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.
{sunyu02,wangshuohuan,liyukun01,fengshikun01,tianhao,wu_hua,wanghaifeng}@baidu.com

ABSTRACT

Recently, pre-trained models have achieved state-of-the-art results in various language understanding tasks, which indicates that pre-training on large-scale corpora may play a crucial role in natural language processing. 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, there exists other valuable lexical, syntactic and semantic information in training corpora, such as named entity, semantic closeness and discourse relations. In order to extract to the fullest extent, the lexical, syntactic and semantic information from training corpora, we propose a continual pre-training framework named ERNIE 2.0 which builds and learns incrementally pre-training tasks through constant multi-task learning. Experimental results demonstrate that ERNIE 2.0 outperforms BERT and XLNet on 16 tasks including English tasks on GLUE benchmarks and several common tasks in Chinese. The source codes and pre-trained models have been released at https://github.com/PaddlePaddle/ERNIE.

1 Introduction

Pre-trained language representations such as ELMo[1], OpenAI GPT[2], BERT [3], ERNIE 1.0 [4][1] and XLNet[5] have been proven to be effective for improving the performances of various natural language understanding tasks including sentiment classification [6], natural language inference [7], named entity recognition [8] 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 proximity between sentences enable the models to learn structure-aware representations. Whereas, 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 constant multi-task learning.

Our ERNIE framework supports the introduction of various customized tasks at any time. These tasks share the same encoding networks and are trained through multi-task learning. This method makes the encoding of lexical, syntactic and semantic information across different tasks possible. Moreover, when given a new task, our framework could incrementally train the distributed representations according to the previous training parameters that it grasped.

In summary, our contributions are as follows:

- We propose a continual pre-training framework ERNIE 2.0, which supports customized training tasks and multi-task pre-training in an incremental way.
- We construct several 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.

[1] In order to distinguish our new ERNIE 2.0 framework and the ERNIE model, the latter is referred to as ERNIE 1.0.[4]

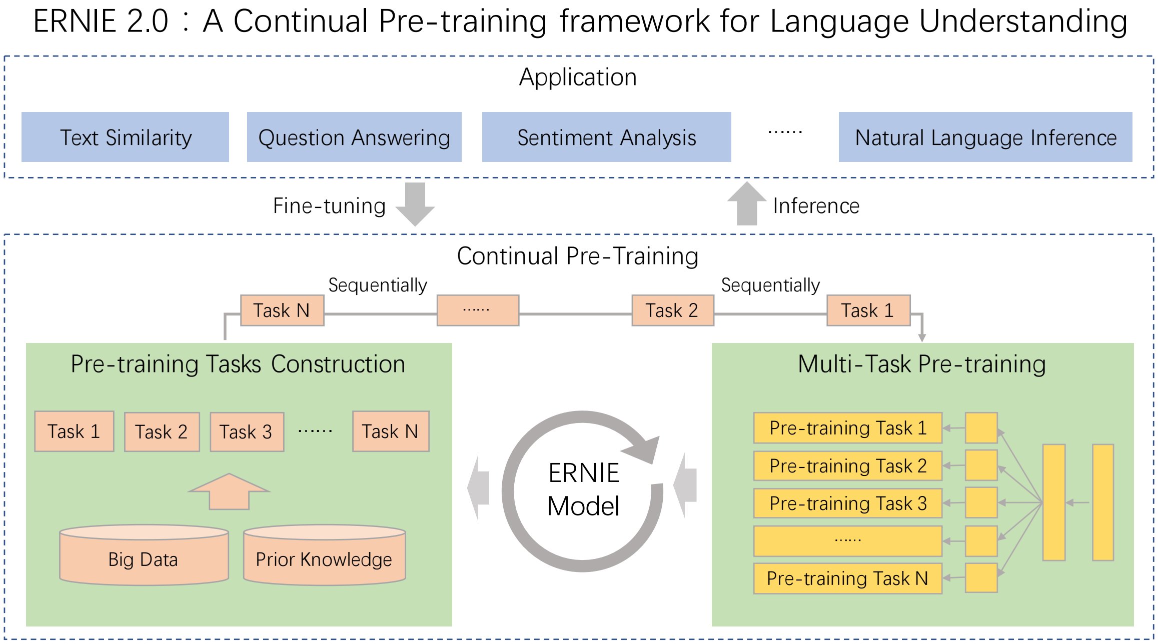

Figure 1: The framework of ERNIE 2.0, where the pre-training tasks can be incrementally constructed, the models are pre-trained through multi-task learning, and the pre-trained model is fine-tuned to adapt to various language understanding tasks.

- Our fine-tuning code of ERNIE 2.0 and models pre-trained on English corpora are available at https://github.com/PaddlePaddle/ERNIE.

2 Related Work

2.1 Unsupervised Transfer Learning for Language Representation

It is effective to learn general language representation by pre-training a language model with a large amount of unannotated data. Traditional methods usually focus on context-independent word embedding. Methods such as Word2Vec[9] and GloVe[10] learn fixed word embeddings through word co-occurrence 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[1] proposes to extract context-sensitive features from a language model. OpenAI GPT[2] enhances the context-sensitive embedding by adjusting the Transformer[11]. BERT[3], however, adopts a masked language model while adding a next sentence prediction task into the pre-training. XLM[12] integrates two methods to learn cross-lingual language models, namely the unsupervised method that relies only on monolingual data and supervised method that leverages parallel bilingual data. MT-DNN[13] achieves a better result through learning several supervised tasks in GLUE[14] 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[5] uses Transformer-XL[15] and proposes a generalized autoregressive pre-training method that learns bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order.

2.2 Continual Learning

Continual learning[16, 17] aims to train the model with several tasks in sequence so that it remembers the previously-learned tasks when learning the new ones. This method is 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.

3 The ERNIE 2.0 Framework

As shown in Figure 1, the ERNIE 2.0 framework is built based on an architecture of pre-training and fine-tuning which recently grows in popularity in natural language processing. ERNIE 2.0 differs from traditional pre-training methods 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. On top of that, ERNIE 2.0 framework keeps updating the pre-trained model with 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.

3.1 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 multi-task learning.

For pre-training task construction, we construct different kinds of tasks including word-aware tasks, 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. For multi-task pre-training, the ERNIE 2.0 framework trains all of these tasks in a continuous learning paradigm. Specifically, we would first train an initial model with a simple task before constantly introducing new pre-training tasks to upgrade the model. When adding a new task, we initialize the parameters of the previous one. Whenever a new task is introduced, it would be trained with the previous ones to make sure that the model does not forget the knowledge it has learnt. In this way, the ERNIE framework would be able to keep learning and accumulating the knowledge acquired during the process, and the accumulation of knowledge would enable the models to perform better during new tasks.

As shown in Figure 2, the architecture of continual pre-training 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[11]. The parameters of the encoder can be updated across all pre-training tasks.

There are two kinds of loss functions in our framework. One is the sequence-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 or more sentence-level loss functions can be combined with multiple token-level loss functions to continually update the model.

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

4 ERNIE 2.0 Model

In order to verify the effectiveness of the framework, we construct several 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.

4.1 Model Structure

Transformer Encoder The model uses a multi-layer Transformer[11] as the basic encoder like other pre-training models such as GPT[2], BERT[3] and XLM[12]. 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 modulate 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.

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

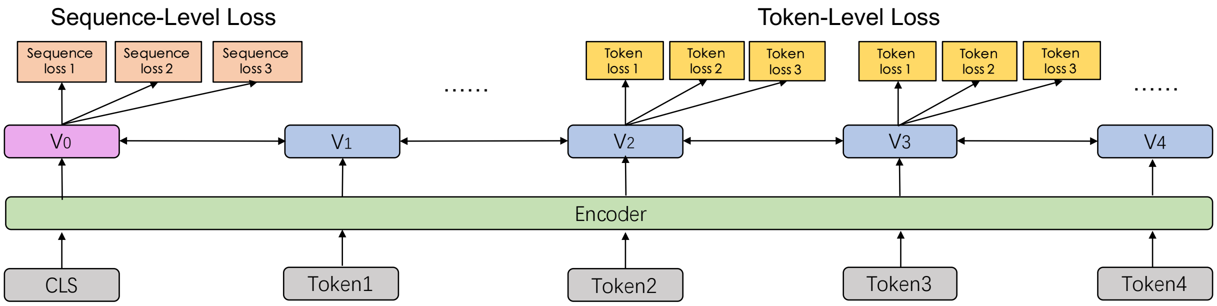

Figure 2: The architecture of multi-task pre-training in the ERNIE 2.0 framework, in which the encoder can be recurrent neural networks or a deep transformer.

4.2 Pre-training Tasks

We construct several tasks to capture different aspects of information in the training corpora. The word-aware tasks teach the model to capture the lexical information, the structure-aware tasks teach the model to capture the syntactic information of the corpus and the semantic-aware tasks are in charge of semantic signals. These pre-training tasks will be described in detail in the following part.

4.2.1 Word-aware Pre-training Tasks

Knowledge Masking Task ERNIE 1.0[4] propose 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 value 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 We add a task to predict 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 key 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.

4.2.2 Structure-aware Pre-training Tasks

Sentence Reordering Task We add a sentence reordering task 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.

4.2.3 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[18] to train a pre-trained model for English tasks. Following the method in Sileo et.al[18], we also automatically construct a Chinese dataset for pre-training.

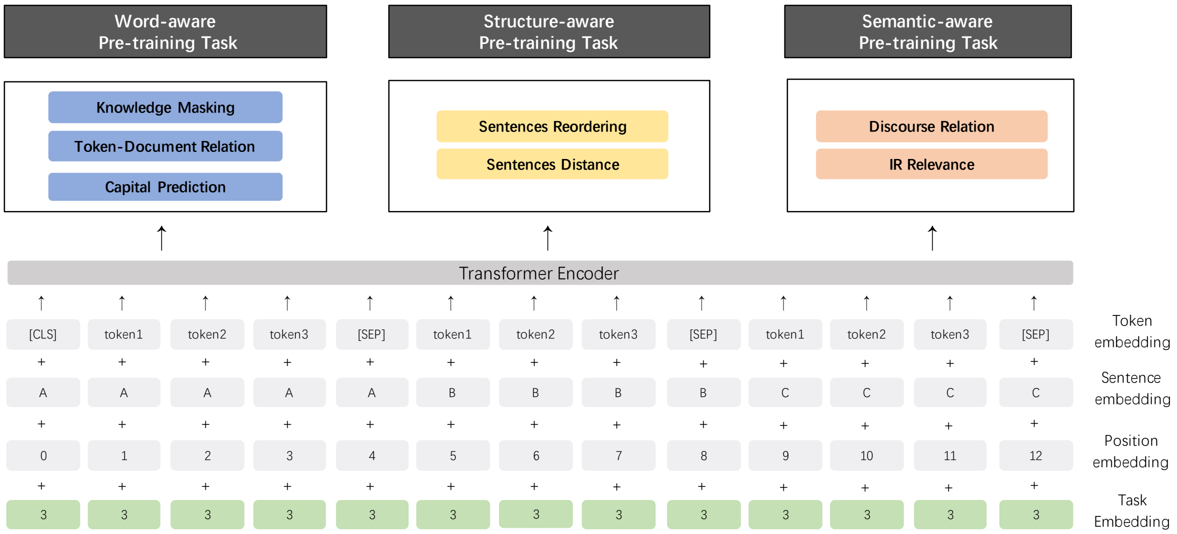

Figure 3: The structure of the ERNIE 2.0 model. The input embedding contains the token embedding, the sentence embedding, the position embedding and the task embedding. Seven pre-training tasks belonging to different kinds are constructed in the ERNIE 2.0 model.

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 Baidu 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.

5 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[3] and XLNet[5] on GLUE. For Chinese tasks, we compare the results with that of BERT[3] and the previous ERNIE 1.0[4] model on several Chinese datasets.

5.1 Pre-training and Implementation

5.1.1 Pre-training Data

Similar to that of BERT, some data in the English corpus are crawled from Wikipedia and BookCorpus. On top of that we collect some from Reddit. We also use the Discovery data[18] 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 Baidu Search Engine. The details of the pre-training data are shown in Table 1.

5.1.2 Pre-training Settings

To compare with BERT[3], 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[5] are same as BERT.

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[11] 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 converges.

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 1: The size of pre-training datasets.

5.2 Fine-tuning Tasks

5.2.1 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, including:

- **CoLA**: The Corpus of Linguistic Acceptability (CoLA)[19] consists of 10657 sentences from 23 linguistics, annotated for acceptability (grammatically) by their original authors. CoLA is commonly used in the task of judging whether a sentence conforms to the syntax specification.
- **SST-2**: The Stanford Sentiment Treebank (SST-2)[6] consists of 9645 movie reviews and is annotated for sentiment analysis.
- **MNLI**: Multi-genre Natural Language Inference (MNLI)[20] is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information, and is usually used for textual inference tasks.
- **RTE**: Recognizing Textual Entailment (RTE)[21] is a corpus similar to MNLI and is usually used for Natural Language Inference task.
- **WNLI**: Winograd Natural Language Inference (WNLI)[22] is a corpus that captures the coreference information between two paragraphs.
- **QQP**: Quora Question Pairs (QQP)[3] consists of over 400,000 sentence pairs with data extracted from Quora QA community and is commonly used in tasks for judging whether the question pairs are duplicates or not.
- **MRPC**: Microsoft Research Paraphrase Corpus (MRPC)[23] contains 5800 pairs of sentences extracted from news on the Internet and is annotated to capture the equivalence of paraphrase or semantic relationship between a pair of sentences. MRPC is commonly used in similar tasks as QQP.
- **STS-B**: The Semantic Textual Similarity Benchmark (STS-B)[24] contains a selection of the English datasets. These datasets contain texts from image captions, news headlines and user forums.
- **QNLI**: Question Natural Language Inference (QNLI)[25, 26] is a corpus that would tell whether the relationship between a pair of given texts are question-answer.
- **AX**: It is a auxilary hand-crafted diagnostic test suite that enables detailed linguistic analysis of models.

Table 2 is the detailed information of the datasets in GLUE.

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

- **Machine Reading Comprehension (MRC)**: CMRC 2018 [27], DRCD [28], DuReader [29]
- **Named Entity Recognition (NER)**: MSRA-NER (SIGHAN 2006) [30]
- **Natural Language Inference (NLI)**: XNLI [31]

[3] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

Corpus	Task	#Train	#Dev	#Test	#Label	Metrics
CoLA	Acceptability	8.5k	1k	1k	2	Matthews corr
SST-2	Sentiment	67k	872	1.8k	2	Accuracy
MNLI	NLI	393k	20k	20k	3	Accuracy
RTE	NLI	2.5k	276	3k	2	Accuracy
WNLI	NLI	634	71	146	2	Accuracy
QQP	Paraphrase	364k	40k	391k	2	Accuracy/F1
MRPC	Paraphrase	3.7k	408	1.7k	2	Accuracy/F1
STS-B	Similarity	7k	1.5k	1.4k	1	Pearson/Spearman corr
QNLI	QA/NLI	108k	5.7k	5.7k	2	Accuracy
AX	NLI	-	-	1.1k	3	Matthews corr

Table 2: The details of GLUE benchmark. The #Train, #Dev and #Test denote the size of the training set, development set and test set of corresponding corpus respectively. The #label denotes the size of the label set of the corresponding corpus.

- **Sentiment Analysis (SA)**: ChnSentiCorp [4]
- **Semantic Similarity (SS)**: LCQMC [32], BQ Corpus [33]
- **Question Answering (QA)**: NLPCC-DBQA [5]

The details of these datasets are shown in Table 3.

Corpus	Task	#Train	#Dev	#Test	#Label	Metrics
CMRC 2018	MRC	10K	3.2K	-	-	EM/F1
DRCD	MRC	27K	3.5K	3.5K	-	EM/F1
DuReader	MRC	271.5K	10K	-	-	EM/F1
MSRA-NER	NER	21K	2.3k	4.6K	7	F1
XNLI	NLI	392K	2.5K	2.5K	3	Accuracy
ChnSentiCorp	SA	9.6K	1.2K	1.2K	2	Accuracy
LCQMC	SS	240K	8.8K	12.5K	2	Accuracy
BQ Corpus	SS	100K	10K	10K	2	Accuracy
NLPCC-DBQA	QA	182K	41K	82K	2	mrr/F1

Table 3: The details of Chinese NLP datasets. The #Train, #Dev and #Test denote the size of the training set, development set and test set of corresponding corpus respectively. The #label denotes the size of the label set of the corresponding corpus.

Machine Reading Comprehension Machine Reading Comprehension (MRC) is a representative document-level modeling task which aims to extract continuous segments from the given text to answer the questions. For MRC task, CMRC 2018 [27], DRCD [28], DuReader [29] are used as test datasets.

- **CMRC 2018**: Chinese Machine Reading Comprehension 2018 (CMRC 2018) is an extractive reading comprehension dataset for machine reading comprehension, and it is released by the Chinese Information Processing Society of China, IFLYTEK and Harbin Institute of Technology.
- **DRCD**: Delta Reading Comprehension Dataset (DRCD) is also an extractive reading comprehension dataset which is released by Delta Research Institute. Notably, DRCD is a Traditional Chinese dataset, so we pre-converted it to simplified Chinese using a released tool[6].
- **DuReader**: DuReader is a large-scale real-world Chinese dataset for machine reading comprehension and question answering, which is released by Baidu at ACL 2018. All questions in the dataset are sampled from real anonymous user queries and the answers to the questions are manually generated. The experiments are carried out on the subset of DuReader for MRC.

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

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

[6] https://github.com/skydark/nstools/tree/master/zhtools

Named Entity Recognition Named Entity Recognition (NER) aims to recognize various entities, including person names, location names and organization names and so on. It can be seen as a sequence labeling task. For the NER task, the MSRA-NER (SIGHAN 2006)[30] dataset released by Microsoft Research Asia is chosen.

Natural Language Inference Natural Language Inference (NLI) aims to determine the semantic relationship (entailment, contradiction and neutral) between two sentences or two words. For the NLI task, we choose the popular XNLI [31] dataset, which is a crowd-sourced collection for the MultiNLI corpus and the pairs in the dataset are annotated with textual entailment and translated into 14 languages including Chinese.

Sentiment Analysis Sentiment Analysis (SA) aims to analyze whether the sentiment of a sentence is positive or negative. SA can be simply regraded as a binary classification task. For the SA task, we use the ChnSentiCorp dataset, which includes comments in several domains such as hotels, books, and electronic computers.

Semantic Similarity Semantic Similarity (SS) aims to identify whether two sentences have the same intention based on the likeness of their meaning or semantic content. For the SS task, we use the LCQMC [32] and BQ Corpus [33] datasets. LCQMC is released by Harbin Institute of Technology at COLTING 2018. BQ Corpus is jointly released by Harbin Institute of Technology and WeBank at EMNLP 2018. Each pair of sentences in the two datasets is associated with a binary label indicating whether the two sentences share the same intention.

Question Answering Question Answering (QA) aims to select answers for the corresponding questions. For the QA task, we use the NLPCC-DBQA dataset, which is released at NLPCC in 2016.

5.3 Implementation Details

Detailed fine-tuning experimental settings of English tasks are shown in Table 4 while that of Chinese tasks are shown in Table 5.

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 4: 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	65536(#tokens)	3	4e-5	65536(#tokens)
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 5: The Experiment Settings for Chinese datasets

5.4 Experimental Results

5.4.1 Results on English Tasks

In order to ensure the integrity of the experiments, we evaluate the performance of the base models and the large models of each comparison method on GLUE. Notably, considering the fact that only the results of the single model XLNet on the dev set are reported, we can only compare the performance of ERNIE 2.0 and XLNet on the dev set. In order to obtain a fair comparison with BERT and XLNet, we run a single-task and single-model[7] ERNIE 2.0 on the dev set. The detailed results on GLUE are depicted in Table 6.

As shown in the *BASE model* columns of Table 6, 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 6, ERNIE 2.0$_{\text{LARGE}}$ consistently outperforms BERT$_{\text{LARGE}}$ and XLNet$_{\text{LARGE}}$ on all of 8 tasks except MNLI-m. Furthermore, as shown in the test columns of *LARGE model* section in Table 6, ERNIE 2.0$_{\text{LARGE}}$ outperforms BERT$_{\text{LARGE}}$ on all of the 10 tasks, with an improvement of 10.1 points, 8.4 points, 3.0 points, 2.9 points, 2.7 points, 2.0 points and 1.9 points on RTE, AX, CoLA, MNLI-mm, WNLI, MNLI-m and QNLI respectively. Concretely, ERNIE 2.0$_{\text{LARGE}}$ 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$_{\text{LARGE}}$.

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 6: The results on GLUE benchmark, where the results on dev set are the median of five experimental results 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.

5.4.2 Results on Chinese Tasks

Table 7 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, ERNIE 2.0$_{\text{LARGE}}$ achieves the best performance and creates new state-of-the-art results on these Chinese NLP tasks. Specifically, ERNIE 2.0$_{\text{LARGE}}$ yields improvements of more than 2 points over BERT$_{\text{BASE}}$ on the CMRC 2018, DRCD, DuReader, XNLI, MSRA-NER and LCQMC tasks, and yields improvements of more than 2 points over ERNIE$_{\text{BASE}}$ on the CMRC 2018, DRCD, DuReader and XNLI tasks.

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 7: The results of 9 common Chinese NLP tasks. ERNIE 1.0 indicates our previous model ERNIE[4]. The reported results are the average of five experimental results, and the state-of-the-art results are in bold.

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

6 Conclusion

We proposed a continual pre-training framework named ERNIE 2.0, in which pre-training tasks can be incrementally built and learned through 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.

References

[1] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[2] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

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

[4] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

[5] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[6] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[7] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

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

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[10] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[12] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

[13] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[14] Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[15] Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[16] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

[17] Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018.

[18] Damien Sileo, Tim Van-De-Cruys, Camille Pradel, and Philippe Muller. Mining discourse markers for unsupervised sentence representation learning. *arXiv preprint arXiv:1903.11850*, 2019.

[19] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

[20] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[21] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. In *TAC*, 2009.

[22] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

[23] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[24] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

[25] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[26] Wei Wang, Ming Yan, and Chen Wu. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *arXiv preprint arXiv:1811.11934*, 2018.

[27] Yiming Cui, Ting Liu, Li Xiao, Zhipeng Chen, Wentao Ma, Wanxiang Che, Shijin Wang, and Guoping Hu. A span-extraction dataset for chinese machine reading comprehension. *CoRR*, abs/1810.07366, 2018.

[28] Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. Drcd: a chinese machine reading comprehension dataset. *arXiv preprint arXiv:1806.00920*, 2018.

[29] Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, et al. Dureader: a chinese machine reading comprehension dataset from real-world applications. *arXiv preprint arXiv:1711.05073*, 2017.

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

[31] Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*, 2018.

[32] Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962, 2018.

[33] Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 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*, pages 4946–4951, 2018.