ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, Haifeng Wang
Baidu Inc., Beijing, China
{sunyu02, wangshuohuan, tianhao, wu_hua, wanghaifeng}@baidu.com

Abstract

Recently pre-trained models have achieved state-of-the-art results in various language understanding tasks. Current pre-training procedures usually focus on training the model with several simple tasks to grasp the co-occurrence of words or sentences. However, besides co-occurring information, there exists other valuable lexical, syntactic and semantic information in training corpora, such as named entities, semantic closeness and discourse relations. In order to extract the lexical, syntactic and semantic information from training corpora, we propose a continual pre-training framework named ERNIE 2.0 which incrementally builds pre-training tasks and then learn pre-trained models on these constructed tasks via continual multi-task learning. Based on this framework, we construct several tasks and train the ERNIE 2.0 model to capture lexical, syntactic and semantic aspects of information in the training data. Experimental results demonstrate that ERNIE 2.0 model outperforms BERT and XLNet on 16 tasks including English tasks on GLUE benchmarks and several similar tasks in Chinese. The source codes and pre-trained models have been released at https://github.com/PaddlePaddle/ERNIE.

Introduction

Pre-trained language representations such as ELMo(Peters et al. 2018), OpenAI GPT(Radford et al. 2018), BERT (Devlin et al. 2018), ERNIE 1.0 (Sun et al. 2019)[1] and XLNet(Yang et al. 2019) have been proven to be effective for improving the performances of various natural language understanding tasks including sentiment classification (Socher et al. 2013), natural language inference (Bowman et al. 2015), named entity recognition (Sang and De Meulder 2003) and so on.

Generally the pre-training of models often train the model based on the co-occurrence of words and sentences. While in fact, there are other lexical, syntactic and semantic information worth examining in training corpora other than co-occurrence. For example, named entities like person names, location names, and organization names, may contain conceptual information. Information like sentence order and sentence proximity enables the models to learn structure-aware representations. And semantic similarity at the document level or discourse relations among sentences allow the models to learn semantic-aware representations. In order to discover all valuable information in training corpora, be it lexical, syntactic or semantic representations, we propose a continual pre-training framework named ERNIE 2.0 which could incrementally build and train a large variety of pre-training tasks through continual multi-task learning.

Our ERNIE framework supports the introduction of various customized tasks continually, which is realized through continual multi-task learning. When given one or more new tasks, the continual multi-task learning method simultaneously trains the newly-introduced tasks together with the original tasks in an efficient way, without forgetting previously learned knowledge. In this way, our framework can incrementally train the distributed representations based on the previously trained parameters that it grasped. Moreover, in this framework, all the tasks share the same encoding networks, thus making the encoding of lexical, syntactic and semantic information across different tasks possible.

In summary, our contributions are as follows:

- We propose a continual pre-training framework ERNIE 2.0, which efficiently supports customized training tasks and continual multi-task learning in an incremental way.
- We construct three kinds of unsupervised language processing tasks to verify the effectiveness of the proposed framework. Experimental results demonstrate that ERNIE 2.0 achieves significant improvements over BERT and XLNet on 16 tasks including English GLUE benchmarks and several Chinese tasks.
- Our fine-tuning code of ERNIE 2.0 and models pre-trained on English corpora are available at https://github.com/PaddlePaddle/ERNIE.

Related Work

Unsupervised Learning for Language Representation

It is effective to learn general language representation by pre-training a language model with a large amount of unan-

[1] In order to distinguish ERNIE 2.0 framework and the ERNIE model, the latter is referred to as ERNIE 1.0.(Sun et al. 2019)

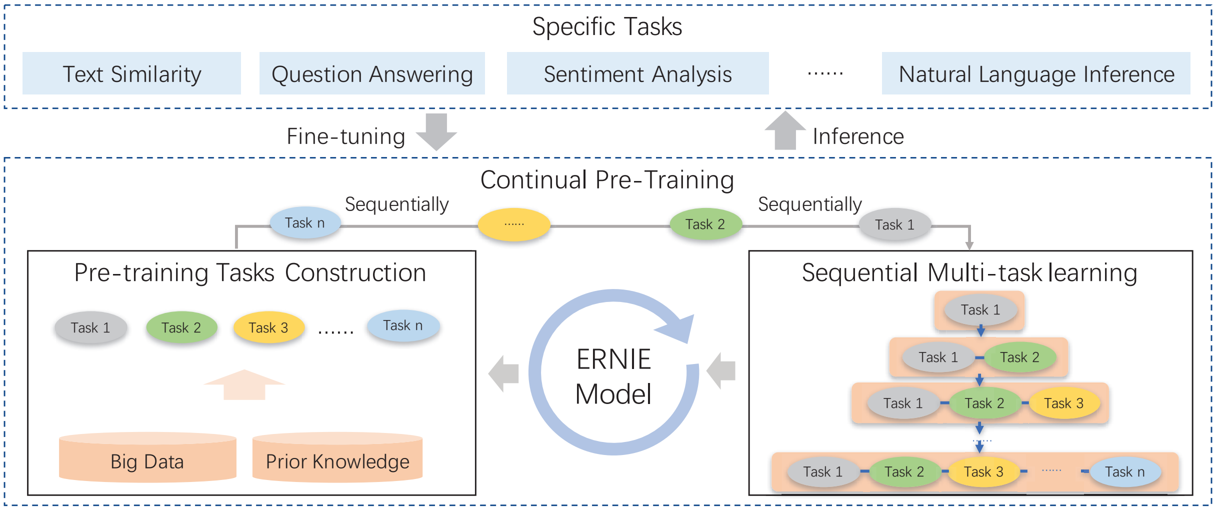

Figure 1: The framework of ERNIE 2.0, where the pre-training tasks can be incrementally constructed, the models are pre-trained through continual multi-task learning, and the pre-trained model is fine-tuned to adapt to various language understanding tasks.

notated data. Traditional methods usually focus on context-independent word embedding. Methods such as Word2Vec (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014) learn fixed word embeddings based on word co-occurring information on large corpora.

Recently, several studies centered on contextualized language representations have been proposed and context-dependent language representations have shown state-of-the-art results in various natural language processing tasks. ELMo (Peters et al. 2018) proposes to extract context-sensitive features from a language model. OpenAI GPT (Radford et al. 2018) enhances the context-sensitive embedding by adjusting the Transformer (Vaswani et al. 2017). BERT (Devlin et al. 2018), however, adopts a masked language model while adding a next sentence prediction task into the pre-training. XLM (Lample and Conneau 2019) integrates two methods to learn cross-lingual language models, namely the unsupervised method that relies only on monolingual data and the supervised method that leverages parallel bilingual data. MT-DNN (Liu et al. 2019) achieves a better result through learning several supervised tasks in GLUE(Wang et al. 2018) together based on the pre-trained model, which eventually leads to improvements on other supervised tasks that are not learned in the stage of multi-task supervised fine-tuning. XLNet (Yang et al. 2019) uses Transformer-XL (Dai et al. 2019) and proposes a generalized autoregressive pre-training method that learns bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order.

Continual Learning

Continual learning(Parisi et al. 2019; Chen and Liu 2018) aims to train the model with several tasks in sequence so that it remembers the previously-learned tasks when learning the new ones. These methods are inspired by the learning process of humans, as humans are capable of continuously accumulating the information acquired by study or experience to efficiently develop new skills. With continual learning, the model should be able to performs well on new tasks thanks to the knowledge acquired during previous training.

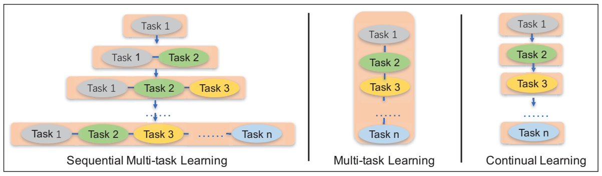

Figure 2: The different methods of continual pre-training.

The ERNIE 2.0 Framework

As shown in Figure 1, the ERNIE 2.0 framework is built based on an widely-used architecture of pre-training and fine-tuning. ERNIE 2.0 differs from the previous pre-training ones in that, instead of training with a small number of pre-training objectives, it could constantly introduce a large variety of pre-training tasks to help the model efficiently learn the lexical, syntactic and semantic representations. Based on this, ERNIE 2.0 framework keeps updating the pre-trained model with continual multi-task learning. During fine-tuning, the ERNIE model is first initialized with the pre-trained parameters, and would be later fine-tuned using data from specific tasks.

Continual Pre-training

The process of continual pre-training contains two steps. Firstly, We continually construct unsupervised pre-training tasks with big data and prior knowledge involved. Secondly, We incrementally update the ERNIE model via continual multi-task learning.

Pre-training Tasks Construction We can construct different kinds of tasks at each time, including word-aware tasks,

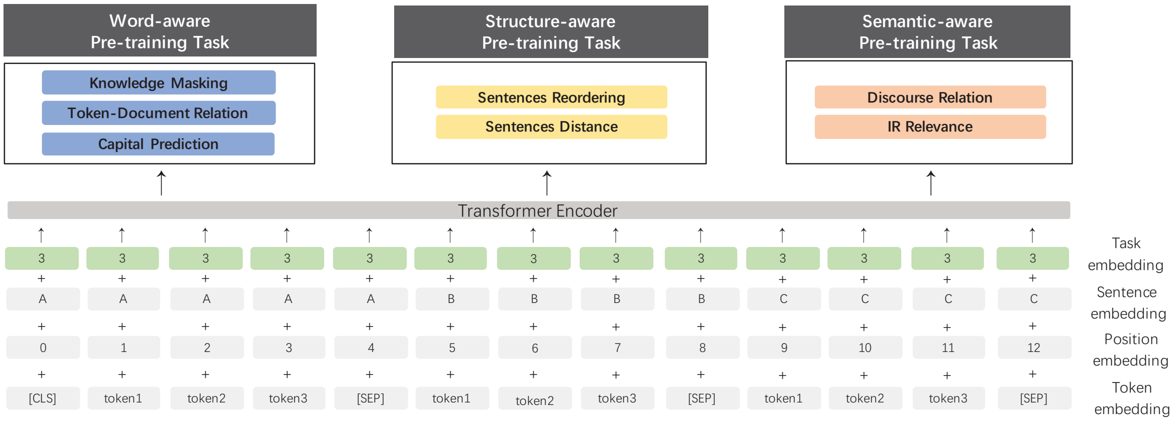

Figure 3: The structure of the ERNIE 2.0 model. The input embedding contains the token embedding, the sentence embedding, the position embedding and the task embedding. Seven pre-training tasks belonging to different kinds are constructed in the ERNIE 2.0 model.

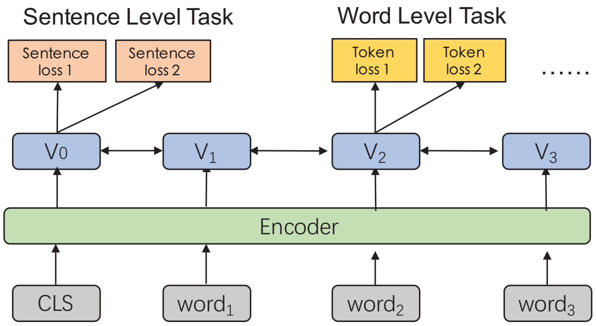

Figure 4: The architecture of multi-task learning in the ERNIE 2.0 framework, in which the encoder can be recurrent neural networks or a deep transformer.

structure-aware tasks and semantic-aware tasks[2]. All of these pre-training tasks rely on self-supervised or weak-supervised signals that could be obtained from massive data without human annotation. Prior knowledge such as named entities, phrases and discourse relations is used to generate labels from large-scale data.

Continual Multi-task Learning The ERNIE 2.0 framework aims to learn lexical, syntactic and semantic information from a number of different tasks. Thus there are two main challenges to overcome. The first is how to train the tasks in a continual way without forgetting the knowledge learned before. The second is how to pre-train these tasks in an efficient way. We propose a continual multi-task learning method to tackle with these two problems. Whenever a new task comes, the continual multi-task learning method first uses the previously learned parameters to initialize the model, and then train the newly-introduced task together with the original tasks simultaneously. This will make sure that the learned parameters encodes the previously-learned knowledge. One left problem is how to make it trained more efficiently. We solve this problem by allocating each task N training iterations. Our framework needs to automatically assign these N iterations for each task to different stages of training. In this way, we can guarantee the efficiency of our method without forgetting the previously trained knowledge [3].

Figure 2 shows the difference among our method, multi-task learning from scratch and previous continual learning. Although multi-task learning from scratch could train multiple tasks at the same time, it is necessary that all customized pre-training tasks are prepared before the training could proceed. So this method takes as much time as continual learning does, if not more. Traditional continual learning method trains the model with only one task at each stage with the demerit that it may forget the previously learned knowledge.

As shown in Figure 4, the architecture of our continual multi-task learning in each stage contains a series of shared text encoding layers to encode contextual information, which can be customized by using recurrent neural networks or a deep Transformer consisting of stacked self-attention layers(Vaswani et al. 2017). The parameters of the encoder can be updated across all learning tasks. There are two kinds of loss functions in our framework. One is the sentence-level loss and the other one is the token-level loss, which are similar to the loss functions of BERT. Each pre-training task has its own loss function. During pre-training, one sentence-level loss function can be combined with multiple token-level loss functions to continually update the model.

Fine-tuning for Application Tasks

By virtue of fine-tuning with task-specific supervised data, the pre-trained model can be adapted to different language understanding tasks, such as question answering, natural language inference, and semantic similarity. Each downstream

[2] For the detailed information of these tasks, please refer to the next section.

[3] For more details, please refer to Table 7 in the experiment section.

Task Corpus	Token-Level Loss			Sentence-Level Loss			
	Knowledge Masking	Capital Prediction	Token-Document Relation	Sentence Reordering	Sentence Distance	Discourse Relation	IR Relevance
Encyclopedia	✓	✓	✓	✓	✓	×	×
BookCorpus	✓	✓	✓	✓	✓	×	×
News	✓	✓	✓	✓	✓	×	×
Dialog	✓	✓	✓	✓	✓	×	×
IR Relevance Data	×	×	×	×	×	×	✓
Discourse Relation Data	×	×	×	×	×	✓	×

Table 1: The Relationship between pre-training task and pre-training dataset. We use different pre-training dataset to construct different tasks. A type of pre-trained dataset can correspond to multiple pre-training tasks.

task has its own fine-tuned models after being fine-tuned.

ERNIE 2.0 Model

In order to verify the effectiveness of the framework, we construct three different kinds of unsupervised language processing tasks and develop a pre-trained model called ERNIE 2.0 model. In this section we introduce the implementation of the model in the proposed framework.

Model Structure

Transformer Encoder The model uses a multi-layer Transformer(Vaswani et al. 2017) as the basic encoder like other pre-training models such as GPT(Radford et al. 2018), BERT(Devlin et al. 2018) and XLM(Lample and Conneau 2019). The transformer can capture the contextual information for each token in the sequence via self-attention, and generate a sequence of contextual embeddings. Given a sequence, the special classification embedding [CLS] is added to the first place of the sequence. Furthermore, the symbol of [SEP] is added as the separator in the intervals of the segments for the multiple input segment tasks.

Task Embedding The model feeds task embedding to represent the characteristic of different tasks. We represent different tasks with an id ranging from 0 to N. Each task id is assigned to one unique task embedding. The corresponding token, segment, position and task embedding are taken as the input of the model. We can use any task id to initialize our model in the fine-tuning process. The model structure is shown in Figure 3.

Pre-training Tasks

We construct three different kinds of tasks to capture different aspects of information in the training corpora. The word-aware tasks enable the model to capture the lexical information, the structure-aware tasks enable the model capture the syntactic information of the corpus and the semantic-aware tasks aims to learn semantic information.

Word-aware Pre-training Tasks

Knowledge Masking Task ERNIE 1.0(Sun et al. 2019) proposed an effective strategy to enhance representation through knowledge integration. It introduced phrase masking and named entity masking and predicts the whole masked phrases and named entities to help the model learn the dependency information in both local contexts and global contexts. We use this task to train an initial version of the model.

Capitalization Prediction Task Capitalized words usually have certain specific semantic information compared to other words in sentences. The cased model has some advantages in tasks like named entity recognition while the uncased model is more suitable for some other tasks. To combine the advantages of both models, we add a task to predict whether the word is capitalized or not.

Token-Document Relation Prediction Task This task predicts whether the token in a segment appears in other segments of the original document. Empirically, the words that appear in many parts of a document are usually commonly-used words or relevant with the main topics of the document. Therefore, through identifying the frequently-occurring words of a document appearing in the segment, the task can enable the ability of a model to capture the key words of the document to some extent.

Structure-aware Pre-training Tasks

Sentence Reordering Task This task aims to learn the relationships among sentences. During the pre-training process of this task, a given paragraph is randomly split into 1 to m segments and then all of the combinations are shuffled by a random permuted order. We let the pre-trained model to reorganize these permuted segments, modeled as a k-class classification problem where $k = \sum_{n=1}^{m} n!$. Empirically, the sentences reordering task can enable the pre-trained model to learn relationships among sentences in a document.

Sentence Distance Task We also construct a pre-training task to learn the sentence distance using document-level information. This task is modeled as a 3-class classification problem. ”0” represents that the two sentences are adjacent in the same document, ”1” represent that the two sentences are in the same document, but not adjacent, and ”2” represents that the two sentences are from two different documents.

Semantic-aware Pre-training Tasks

Discourse Relation Task Beside the distance task mentioned above, we introduce a task to predict the semantic or rhetorical relation between two sentences. We use the data built by Sileo et.al(Sileo et al. 2019) to train a pre-trained model for English tasks. Following the method in Sileo et.al(Sileo et al. 2019), we also automatically construct a Chinese dataset for pre-training.

IR Relevance Task We build a pre-training task to learn the short text relevance in information retrieval. It is a 3-class classification task which predicts the relationship between a query and a title. We take the query as the first sentence and the title as the second sentence. The search log data from a commercial search engine is used as our pre-training data. There are three kinds of labels in this task. The query and title pairs that are labelled as ” 0” stand for strong relevance, which means that the title is clicked by the users after they input the query. Those labelled as ”1” represent weak relevance, which implies that when the query is input by the users, these titles appear in the search results but failed to be clicked by users. The label ”2” means that the query and title are completely irrelevant and random in terms of semantic information.

Experiments

We compare the performance of ERNIE 2.0 with the state-of-the-art pre-training models. For English tasks, we compare our results with BERT (Devlin et al. 2018) and XLNet (Yang et al. 2019) on GLUE. For Chinese tasks, we compare the results with that of BERT (Devlin et al. 2018) and the previous ERNIE 1.0 (Sun et al. 2019) model on several Chinese datasets. Moreover, we will compare our method with multi-task learning and traditional continual learning.

Corpus Type	English(#tokens)	Chinese(#tokens)
Encyclopedia	2021M	7378M
BookCorpus	805M	-
News	-	1478M
Dialog	4908M	522M
IR Relevance Data	-	4500M
Discourse Relation Data	171M	1110M

Table 2: The size of pre-training datasets.

Pre-training and Implementation

Pre-training Data Similar to that of BERT, some data in the English corpus are crawled from Wikipedia and BookCorpus. Besides we also collect some data from Reddit and use the Discovery data (Sileo et al. 2019) as our discourse relation data. For the Chinese corpus, we collect a variety of data, such as encyclopedia, news, dialogue, information retrieval and discourse relation data from a search engine. The details of the pre-training data are shown in Table 2. The relationship between pre-training task and pre-training dataset is shown in Table 1.

Pre-training Settings To compare with BERT(Devlin et al. 2018), We use the same model settings of transformer as BERT. The base model contains 12 layers, 12 self-attention heads and 768-dimensional of hidden size while the large model contains 24 layers, 16 self-attention heads and 1024-dimensional of hidden size. The model settings of XLNet (Yang et al. 2019) are same as BERT.

Task	*BASE*			*LARGE*		
	Epoch	Learning Rate	Batch Size	Epoch	Learning Rate	Batch Size
CoLA	3	3e-5	64	5	3e-5	32
SST-2	4	2e-5	256	4	2e-5	64
MRPC	4	3e-5	32	4	3e-5	16
STS-B	3	5e-5	128	3	5e-5	128
QQP	3	3e-5	256	3	5e-5	256
MNLI	3	3e-5	512	3	3e-5	256
QNLI	4	2e-5	256	4	2e-5	256
RTE	4	2e-5	4	5	3e-5	16
WNLI	4	2e-5	8	4	2e-5	8

Table 3: The Experiment settings for GLUE dataset

Task	*BASE*			*LARGE*		
	Epoch	Learning Rate	Batch Size	Epoch	Learning Rate	Batch Size
CMRC 2018	2	3e-5	64	2	3e-5	64
DRCD	2	5e-5	64	2	3e-5	64
DuReader	2	5e-5	64	2	2e-5	64
MSRA-NER	6	5e-5	16	6	1e-5	16
XNLI	3	1e-4	512	3	4e-5	512
ChnSentiCorp	10	5e-5	24	10	1e-5	24
LCQMC	3	2e-5	32	3	5e-6	32
BQ Corpus	3	3e-5	64	3	1.5e-5	64
NLPCC-DBQA	3	2e-5	64	3	1e-5	64

Table 4: The Experiment Settings for Chinese datasets

ERNIE 2.0 is trained on 48 NVidia v100 GPU cards for the base model and 64 NVidia v100 GPU cards for the large model in both English and Chinese. The ERNIE 2.0 framework is implemented on PaddlePaddle, which is an end-to-end open source deep learning platform developed by Baidu. We use Adam optimizer that parameters of which are fixed to $\beta_1 = 0.9$, $\beta_2 = 0.98$, with a batch size of 393216 tokens. The learning rate is set as 5e-5 for English model and 1.28e-4 for Chinese model. It is scheduled by decay scheme noam (Vaswani et al. 2017) with warmup over the first 4,000 steps for every pre-training task. By virtue of float16 operations, we manage to accelerate the training and reduce the memory usage of our models. Each of the pre-training tasks is trained until the metrics of pre-training tasks converge.

Fine-tuning Tasks

English Task As a multi-task benchmark and analysis platform for natural language understanding, General Language Understanding Evaluation (GLUE) is usually applied to evaluate the performance of models. We also test the performance of ERNIE 2.0 on GLUE. Specifically, GLUE covers a diverse range of NLP datasets, the details is shown (Wang et al. 2018).

Chinese Tasks We executed extensive experiments on 9 Chinese NLP tasks, including machine reading comprehension, named entity recognition, natural language inference, semantic similarity, sentiment analysis and question answer-

Task(Metrics)	*BASE model*		*LARGE model*				
	Test		Dev			Test	
	BERT	ERNIE 2.0	BERT	XLNet	ERNIE 2.0	BERT	ERNIE 2.0
CoLA (Matthew Corr.)	52.1	**55.2**	60.6	63.6	**65.4**	60.5	**63.5**
SST-2 (Accuracy)	93.5	**95.0**	93.2	95.6	**96.0**	94.9	**95.6**
MRPC (Accurary/F1)	84.8/88.9	**86.1/89.9**	88.0/-	89.2/-	**89.7/-**	85.4/89.3	**87.4/90.2**
STS-B (Pearson Corr./Spearman Corr.)	87.1/85.8	**87.6/86.5**	90.0/-	91.8/-	**92.3/-**	87.6/86.5	**91.2/90.6**
QQP (Accuracy/F1)	89.2/71.2	**89.8/73.2**	91.3/-	91.8/-	**92.5/-**	89.3/72.1	**90.1/73.8**
MNLI-m/mm (Accuracy)	84.6/83.4	**86.1/85.5**	86.6/-	**89.8/-**	89.1/-	86.7/85.9	**88.7/88.8**
QNLI (Accuracy)	90.5	**92.9**	92.3	93.9	**94.3**	92.7	**94.6**
RTE (Accuracy)	66.4	**74.8**	70.4	83.8	**85.2**	70.1	**80.2**
WNLI (Accuracy)	**65.1**	**65.1**	-	-	-	65.1	**67.8**
AX(Matthew Corr.)	34.2	**37.4**	-	-	-	39.6	**48.0**
Score	78.3	**80.6**	-	-	-	80.5	**83.6**

Table 5: The results on GLUE benchmark, where the results on dev set are the median of five runs and the results on test set are scored by the GLUE evaluation server (https://gluebenchmark.com/leaderboard). The state-of-the-art results are in bold. All of the fine-tuned models of AX is trained by the data of MNLI.

Task	Metrics	BERT$_{BASE}$		ERNIE 1.0$_{BASE}$		ERNIE 2.0$_{BASE}$		ERNIE 2.0$_{LARGE}$	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
CMRC 2018	EM/F1	66.3/85.9	-	65.1/85.1	-	69.1/88.6	-	**71.5/89.9**	-
DRCD	EM/F1	85.7/91.6	84.9/90.9	84.6/90.9	84.0/90.5	88.5/93.8	88.0/93.4	**89.7/94.7**	**89.0/94.2**
DuReader	EM/F1	59.5/73.1	-	57.9/72.1	-	61.3/74.9	-	**64.2/77.3**	-
MSRA-NER	F1	94.0	92.6	95.0	93.8	95.2	93.8	**96.3**	**95.0**
XNLI	Accuracy	78.1	77.2	79.9	78.4	81.2	79.7	**82.6**	**81.0**
ChnSentiCorp	Accuracy	94.6	94.3	95.2	95.4	95.7	95.5	**96.1**	**95.8**
LCQMC	Accuracy	88.8	87.0	89.7	87.4	**90.9**	**87.9**	**90.9**	**87.9**
BQ Corpus	Accuracy	85.9	84.8	86.1	84.8	86.4	85.0	**86.5**	**85.2**
NLPCC-DBQA	MRR/F1	94.7/80.7	94.6/80.8	95.0/82.3	95.1/82.7	95.7/84.7	95.7/85.3	**95.9/85.3**	**95.8/85.8**

Table 6: The results of 9 common Chinese NLP tasks. ERNIE 1.0 indicates model released by (Sun et al. 2019, ERNIE) . The reported results are the average of five experimental results, and the state-of-the-art results are in bold.

Pre-training method	Pre-training task	Training iterations (steps)				Fine-tuning result		
		Stage 1	Stage 2	Stage 3	Stage 4	MNLI	SST-2	MRPC
Continual Learning	Knowledge Masking	50k	-	-	-	77.3	86.4	82.5
	Capital Prediction	-	50k	-	-			
	Token-Document Relation	-	-	50k	-			
	Sentence Reordering	-	-	-	50k			
Multi-task Learning	Knowledge Masking	50k				78.7	87.5	83.0
	Capital Prediction	50k						
	Token-Document Relation	50k						
	Sentence Reordering	50k						
continual Multi-task Learning	Knowledge Masking	20k	10k	10k	10k	**79.0**	**87.8**	**84.0**
	Capital Prediction	-	30k	10k	10k			
	Token-Document Relation	-	-	40k	10k			
	Sentence Reordering	-	-	-	50k			

Table 7: The results of different methods of continual pre-training. We use knowledge masking, capital prediction, token-document relation and sentence reordering as our pre-training tasks. we sample 10% training data from our whole pre-training corpus. We train the model with 4 tasks altogether from scratch in multi-task learning method and train the model in 4 stages in other two learning methods. We train different tasks in different stages. The learning order of these tasks is the same as the above tasks listed. To compare the result fairly, each of these 4 tasks are updated in 50,000 steps . The size of pre-training model is same as ERNIE base. We choose MNLI-m, SST-2 and MRPC as our fine-tuning dataset. The fine-tuning result is average of five random start. the fine-tuning experiment set is same as Table 3.

ing. Specifically, the following Chinese datasets are chosen to evaluate the performance of ERNIE 2.0 on Chinese tasks:

- **Machine Reading Comprehension (MRC)**: CMRC 2018 (Cui et al. 2018), DRCD (Shao et al. 2018), and DuReader (He et al. 2017).
- **Named Entity Recognition (NER)**: MSRA-NER (Levow 2006).
- **Natural Language Inference (NLI)**: XNLI (Conneau et al. 2018).
- **Sentiment Analysis (SA)**: ChnSentiCorp [4].
- **Semantic Similarity (SS)**: LCQMC (Liu et al. 2018), and BQ Corpus (Chen et al. 2018).
- **Question Answering (QA)**: NLPCC-DBQA [5].

Implementation Details for Fine-tuning

Detailed fine-tuning experimental settings of English tasks are shown in Table 3 while that of Chinese tasks are shown in Table 4.

Experimental Results

Results on English Tasks We evaluate the performance of the base models and the large models of each method on GLUE. Considering the fact that only the results of the single model XLNet on the dev set are reported, we also reports the results of each method on the dev set. In order to obtain a fair comparison with BERT and XLNet, we run a single-task and single-model [6] ERNIE 2.0 on the dev set. The detailed results on GLUE are depicted in Table 5.

As shown in the *BASE model* columns of Table 5, ERNIE 2.0$_{\text{BASE}}$ outperforms BERT$_{\text{BASE}}$ on all of the 10 tasks and obtains a score of 80.6. As shown in the dev columns of *LARGE model* section in Table 5, ERNIE 2.0$_{\text{LARGE}}$ consistently outperforms BERT$_{\text{LARGE}}$ and XLNet$_{\text{LARGE}}$ on most of the tasks except MNLI-m. Furthermore, as shown in the *LARGE model* section in Table 5, ERNIE 2.0$_{\text{LARGE}}$ outperforms BERT$_{\text{LARGE}}$ on all of the 10 tasks, which gets a score of 83.6 on the GLUE test set and achieves a 3.1% improvement over the previous SOTA pre-training model BERT$_{\text{LARGE}}$.

Results on Chinese Tasks Table 6 shows the performances on 9 classical Chinese NLP tasks. It can be seen that ERNIE 1.0$_{\text{BASE}}$ outperforms BERT$_{\text{BASE}}$ on XNLI, MSRA-NER, ChnSentiCorp, LCQMC and NLPCC-DBQA tasks, yet the performance is less ideal on the rest, which is caused by the difference in pre-training between the two methods. Specifically, the pre-training data of ERNIE 1.0$_{\text{BASE}}$ does not contain instances whose length exceeds 128, but BERT$_{\text{BASE}}$ is pre-trained with the instances whose length are 512. From the results, it can be also seen that the proposed ERNIE 2.0 makes further progress, which significantly outperforms BERT$_{\text{BASE}}$ on all of the nine tasks. Furthermore, we train a large version of ERNIE 2.0. ERNIE 2.0$_{\text{LARGE}}$ achieves the best performance and creates new state-of-the-art results on these Chinese NLP tasks.

Comparison of Different Learning Methods

In order to analyze the effectiveness of the continual multi-task learning strategy adopted in our framework, we compare this method with two other methods as shown in figure 2. Table 7 describes the detailed information. For all the methods, we assume that the training iterations are the same for each task. In our settings, each task can be trained in 50k iterations, with 200k iterations for all of the tasks. As it can be seen, multi-task learning trains all the tasks in one stage, continual pre-training trains the tasks one by one, while our continual multi-task learning method can assign different iterations to each task in different training stages. The experimental result shows that continual multi-task learning obtains the better performance on downstream tasks compared with the other two methods, without sacrificing any efficiency. The result also indicates that our pre-training method can trains the new tasks in a more effective and efficient way. Moreover, the comparison between continual multi-task learning, multi-task learning and traditional continual learning shows that the first two methods outperform the third one, which confirms our intuition that traditional continual learning tends to forget the knowledge it has learnt when there is only one new task involved each time.

Conclusion

We proposed a continual pre-training framework named ERNIE 2.0, in which pre-training tasks can be incrementally built and learned through continual multi-task learning in a continual way. Based on the framework, we constructed several pre-training tasks covering different aspects of language and trained a new model called ERNIE 2.0 model which is more competent in language representation. ERNIE 2.0 was tested on the GLUE benchmarks and various Chinese tasks. It obtained significant improvements over BERT and XLNet. In the future, we will introduce more pre-training tasks to the ERNIE 2.0 framework to further improve the performance of the model. We will also investigate other sophisticated continual learning method in our framework.

Acknowledgements

This work is supported by the National Key Research and Development Project of China (No. 2018AAA0101900).

References

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Chen, Z., and Liu, B. 2018. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12(3):1–207.

Chen, J.; Chen, Q.; Liu, X.; Yang, H.; Lu, D.; and Tang, B. 2018. The bq corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4946–4951.

[4] https://github.com/pengming617/bert_classification

[5] http://tcci.ccf.org.cn/conference/2016/dldoc/evagline2.pdf

[6] which mean the result without additional tricks such as ensemble and multi-task losses.

Conneau, A.; Lample, G.; Rinott, R.; Williams, A.; Bowman, S. R.; Schwenk, H.; and Stoyanov, V. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Cui, Y.; Liu, T.; Xiao, L.; Chen, Z.; Ma, W.; Che, W.; Wang, S.; and Hu, G. 2018. A span-extraction dataset for chinese machine reading comprehension. *CoRR* abs/1810.07366.

Dai, Z.; Yang, Z.; Yang, Y.; Cohen, W. W.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

He, W.; Liu, K.; Liu, J.; Lyu, Y.; Zhao, S.; Xiao, X.; Liu, Y.; Wang, Y.; Wu, H.; She, Q.; et al. 2017. Dureader: a chinese machine reading comprehension dataset from real-world applications. *arXiv preprint arXiv:1711.05073*.

Lample, G., and Conneau, A. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Levow, G.-A. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 108–117.

Liu, X.; Chen, Q.; Deng, C.; Zeng, H.; Chen, J.; Li, D.; and Tang, B. 2018. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1952–1962.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Sang, E. F., and De Meulder, F. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Shao, C. C.; Liu, T.; Lai, Y.; Tseng, Y.; and Tsai, S. 2018. Drcd: a chinese machine reading comprehension dataset. *arXiv preprint arXiv:1806.00920*.

Sileo, D.; Van-De-Cruys, T.; Pradel, C.; and Muller, P. 2019. Mining discourse markers for unsupervised sentence representation learning. *arXiv preprint arXiv:1903.11850*.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; and Wu, H. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.