ERNIE: Enhanced Language Representation with Informative Entities

by Daria Pirozhkova

February 2020

The knowledge graphs (KGs)

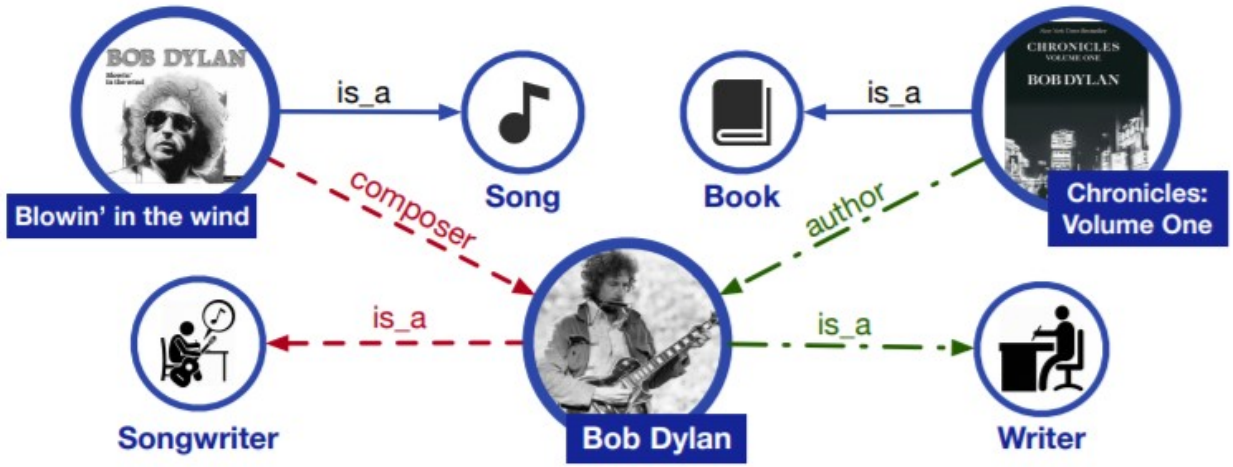

Figure 1: An example of incorporating extra knowledge information for language understanding. The solid lines present the existing knowledge facts. The red dotted lines present the facts extracted from the sentence in red. The green dotdash lines present the facts extracted from the sentence in green.

Considering rich knowledge information can lead to better language understanding and accordingly benefits various knowledge-driven applications, e.g. entity typing and relation classification.

Which problems exist for incorporating external knowledge into language representation models?

- Structured Knowledge Encoding: regarding to the given text, how to effectively extract and encode its related informative facts in KGs for language representation models is an important problem;
- Heterogeneous Information Fusion: the pre-training procedure for language representation is quite different from the knowledge representation procedure, leading to two individual vector spaces.
- To design a special pre-training objective to fuse lexical, syntactic, and knowledge information.

The decision is Enhanced Language Representation with Informative Entities (ERNIE), which pretrains a language representation model on both large-scale textual corpora and KGs.

ERNIE

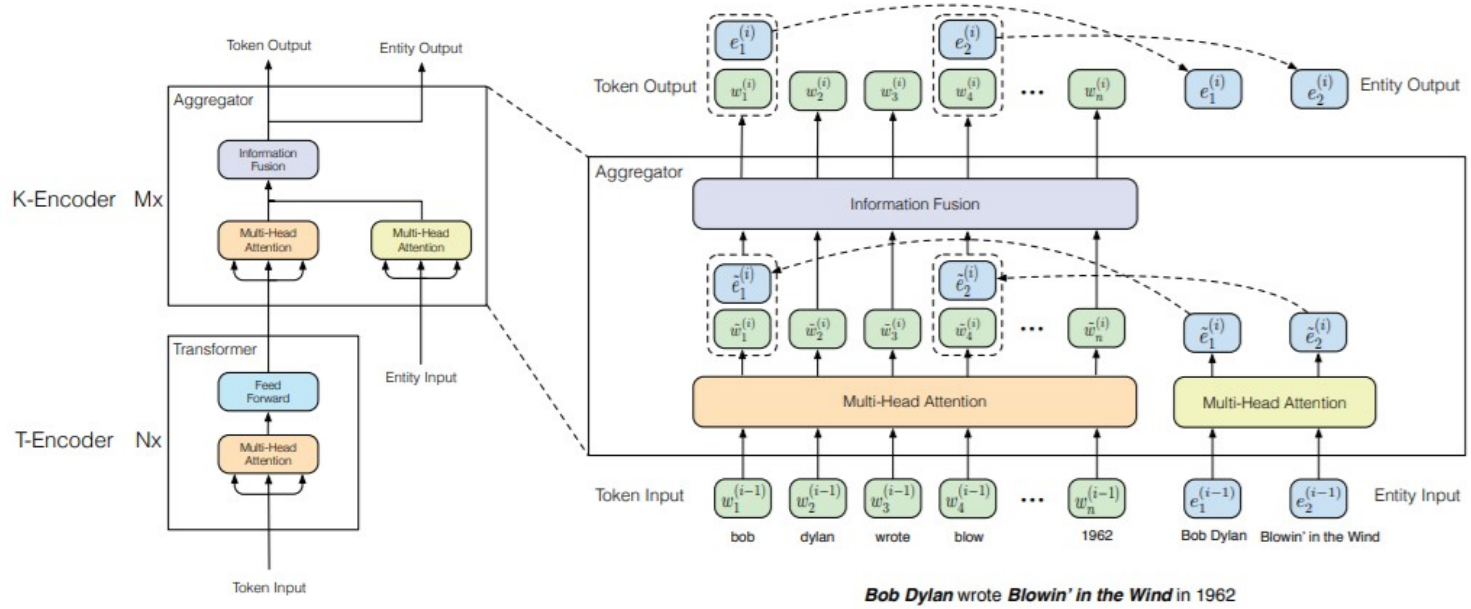

Figure 2: The left part is the architecture of ERNIE. The right part is the aggregator for the mutual integration of the input of tokens and entities. Information fusion layer takes two kinds of input: one is the token embedding, and the other one is the concatenation of the token embedding and entity embedding. After information fusion, it outputs new token embeddings and entity embeddings for the next layer.

"w1,...,wn" - the tokens sequence

"e1,...,em" - the entities sequence

If a token w in V has a corresponding entity e in E, their alignment is defined as f(w) = e.

ERNIE

$$\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\} = \texttt{T-Encoder}(\{w_1, \ldots, w_n\}),$$

where T-Encoder(·) is a multi-layer bidirectional Transformer encoder

After computing w1, . . . , wn, ERNIE adopts a knowledgeable encoder K-Encoder to inject the knowledge information into language representation.

$$\{\boldsymbol{w}_1^o, \ldots, \boldsymbol{w}_n^o\}, \{\boldsymbol{e}_1^o, \ldots, \boldsymbol{e}_n^o\} = \texttt{K-Encoder}(\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}, \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m\}).$$

For a token wj and its aligned entity ek = f(wj), the information fusion process is as follows

$$\begin{aligned}
\boldsymbol{h}_j &= \sigma(\tilde{\boldsymbol{W}}_t^{(i)} \tilde{\boldsymbol{w}}_j^{(i)} + \tilde{\boldsymbol{W}}_e^{(i)} \tilde{\boldsymbol{e}}_k^{(i)} + \tilde{\boldsymbol{b}}^{(i)}) \\
\boldsymbol{w}_j^{(i)} &= \sigma(\boldsymbol{W}_t^{(i)} \boldsymbol{h}_j + \boldsymbol{b}_t^{(i)}), \\
\boldsymbol{e}_k^{(i)} &= \sigma(\boldsymbol{W}_e^{(i)} \boldsymbol{h}_j + \boldsymbol{b}_e^{(i)}).
\end{aligned}$$

where hj is the inner hidden state integrating the information of both the token and the entity. (·) is the non-linear activation function.

Pre-training for Injecting Knowledge

Considering that there are some errors in tokenentity alignments, we perform the following operations for dEA:

- In 5 percent of the time, for a given token-entity alignment, replace the entity with another random entity.
- In 15 percent of the time, mask token-entity alignments, which aims to train our model to correct the errors.
- In the rest of the time, keep tokenentity alignments unchanged, which aims to encourage our model to integrate the entity information into token representations for better language understanding.

Similar to BERT, ERNIE also adopts the masked language model (MLM) and the next sentence prediction (NSP) as pre-training tasks to enable ERNIE to capture lexical and syntactic information from tokens in text.

Fine-tuning for Specific Tasks

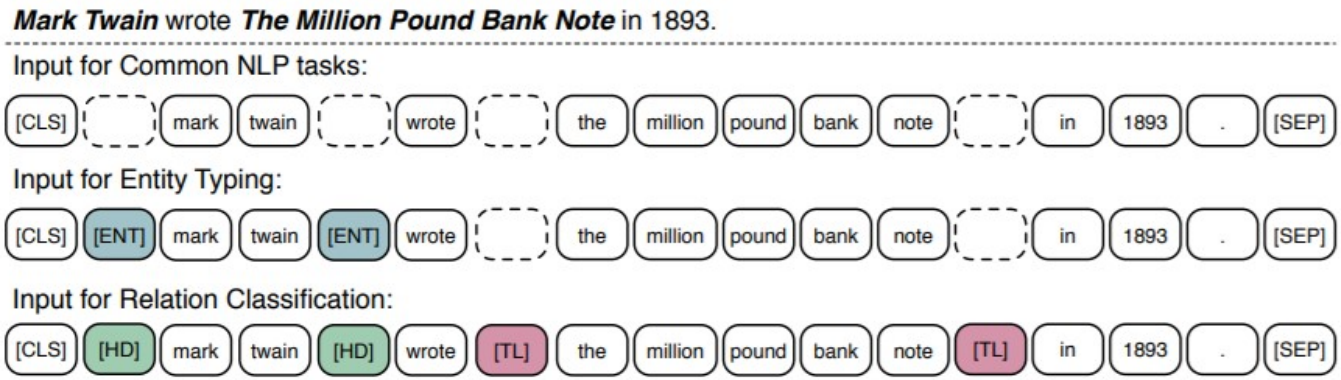

Figure 3: Modifying the input sequence for the specific tasks. To align tokens among different types of input, we use dotted rectangles as placeholder. The colorful rectangles present the specific mark tokens.

Entity Typing

The training set of FIGER is labeled with distant supervision, and its test set is annotated by human. Open Entity is a completely manually-annotated dataset.

Dataset	Train	Develop	Test	Type
FIGER	2,000,000	10,000	563	113
Open Entity	2,000	2,000	2,000	6

Table 1: The statistics of the entity typing datasets FIGER and Open Entity.

Model	Acc.	Macro	Micro
NFGEC (Attentive)	54.53	74.76	71.58
NFGEC (LSTM)	55.60	75.15	71.73
BERT	52.04	75.16	71.63
ERNIE	**57.19**	**76.51**	**73.39**

Table 2: Results of various models on FIGER (%).

Model	P	R	F1
NFGEC (LSTM)	68.80	53.30	60.10
UFET	77.40	60.60	68.00
BERT	76.37	70.96	73.56
ERNIE	**78.42**	**72.90**	**75.56**

Table 3: Results of various models on Open Entity (%).

Relation Classification

Dataset	Train	Develop	Test	Relation
FewRel	8,000	16,000	16,000	80
TACRED	68,124	22,631	15,509	42

Table 4: The statistics of the relation classification datasets FewRel and TACRED.

Model	FewRel			TACRED		
	P	R	F1	P	R	F1
CNN	69.51	69.64	69.35	70.30	54.20	61.20
PA-LSTM	-	-	-	65.70	64.50	65.10
C-GCN	-	-	-	69.90	63.30	66.40
BERT	85.05	85.11	84.89	67.23	64.81	66.00
ERNIE	88.49	88.44	**88.32**	69.97	66.08	**67.97**

Table 5: Results of various models on FewRel and TACRED (%).

GLUE

The General Language Understanding Evaluation (GLUE) benchmark is a collection of diverse natural language understanding tasks.

Model	MNLI-(m/mm) 392k	QQP 363k	QNLI 104k	SST-2 67k
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	-	93.5
ERNIE	84.0/83.2	71.2	91.3	93.5
Model	**CoLA 8.5k**	**STS-B 5.7k**	**MRPC 3.5k**	**RTE 2.5k**
$\text{BERT}_{\text{BASE}}$	52.1	85.8	88.9	66.4
ERNIE	52.3	83.2	88.2	68.8

Table 6: Results of BERT and ERNIE on different tasks of GLUE (%).

The End