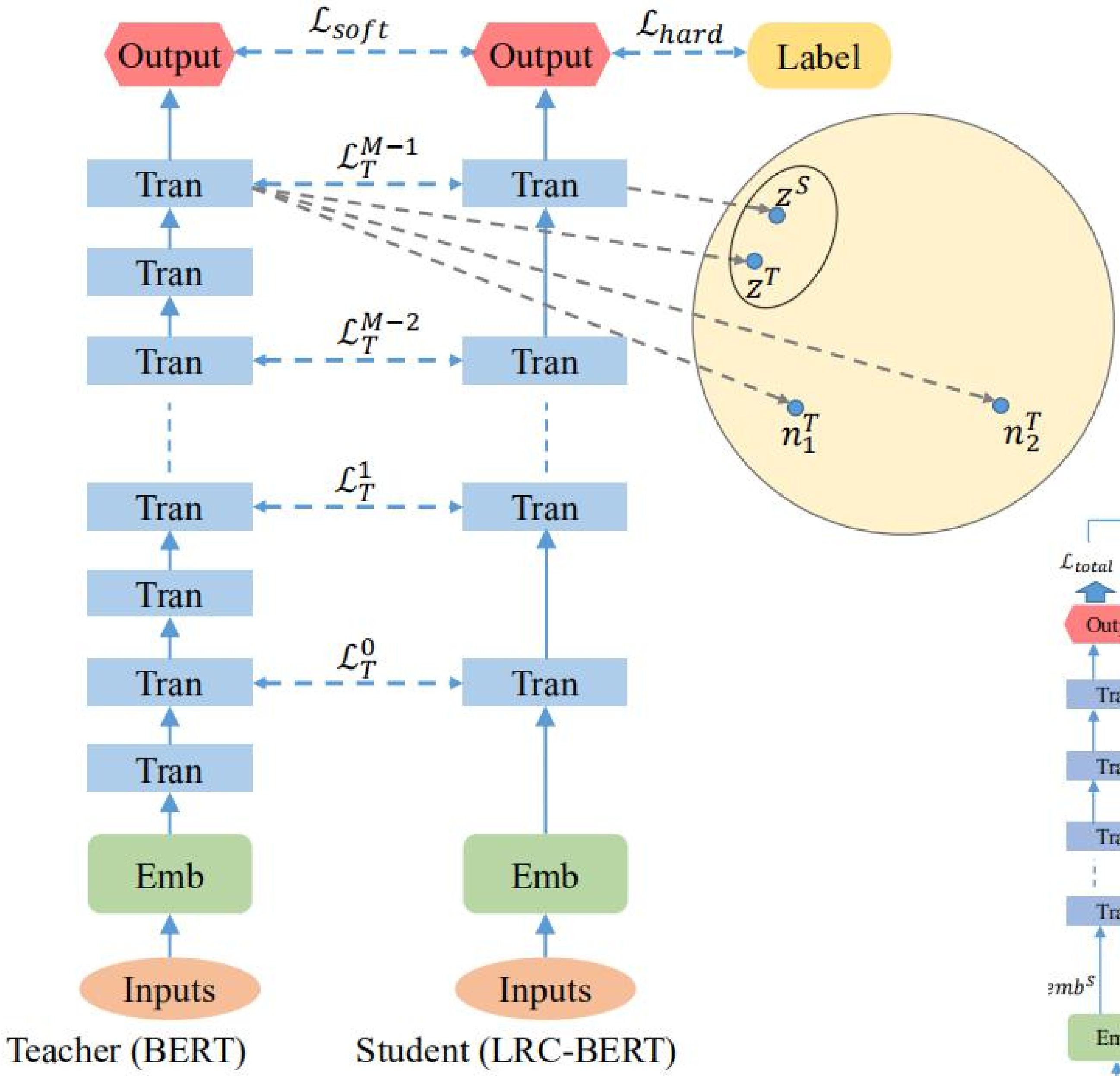


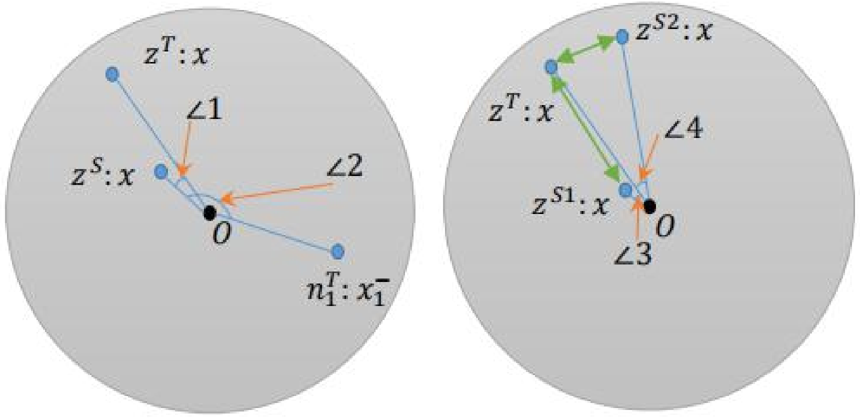
Background


SimCLR Framework
Representation
Original Image
Data Augmentation
T
x_i
x_j
Transformed Images
Encoder
Encoder
Base Encoder f(.)
h_i
h_j
Dense
Relu
Dense
Dense
Relu
Dense
Projection Head g(.)
z_i
z_j
Maximize similarity
Downstream tasks

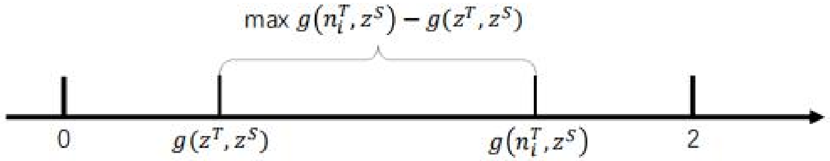

1
1
ONE
DOLLAR
1
1
In God we trust

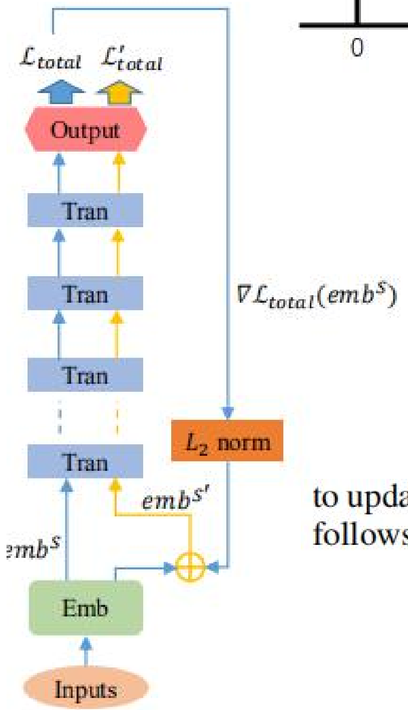

FEDERAL RESERVE NOTE
THE UNITED STATES OF AMERICA
1
1
L
WASHINGTON, D.C.
ONE
ONE DOLLAR





(a) (b)





to update the network parameters. The specific process is as follows:

$$\boxed{emb^{S'} = emb^S + g/||g||_2,} \tag{7}$$

$$g = \nabla \mathcal{L}_{total}(emb^S). \tag{8}$$

Setting

Distillation Setup

We use BERT-base (Devlin et al. 2019) as our teacher.

For the distillation of each task on GLUE, we fine-tune a BERT-base teacher, choosing learning rates of 5e-5, 1e-4, and 3e-4 with batchsize of 16 to distill LRC-BERT and LRC-BERT$_1$. For each sample, we choose the remaining 15 samples in batchsize as negative samples, i.e. $K = 15$. Among them, 90 epochs of distillation are performed on the MRPC, RTE, and CoLA with the training dataset less than 10K, and 18 epochs of distillation on other tasks. For the proposed two-stage training method, the first 80% of the steps are chosen as the first stage of training, the rest 20% of

Datasets

We evaluate LRC-BERT on GLUE benchmark. The datasets

Results

Model	Accuracy
LRC-BERT	83.4
LRC-BERT$_2$	79.4

Table 4: Effect of two-stage training method on MNLI-m task (dev).

Results

Model	Params	MNLI-m (393k)	MNLI-mm (393k)	QQP (364k)	SST-2 (67k)	QNLI (105k)	MRPC (3.7k)	RTE (2.5k)	CoLA (8.5k)	STS-B (5.7k)	Avg
BERT-base (teacher)	109M	84.3	83.8	71.4	93.6	90.9	88.0	66.4	53.0	84.8	79.6
DistilBERT	52.2M	78.9	78.0	68.5	91.4	85.2	82.4	54.1	32.8	76.1	71.9
BERT-PKD	52.2M	79.9	79.3	70.2	89.4	85.1	82.6	62.3	24.8	79.8	72.6
TinyBERT	14.5M	82.5	81.8	71.3	92.6	87.7	86.4	62.9	43.3	79.9	76.5
LRC-BERT$_1$	14.5M	82.8	82.6	71.9	90.7	88.3	83.0	51.0	31.6	79.8	73.5
LRC-BERT	14.5M	**83.1**	**82.7**	**72.2**	**92.9**	**88.7**	**87.0**	**63.1**	**46.5**	**81.2**	**77.5**

Table 1: The results are evaluated from the official website of GLUE benchmark, and the optimal experimental results are identified in bold. The number under each dataset represents the corresponding number of training samples.

Model	transformer layers	hidden size	Params	inference time(s)
BERT-base	12	768	109M	121.4
LRC-BERT	4	312	14.5M	12.7

Table 2: The number of parameters and inference time before and after model compression.

Model	MNLI-m	MNLI-mm	MRPC	CoLA
LRC-BERT	83.4	83.5	89.0	50.0
LRC-BERT$_C$	78.0	78.2	81.5	37.0
LRC-BERT$_S$	82.7	83.0	89.4	48.8
LRC-BERT$_H$	83.0	83.5	88.7	48.6

Table 3: Ablation studies of different loss functions (dev).

Logic-guided Semantic Representation Learning for Zero-Shot Relation Classification

Juan Li[1,2]*, **Ruoxu Wang**[1,2]*, **Ningyu Zhang**[1,2*†], **Wen Zhang**[1,2], **Fan Yang**[1,2], **Huajun Chen**[1,2 †]
[1] Zhejiang University
[2] AZFT Joint Lab for Knowledge Engine
{lijuan18,ruoxuwang,zhangningyu,wenzhang2015,21821249,huajunsir}@zju.edu.cn

——浙大(COLING2020)

The contributions of this paper:

- To recognize unseen relations at test time, they explore the problem of zero-shot relation classification.
- They propose a novel logic-guided semantic representation learning model for zero-shot relation classification.

Background

① 知识图嵌入的隐含语义联系

Implicit Semantic Connection with Knowledge Graph Embedding. Previous studies (Yang et al., 2014) have shown that the Knowledge Graph Embeddings (KGEs) of semantically similar relations are located near each other in the latent space.

② 规则学习明确的语义联系

Explicit Semantic Connection with Rule Learning. We human can easily recognize unseen relations via symbolic reasoning. As the example shown in Figure 1, with the rule that *basin_country_of(y,z)* can be deduced if *located_in_country(x,y)* and *next_to_body_of_water(x,z)*, we can recognize the unseen relation *basin_country_of* based on seen relations *located_in_country* and *next_to_body_of_water*. To this end, it is intuitive to infuse rule knowledge to bridge the connections between seen and zero-shot relations.

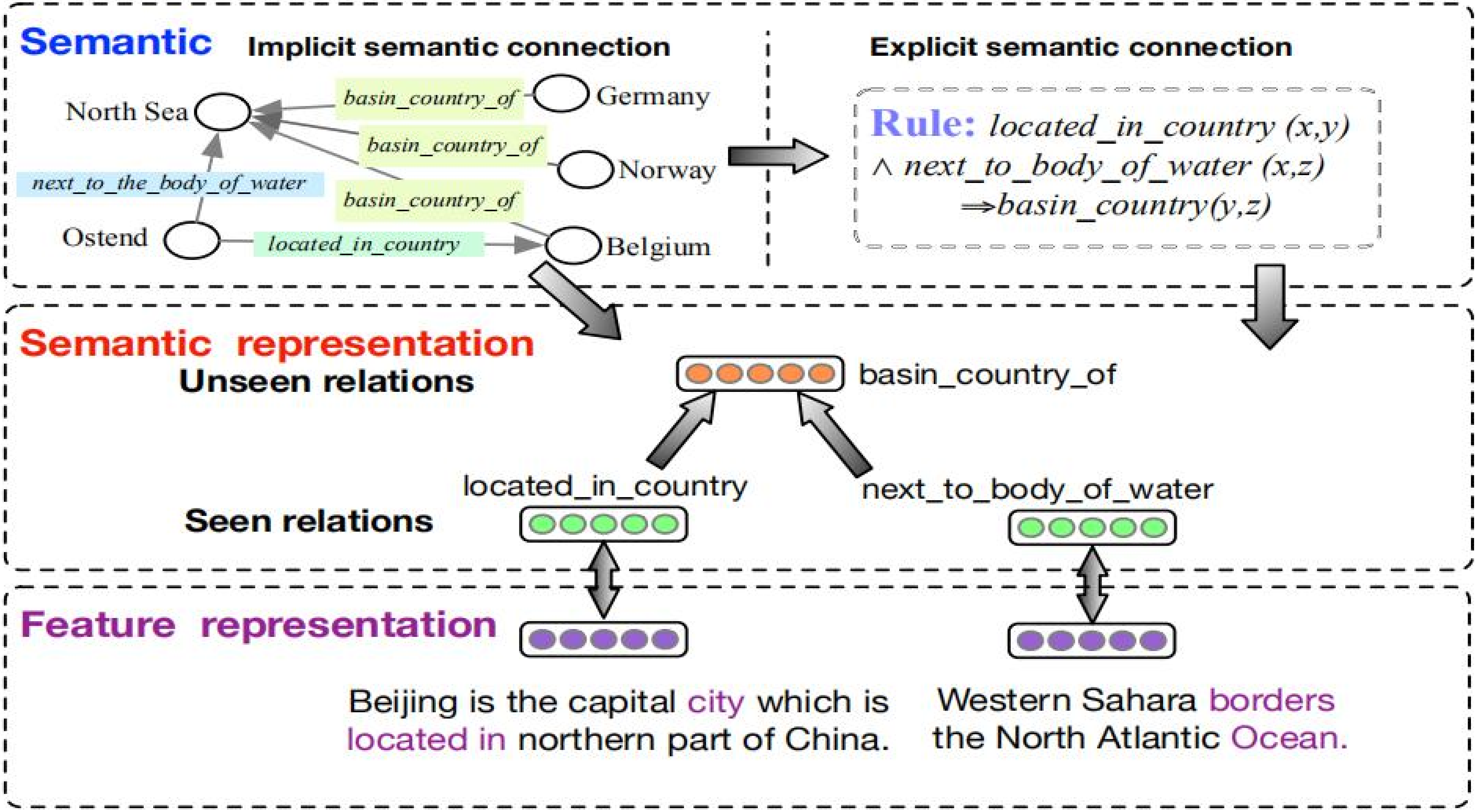


Figure 1: Knowledge graph embedding and rule learning for zero-shot relation classification.

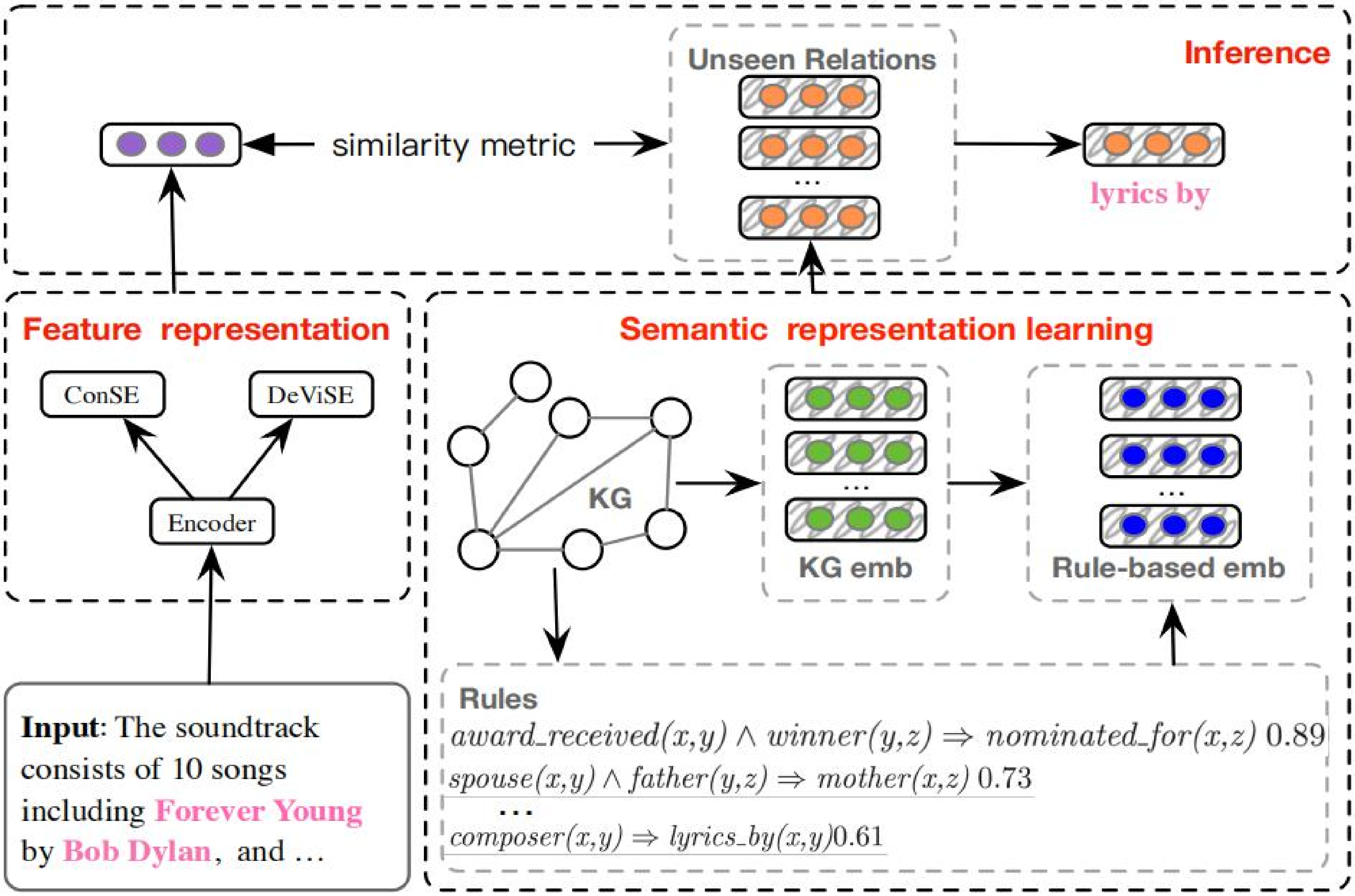
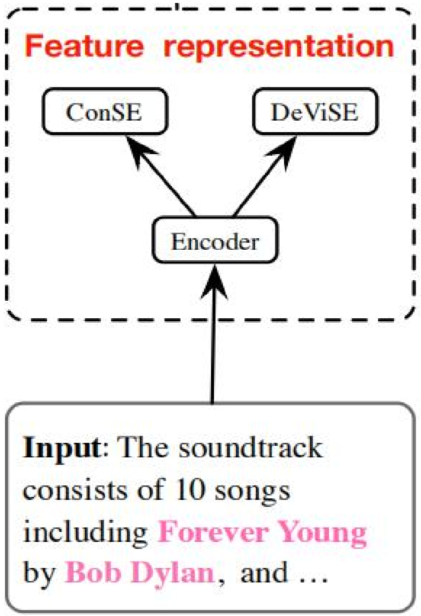


Figure 2: The architecture of Logic-guided Semantic Representation Learning model.



3.2 Feature Representation

The input of feature representation is a sentence, and the output is its vector representation. Firstly, we use the Piecewise Convolutional Neural Networks(PCNNs) (Zeng et al., 2015) model to encode input instance, and then use two types of projection functions including *DeViSE* and *ConSE* to get the final feature representation of the input instance.

$$f = PCNN(x_1, ..., x_n)$$

$$g = W * f + b$$

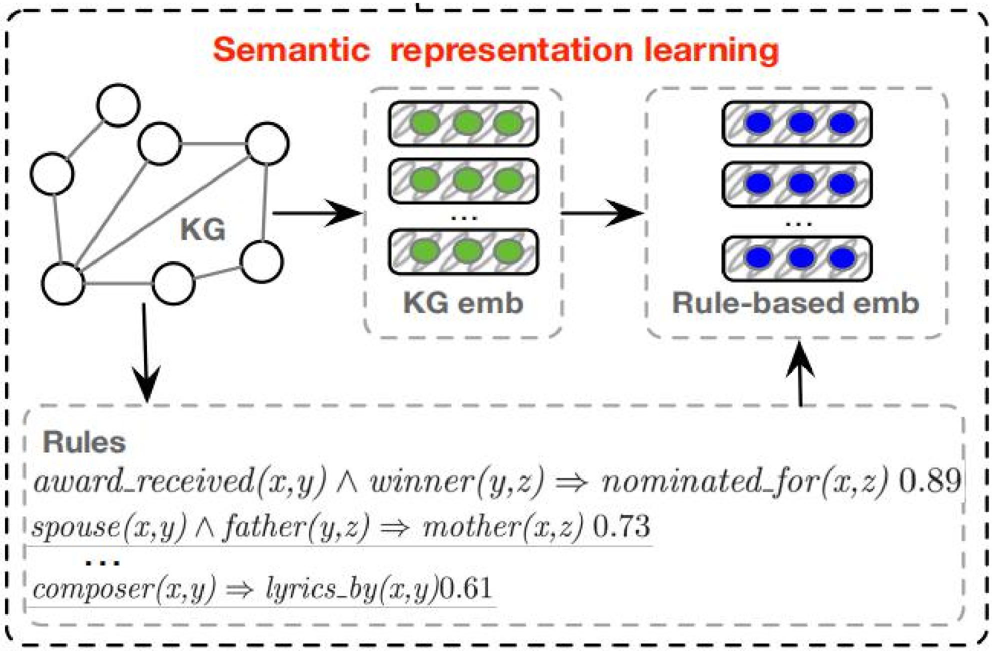
DeViSE

$$R_t^S, p_t, E(R_t^S) = C(f), t = 1, ..., T$$

ConSE

$$g = \sum_{t=1}^{T} p_t * E(R_t^S)$$

Feature Representation



$$E_{rl}(R_i^U) = \frac{\sum_{j=1}^{K} conf_j * E_{kg}(Rule_{ij}^U)}{\sum_{j=1}^{K} conf_j}$$

Rule Embedding

$$E_{kr} = \lambda * E_{rl} + (1 - \lambda) * E_{kg}$$

Rule+Word Embedding

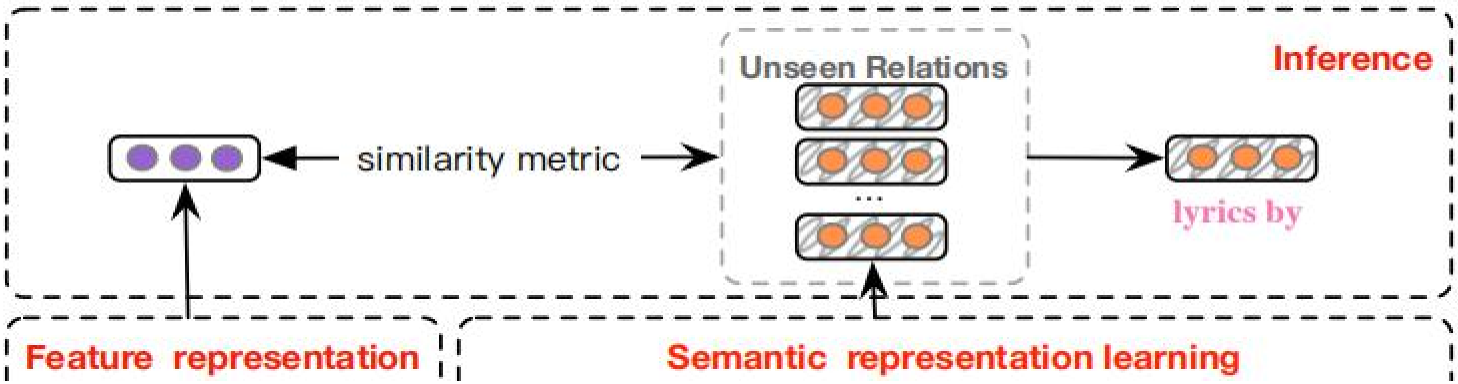
Word EmbeddingKG Embedding

KG+Word embedding

$$E_{kw} = W_2 * ([E_{kg}; E_{wd}] + b_2)$$

3.3 Semantic Representation Learning

Semantic representation builds connections between unseen and seen relations in ZSRC via external resources. We describe the following three kinds of embedding representations in a semantic space.



$$\overline{y_i} = sim(f_{x_i}, E(R^U_{x_i}))$$

Dataset

We construct a new dataset based upon Wikipedia-Wikidata (Sorokin and Gurevych, 2017) relation extraction dataset which contains 353 relations and 856,217 instances. To evaluate the capability of injecting rule logic into the zero-shot prediction models, we ensure that relations have certain connections in our dataset.

Results

	ConSE(Hit@n)			DeViSE(Hit@n)		
	1	2	5	1	2	5
$+E_{wd}$	0.21	0.30	0.43	0.11	0.19	0.39
$+E_{kg}$	0.39	0.53	0.69	0.22	0.38	0.57
$+E_{rl}$	0.40	0.54	0.72	0.23	0.39	0.58
$+E_{kw}$	0.39	0.55	0.72	0.23	**0.40**	**0.59**
$+E_{rw}$	0.40	0.55	0.70	0.23	0.34	0.57
$+E_{kr}$	**0.43**	**0.57**	**0.74**	**0.25**	0.39	**0.59**

Table 2: Performance of DeViSE and ConSE in the case of different embedding methods, including Word(E_{wd}), KG(E_{kg}), Rule(E_{rl}), KG+Word(E_{kw}), Rule+Word(E_{rw}) and KG+Rule(E_{kr}) embeddings.

Results

<table>
<tr><th rowspan="2">Unseen Relations</th><th colspan="2">F1-score</th><th colspan="2">Top 3 Related Seen Relations</th></tr>
<tr><th>$+E_{kg}$</th><th>$+E_{wd}$</th><th>$+E_{kg}$</th><th>$+E_{wd}$</th></tr>
<tr><td rowspan="3">lyrics_by</td><td rowspan="3">0.52</td><td rowspan="3">0.06</td><td>performer</td><td>influenced_by</td></tr>
<tr><td>composer</td><td>spouse</td></tr>
<tr><td>cast_member</td><td>cast_member</td></tr>
<tr><td rowspan="3">after_a_work_by</td><td rowspan="3">0.51</td><td rowspan="3">0.01</td><td>author</td><td>named_after</td></tr>
<tr><td>screenwriter</td><td>author</td></tr>
<tr><td>creator</td><td>characters</td></tr>
<tr><td rowspan="3">location_of_formation</td><td rowspan="3">0.46</td><td rowspan="3">0.02</td><td>headquarters_location</td><td>subclass_of</td></tr>
<tr><td>location</td><td>opposite_of</td></tr>
<tr><td>capital</td><td>part_of</td></tr>
<tr><td rowspan="3">nominated_for</td><td rowspan="3">0.97</td><td rowspan="3">0.56</td><td>award_received</td><td>award_received</td></tr>
<tr><td>winner</td><td>part_of</td></tr>
<tr><td>participant_of</td><td>member_of</td></tr>
<tr><td rowspan="3">mother</td><td rowspan="3">0.40</td><td rowspan="3">0.83</td><td>follows</td><td>child</td></tr>
<tr><td>spouse</td><td>spouse</td></tr>
<tr><td>twinned_administrative_body</td><td>father</td></tr>
<tr><td rowspan="3">developer</td><td rowspan="3">0.38</td><td rowspan="3">0.49</td><td>publisher</td><td>manufacturer</td></tr>
<tr><td>manufacturer</td><td>publisher</td></tr>
<tr><td>owned_by</td><td>owned_by</td></tr>
<tr><td rowspan="3">office_contested</td><td rowspan="3">0.26</td><td rowspan="3">0.00</td><td>position_held</td><td rowspan="3">———</td></tr>
<tr><td>successful_candidate</td></tr>
<tr><td>applies_to_jurisdiction</td></tr>
<tr><td rowspan="3">occupant</td><td rowspan="3">0.31</td><td rowspan="3">0.00</td><td>owned_by</td><td rowspan="3">———</td></tr>
<tr><td>location</td></tr>
<tr><td>headquarters_location</td></tr>
<tr><td rowspan="3">drafted_by</td><td rowspan="3">0.81</td><td rowspan="3">0.00</td><td>member_of_sports_team</td><td rowspan="3">———</td></tr>
<tr><td>educated_at</td></tr>
<tr><td>member_of</td></tr>
</table>

Table 3: Results of KG embedding and word embedding on F1 score when using ConSE as projection function. And top 3 most influential seen relations of the corresponding unseen relation are presented.

Unseen Relations	F1-score						Related rules w.r.t. unseen relations
	$+E_{wd}$	$+E_{kg}$	$+E_{rl}$	$+E_{kw}$	$+E_{rw}$	$+E_{kr}$	
mother	**0.83**	0.40	0.77	0.53	0.80	0.78	*mother(x,z) ⇐ spouse(x,y) ∧ father(y,z)* *mother(x,y) ⇐ child(y,x)*
lyrics_by	0.06	0.52	0.51	0.49	0.48	**0.52**	*lyrics_by(x,y) ⇐ composer(x,y)*
nominated_for	0.56	**0.97**	0.96	**0.97**	0.96	0.96	*nominated_for(x,z) ⇐ award_received(x,y) ∧ winner(y,z)*
producer	0.41	0.52	**0.55**	0.54	0.52	0.53	*producer(x,y) ⇐ director(x,y)* *producer(x,y) ⇐ screenwriter(x,y)* *producer(x,y) ⇐ cast_member(x,y)*
field_of_work	0.04	0.14	0.29	0.11	0.29	**0.37**	*field_of_work(x,y) ⇐ occupation(x,y)*
connecting_line	0.00	0.10	0.43	0.28	0.42	**0.47**	*connecting_line(x,z) ⇐ adjacent_station(y,x) ∧ part of(y,z)*
residence	0.01	0.32	0.30	0.30	0.38	**0.39**	*residence(x,y) ⇐ place_of_birth(x,y)* *residence(x,y) ⇐ place_of_death(x,y)*

Table 4: Results of all different embeddings on F1 score when regrading ConSE as project funtion, and related rules w.r.t unseen relations.

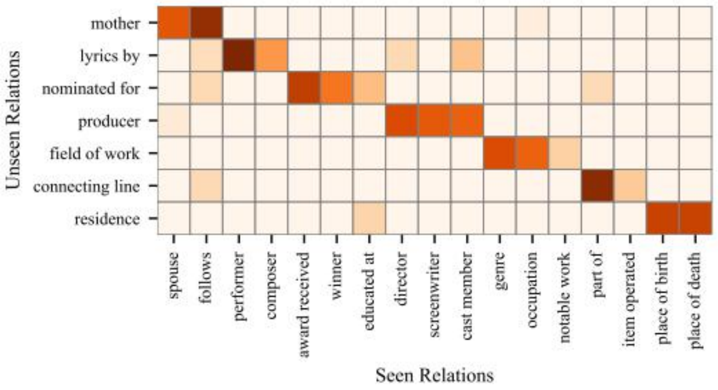


Figure 3: This heatmap is constructed from the result of ConSE+KG, reflecting the incidence of seen relations on unseen relations. Where the horizontal axis represents seen classes and the vertical axis represents unseen classes.

LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding

Yang Xu[1]*, Yiheng Xu[2]*, Tengchao Lv[2]*, Lei Cui[2], Furu Wei[2], Guoxin Wang[3], Yijuan Lu[3], Dinei Florencio[3], Cha Zhang[3], Wanxiang Che[1], Min Zhang[4], Lidong Zhou[2]
[1]Harbin Institute of Technology
[2]Microsoft Research Asia
[3]Microsoft Cloud&AI Team
[4]Soochow University
{yxu,car}@ir.hit.edu.cn
{v-yixu,v-telv,lecu,fuwei,lidongz}@microsoft.com
{guow,yijlu,dinei,chazhang}@microsoft.com
minzhang@suda.edu.cn

——微软亚洲研究院(KDD2020)

The contributions of this paper:

- This paper presents an improved version of LayoutLM (Xu et al., 2020), aka LayoutLMv2.
- Extending the existing research work, they propose new model architectures and pre-training objectives in the LayoutLMv2 model.
- https://github.com/microsoft/unilm/tree/master/layoutlm

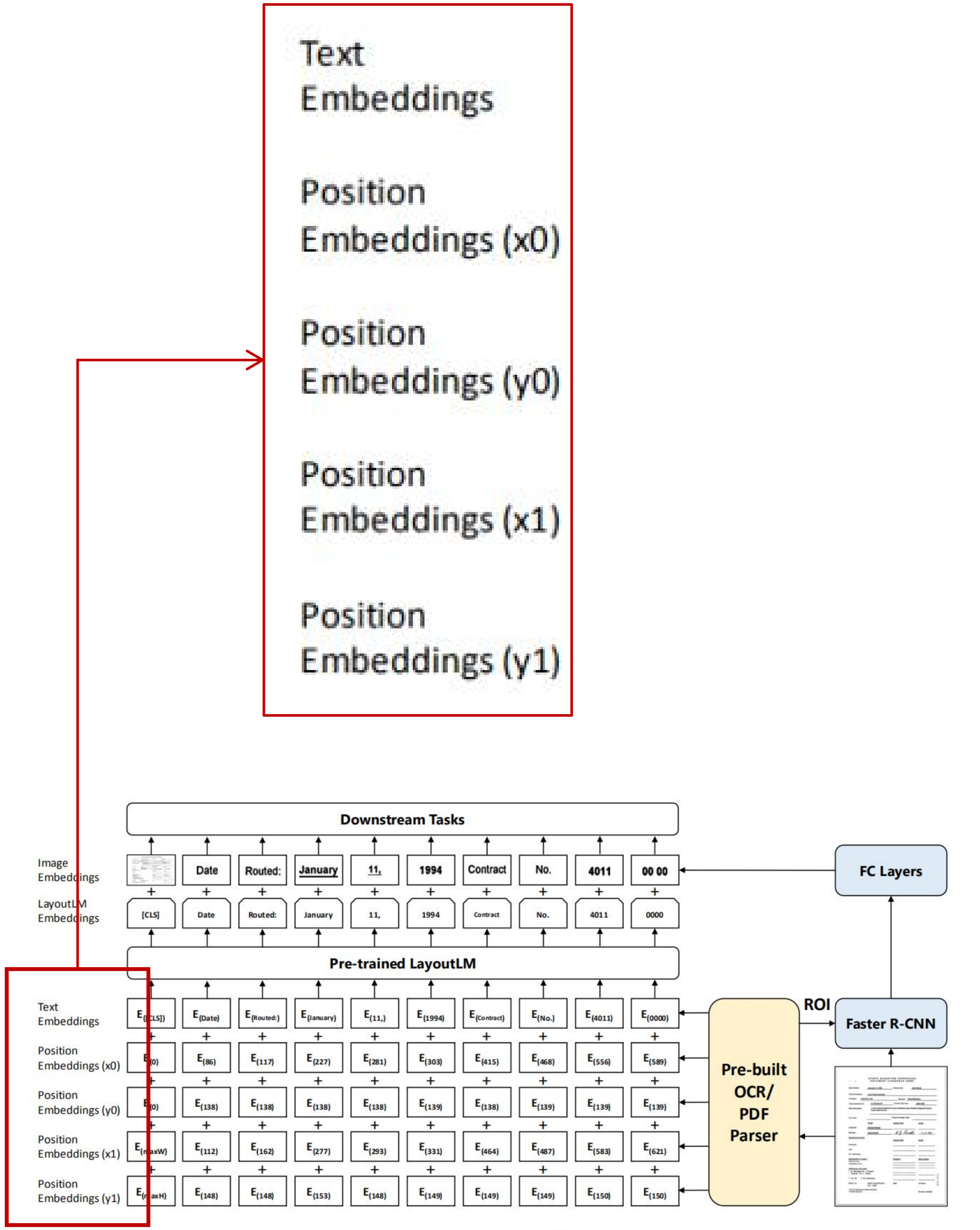


Figure 2: An example of LayoutLM, where 2-D layout and image embeddings are integrated into the original BERT architecture. The LayoutLM embeddings and image embeddings from Faster R-CNN work together for downstream tasks.

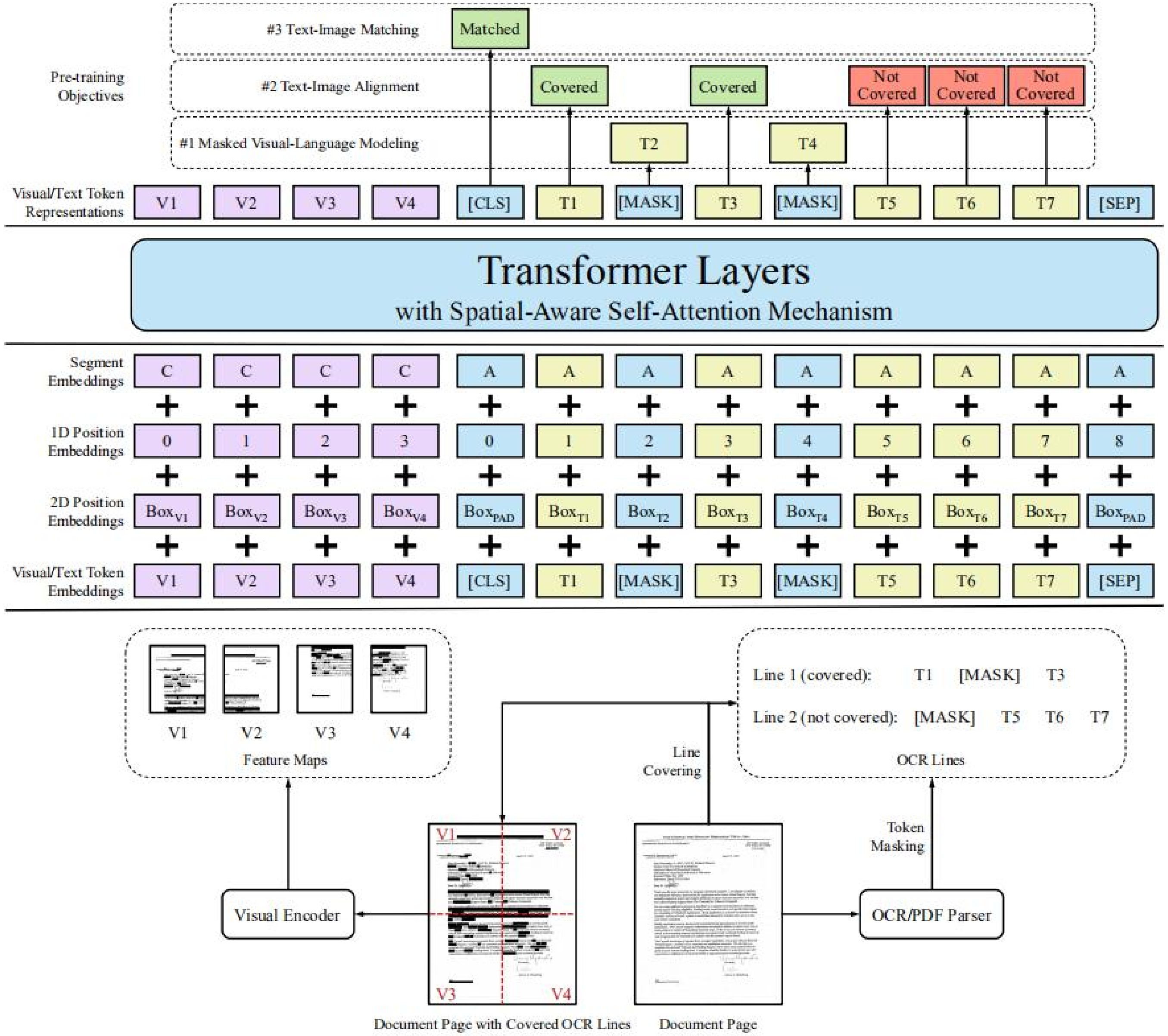
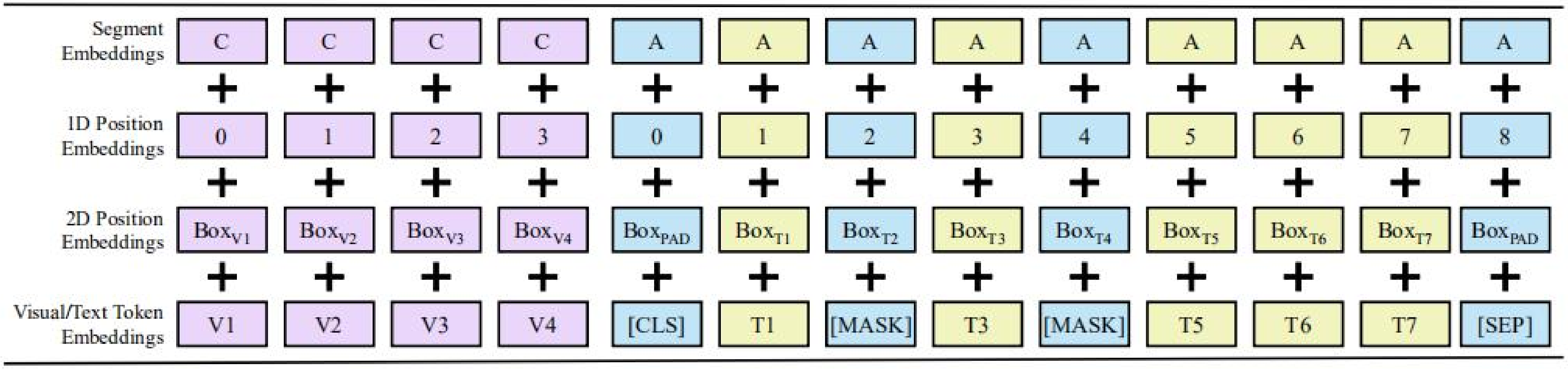


Figure 2: An illustration of the model architecture and pre-training strategies for LayoutLMv2



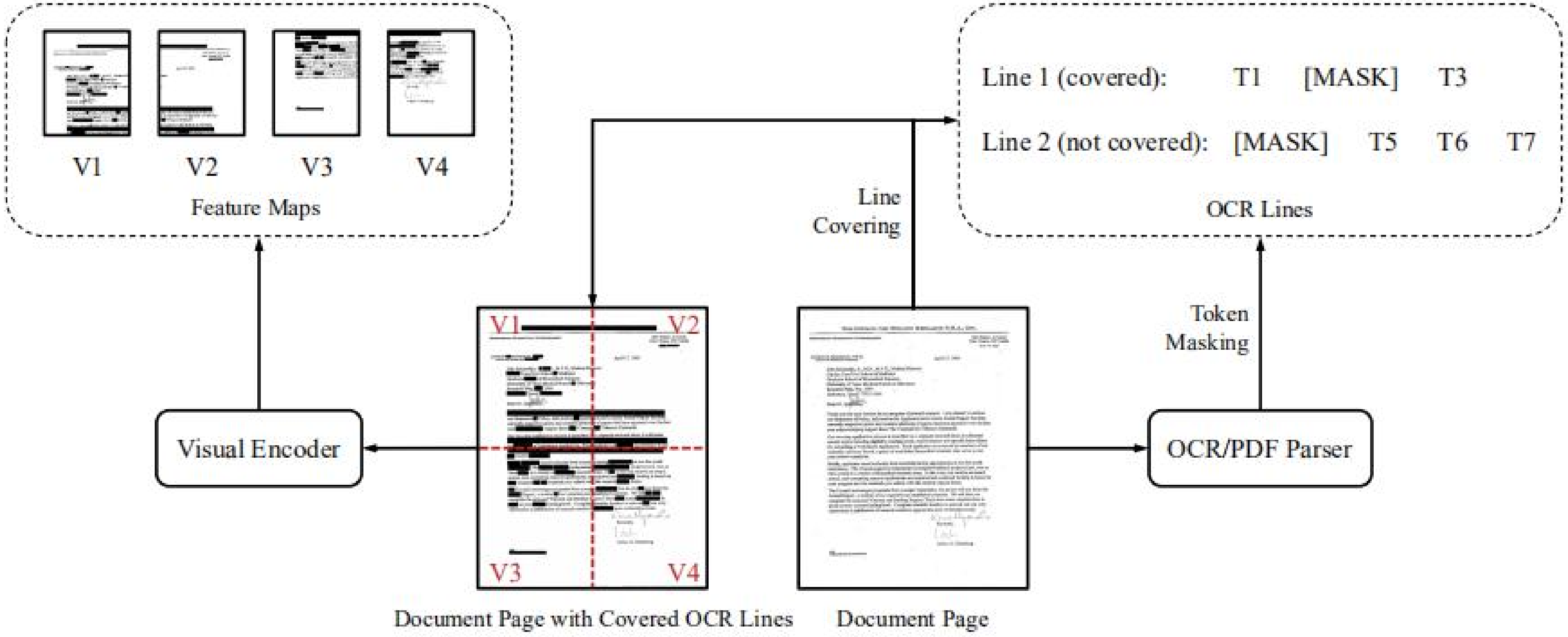
Text Embedding We recognize text and serialize it in a reasonable reading order using off-the-shelf OCR tools and PDF parsers. Following the common practice, we use WordPiece (Wu et al.,

$$S = \{\texttt{[CLS]}, w_1, w_2, ..., \texttt{[SEP]}, \texttt{[PAD]}, \texttt{[PAD]}, ...\}, |S| = L$$

$$\mathbf{t}_i = \mathrm{TokEmb}(w_i) + \mathrm{PosEmb1D}(i) + \mathrm{SegEmb}(s_i), 0 \leq i < L$$

Layout Embedding The layout embedding layer aims to embed the spatial layout information represented by token bounding boxes in which corner coordinates and box shapes are identified explicitly. Following the vanilla LayoutLM, we normalize and discretize all coordinates to integers in the range $[0, 1000]$, and use two embedding layers to embed x-axis features and y-axis features sepa-

$$\mathbf{l}_i = \mathrm{Concat}\big(\mathrm{PosEmb2D_x}(x_0, x_1, w), \mathrm{PosEmb2D_y}(y_0, y_1, h)\big), 0 \leq i < WH + L$$



Visual Embedding We use ResNeXt-FPN (Xie et al., 2016; Lin et al., 2017) architecture as the backbone of the visual encoder. Given a document page image I, it is resized to 224×224 then fed

$$\mathbf{v}_i = \mathrm{Proj}\big(\mathrm{VisTokEmb}(I)_i\big) + \mathrm{PosEmb1D}(i) + \mathrm{SegEmb}(\texttt{[C]}),\ 0 \leq i < WH$$

Transformer Layers

with Spatial-Aware Self-Attention Mechanism

基于空间感知自注意力机制的多模态编码器

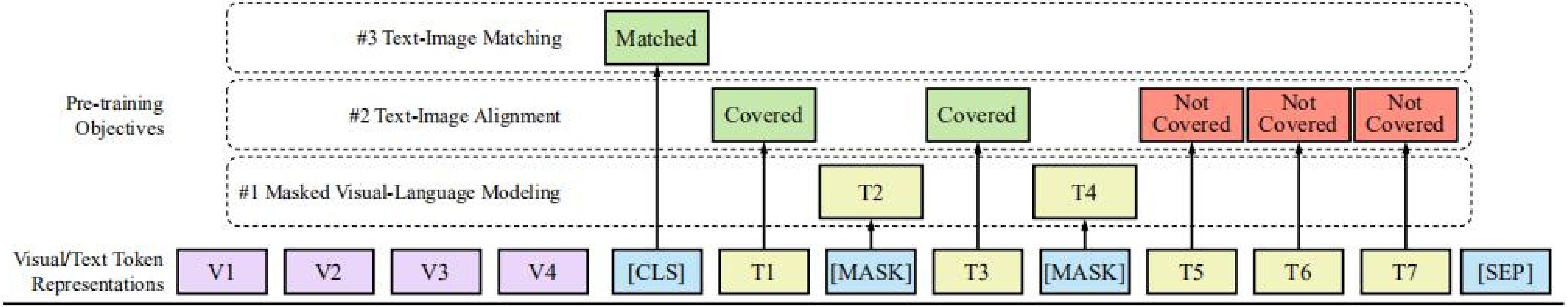
Multi-modal Encoder with Spatial-Aware Self-Attention Mechanism The encoder concatenates visual embeddings $\{\mathbf{v}_0, ..., \mathbf{v}_{WH-1}\}$ and text embeddings $\{\mathbf{t}_0, ..., \mathbf{t}_{L-1}\}$ to a unified sequence X and fuses spatial information by adding the layout embeddings to get the first layer input $\mathbf{x}^{(0)}$.

$$\mathbf{x}_i^{(0)} = X_i + \mathbf{l}_i \text{ , where } X = \{\mathbf{v}_0, ..., \mathbf{v}_{WH-1}, \mathbf{t}_0, ..., \mathbf{t}_{L-1}\}$$

$$\alpha_{ij} = \frac{1}{\sqrt{d_{head}}} \left(\mathbf{x}_i \mathbf{W}^Q\right) \left(\mathbf{x}_j \mathbf{W}^K\right)^\top$$

$$\alpha'_{ij} = \alpha_{ij} + \mathbf{b}^{(1\mathrm{D})}_{j-i} + \mathbf{b}^{(2\mathrm{D_x})}_{x_j - x_i} + \mathbf{b}^{(2\mathrm{D_y})}_{y_j - y_i}$$

$$\mathbf{h}_i = \sum_j \frac{\exp\left(\alpha'_{ij}\right)}{\sum_k \exp\left(\alpha'_{ik}\right)} \mathbf{x}_j \mathbf{W}^V$$



Masked Visual-Language Modeling Similar to the vanilla LayoutLM, we use the Masked Visual-Language Modeling (MVLM) to make the model learn better in the language side with the cross-modality clues.

Text-Image Alignment In addition to the MVLM, we propose the Text-Image Alignment (TIA) as a fine-grained cross-modality alignment task.

Text-Image Matching Furthermore, a coarse-grained cross-modality alignment task, Text-Image Matching (TIM) is applied during the pre-training stage.

Results

Model	Precision	Recall	F1	#Parameters
$BERT_{BASE}$	0.5469	0.6710	0.6026	110M
$UniLMv2_{BASE}$	0.6349	0.6975	0.6648	125M
$BERT_{LARGE}$	0.6113	0.7085	0.6563	340M
$UniLMv2_{LARGE}$	0.6780	0.7391	0.7072	355M
$LayoutLM_{BASE}$	0.7597	0.8155	0.7866	113M
$LayoutLM_{LARGE}$	0.7596	0.8219	0.7895	343M
$LayoutLMv2_{BASE}$	0.8029	0.8539	0.8276	200M
$LayoutLMv2_{LARGE}$	**0.8324**	**0.8519**	**0.8420**	426M
BROS (Anonymous, 2021)	0.8056	0.8188	0.8121	-

Table 1: Model accuracy (entity-level Precision, Recall, F1) on the FUNSD dataset

Model	Precision	Recall	F1	#Parameters
$BERT_{BASE}$	0.8833	0.9107	0.8968	110M
$UniLMv2_{BASE}$	0.8987	0.9198	0.9092	125M
$BERT_{LARGE}$	0.8886	0.9168	0.9025	340M
$UniLMv2_{LARGE}$	0.9123	0.9289	0.9205	355M
$LayoutLM_{BASE}$	0.9437	0.9508	0.9472	113M
$LayoutLM_{LARGE}$	0.9432	0.9554	0.9493	343M
$LayoutLMv2_{BASE}$	0.9453	0.9539	0.9495	200M
$LayoutLMv2_{LARGE}$	**0.9565**	**0.9637**	**0.9601**	426M
SPADE (Hwang et al., 2020)	-	-	0.9150	-
BROS (Anonymous, 2021)	0.9558	0.9514	0.9536	-

Table 2: Model accuracy (entity-level Precision, Recall, F1) on the CORD dataset

Model	Precision	Recall	F1	#Parameters
$BERT_{BASE}$	0.9099	0.9099	0.9099	110M
$UniLMv2_{BASE}$	0.9459	0.9459	0.9459	125M
$BERT_{LARGE}$	0.9200	0.9200	0.9200	340M
$UniLMv2_{LARGE}$	0.9488	0.9488	0.9488	355M
$LayoutLM_{BASE}$	0.9438	0.9438	0.9438	113M
$LayoutLM_{LARGE}$	0.9524	0.9524	0.9524	343M
$LayoutLMv2_{BASE}$	0.9625	0.9625	0.9625	200M
$LayoutLMv2_{LARGE}$	0.9661	0.9661	0.9661	426M
$LayoutLMv2_{LARGE}$ (Excluding OCR mismatch)	**0.9904**	**0.9661**	**0.9781**	426M
BROS (Anonymous, 2021)	0.9493	0.9603	0.9548	-
PICK (Yu et al., 2020)	0.9679	0.9546	0.9612	-
TRIE (Zhang et al., 2020)	-	-	0.9618	-
Top-1 on SROIE Leaderboard (Excluding OCR mismatch)[5]	0.9889	0.9647	0.9767	-

Table 3: Model accuracy (entity-level Precision, Recall, F1) on the SROIE dataset (until 2020-12-24)

Model	Fine-tuning set	ANLS	#Parameters
$BERT_{BASE}$	train	0.6354	110M
$UniLMv2_{BASE}$	train	0.7134	125M
$BERT_{LARGE}$	train	0.6768	340M
$UniLMv2_{LARGE}$	train	0.7709	355M
$LayoutLM_{BASE}$	train	0.6979	113M
$LayoutLM_{LARGE}$	train	0.7259	343M
$LayoutLMv2_{BASE}$	train	0.7808	200M
$LayoutLMv2_{LARGE}$	train	0.8348	426M
$LayoutLMv2_{LARGE}$	train + dev	0.8529	426M
$LayoutLMv2_{LARGE}$ + QG	train + dev	**0.8672**	426M
Top-1 on DocVQA Leaderboard (30 models ensemble)[7]	-	0.8506	-

Table 6: Average Normalized Levenshtein Similarity (ANLS) score on the DocVQA dataset (until 2020-12-24), "QG" denotes the data augmentation with the question generation dataset.

Thanks