8-2021

A BERT-based two-stage model for Chinese Chengyu recommendation

Minghuan TAN
Singapore Management University

Jing JIANG
Singapore Management University, jingjiang@smu.edu.sg

Bingtian DAI
Singapore Management University, btdai@smu.edu.sg

Citation

TAN, Minghuan; Jing JIANG; and DAI, Bingtian. A BERT-based two-stage model for Chinese Chengyu recommendation. (2021). *ACM Transactions on Asian and Low-Resource Language Information Processing*. Research Collection School Of Computing and Information Systems.

Available at: https://ink.library.smu.edu.sg/sis_research/5821

A BERT-based Two-Stage Model for Chinese Chengyu Recommendation

MINGHUAN TAN, JING JIANG, and BING TIAN DAI, Singapore Management University, Singapore

In Chinese, Chengyu are fixed phrases consisting of four characters. As a type of idioms, their meanings usually cannot be derived from their component characters. In this paper, we study the task of recommending a Chengyu given a textual context. Observing some of the limitations with existing work, we propose a two-stage model, where during the first stage we re-train a Chinese BERT model by masking out Chengyu from a large Chinese corpus with a wide coverage of Chengyu. During the second stage, we fine-tune the retrained, Chengyu-oriented BERT on a specific Chengyu recommendation dataset. We evaluate this method on ChID and CCT datasets and find that it can achieve the state of the art on both datasets. Ablation studies show that both stages of training are critical for the performance gain.

CCS Concepts: • **Computing methodologies → Natural language processing**.

Additional Key Words and Phrases: question answering, chengyu recommendation, idiom understanding

1 INTRODUCTION

Chengyu (成语) in Chinese are fixed phrases with idiomatic meanings. They usually consist of four characters and their meanings often cannot be directly derived from their component characters [Wang and Yu 2010]. For example, the Chengyu "虎头蛇尾" means "to start strong but finish weak." However, the literal meanings of the four Chinese characters are "tiger," "head," "snake" and "tail." Most Chengyu originated from ancient literature like Chinese Classics, which may be hard to grasp even for native speakers. But when properly used, Chengyu can make the language concise and elegant [Liu et al. 2019b], which is why they are being widely used in both formal writings and colloquial conversations. Researchers have shown that it is important for Chinese language processing methods to consider Chengyu when performing various NLP tasks such as computer-assisted essay writing [Liu et al. 2019b] and machine translation [Ho et al. 2014; Shao et al. 2018b].

In this paper we study how to train neural network models to "understand" Chengyu. While there are different ways to evaluate whether a model "understands" Chengyu, here we focus on the task of Chengyu recommendation, that is, given a context such as a paragraph of text with a missing word in the middle, the machine needs to recommend a Chengyu to fill in the blank. Table 1 shows an example of the Chengyu recommendation task. We choose this task because it is very similar to how we would test a human's understanding of Chengyu.

Despite the importance of Chengyu in Chinese language understanding, there have been only a few pieces of work on Chengyu recommendation using neural models [Jiang et al. 2018; Liu et al. 2019b; Zheng et al. 2019]. Existing work falls under two settings. The first setting is to recommend a Chengyu given a context without any candidate answers. In this case essentially all Chinese Chengyu are candidates. We refer to this setting as *open-ended* Chengyu recommendation. Liu et al. [2019b] studied this setting and proposed an encoder-decoder model that generates the answer Chengyu character by character. However, because Chengyu's meanings are oftentimes not compositional from their component characters, this method may generate characters that cannot be combined into a meaningful Chengyu and thus affect the performance. The second setting assumes that a relatively small set of candidate Chengyu is given, from which the machine

Authors' address: Minghuan Tan, mhtan.2017@phdcs.smu.edu.sg; Jing Jiang, jingjiang@smu.edu.sg; Bing Tian Dai, btdai@smu.edu.sg, Singapore Management University, School of Computing and Information Systems, 80 Stamford Road, Singapore, Singapore, Singapore, 178902.

Passage: 改建过程中，随时可以添加一些经典的内置储藏柜。用这样的柜子存放香料和调味品，使用金属罐来增添老式情调，完全不会有______的感觉。
During the renovation process, you can add some classic built-in storage cabinets at any time. With such a cabinet to store spices and condiments, together with metal jars to create an old-fashioned atmosphere, you will not feel ______ at all.

Candidates:

○ 深明大义 deep and righteous	○ 前功尽弃 all one's previous efforts wasted
○ 天旋地转 very dizzy	○ 七零八碎 bits and pieces
○ 错落有致 well-arranged	● 杂乱无章 disorganized
○ 井然有序 in good order	

Table 1. An example passage with a blank to be filled, together with the candidate answers. The answer beside the solid circle is the ground truth answer.

needs to pick the best answer. Table 1 is such an example. We refer to this setting as *multiple-choice* Chengyu recommendation. Jiang et al. [2018] and Zheng et al. [2019] both formulated the task in this way and trained the recommendation model to separate the ground truth Chengyu from the incorrect candidate answers. However, this training objective ignores the fact that other Chengyu not in the candidate set are essentially also negative examples and not utilizing these negative examples may potentially lose much useful information.

In this paper, we focus on multiple-choice Chengyu recommendation, mainly because the two benchmark datasets we have, ChID [Zheng et al. 2019] and CCT [Jiang et al. 2018], both define the task as multiple-choice recommendation. To address the aforementioned limitations with existing work, we first treat each Chengyu as a single token rather than four separate characters. We further hypothesize that considering all other Chengyu not in the candidate set as negative examples may help multiple-choice recommendations. Hence, we propose a two-stage Chengyu recommendation model. Our model consists of a pre-training stage and a fine-tuning stage. The pre-training stage produces a Chengyu-oriented Chinese BERT model trained on open-ended Chengyu recommendation task. The fine-tuning stage further fine-tunes the pre-trained BERT on multiple-choice Chengyu recommendation data in order to optimize it for multiple-choice recommendation.

Another limitation with existing studies is that the corpora they used do not have a high coverage of Chengyu. The ChID dataset, for example, covers 3,848 Chengyu. However, Chinese Chengyu dictionaries typically include around 20,000 Chengyu entries. To address this limitation, we collect a large corpus of Chinese text covering a much wider range of Chengyu and use this corpus for the pre-training stage.

We conduct experiments first on the ChID dataset to evaluate our two-stage model for multiple-choice Chengyu recommendation. We find that the two-stage model works very well, achieving state-of-the-art performance and substantially outperforming previous methods on the official release of ChID. We also conduct ablation studies to test the effectiveness of pre-training and fine-tuning separately, and we find that both stages of training are critical for the performance gain. We further test the model on a ChID competition dataset and CCT, another Chengyu recommendation dataset, and find that our model also works well on both, outperforming the state of the art. We further show that the Chengyu embeddings produced by pre-training can also be used for Chengyu emotion prediction and achieve decent performance.

2 RELATED WORK

2.1 Multiword Expressions and Idiom Recognition

Multiword Expressions (MWEs) are defined as "idiosyncratic interpretations that cross word boundaries (or spaces)" or simply words-with-spaces [Sag et al. 2002]. Discrimination between compositional and non-compositional MWEs [Katz and Giesbrecht 2006] has been an important research topic as idiomatic uses of non-compositional MWEs can affect the semantics of the text.

Recognition of idioms as a special kind of MWEs with non-compositionality has important values in sentence understanding and failures of recognition may lead to mistranslation between languages [Hashimoto et al. 2006; Lin 1999]. Statistical approaches [Hashimoto et al. 2006; Katz and Giesbrecht 2006] use lexical knowledge and linguistic properties to create either token-level or phrase-level classifiers to identify idioms. However, manually annotated data are required given additional challenges of ambiguity and fixedness.

In this work, we focus on a special kind of idiom, i.e., Chengyu in Chinese, which has high fixedness and low ambiguity. The recognition of Chengyu is straightforward since they almost always consist of four consecutive characters and can be identified from a Chengyu dictionary.

2.2 Chinese Chengyu Recommendation

Chinese Chengyu Recommendation (CCR) has been addressed in recent years by [Jiang et al. 2018; Liu et al. 2019b]. Jiang et al. [2018] formulate the CCR task as a cloze-test via incorporation of two BiLSTM networks to encode the definition of Chengyu and the context sentence separately followed by computing bilinear attentions following [Chen et al. 2016]. Liu et al. [2019b] reformulate the CCR problem as context-to-idiom machine translation problem by leveraging the attention-based encoder-decoder framework under the assumption that Chengyu are constructed from a pseudo language with positional vocabularies. Zheng et al. [2019] constructs the first large scale Chengyu cloze-test dataset ChID and offers strong baselines using Attentive Reader (AR) [Hermann et al. 2015] and Stanford Attentive Reader (SAR) [Chen et al. 2016].

Different from all the previous works, we aim at including as many Chengyu as possible and our pretraining task is open-ended Chengyu recommendation, which is more challenging.

2.3 Pre-training of Language Models

In the past several years, pre-trained language models have been shown to be highly effective in many NLP tasks. LM-LSTMs [Dai and Le 2015] is the first language model that adopts self-supervised pre-training using millions of in-domain documents. ULMFiT [Howard and Ruder 2018] improves language modeling transfer learning robustness and efficiency through discriminative fine-tuning, slanted triangular learning rate and gradual unfreezing. ELMO [Peters et al. 2018] pre-trains a bidirectional language model (biLM) offering high quality deep context-dependent representations.

With the Transformer [Radford et al. 2018; Vaswani et al. 2017] drawing more attentions, BERT [Devlin et al. 2019] proposes a two stage framework constructed over a multi-layer bidirectional Transformer. During pre-training, a large amount of data is fed into the model to be trained using self-supervised pre-training tasks. During fine-tuning, the model will be supervised by the labels of downstream tasks. BERT adopts two pre-training tasks, namely, the Masked Language Model (MLM) task and the Next Sentence Prediction (NSP) task.

There has been much work following BERT that modifies existing pre-training objectives and designs new pre-training tasks. Basically, these modifications can be grouped into masking-based approach and structural-based approach. The WWM method is masking-based since it is trying to fix masking where whole word is segmented into word pieces. Similar masking-based approaches

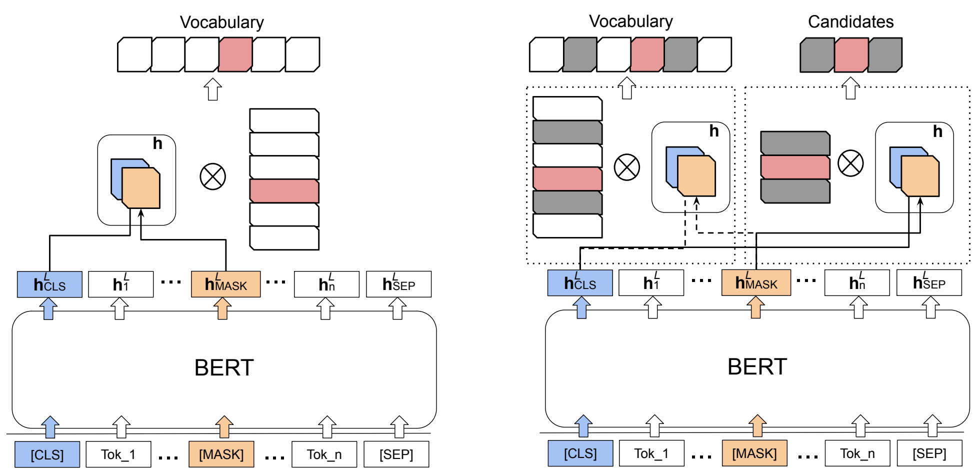

Fig. 1. Left: The network structure used for pre-training. Right: The network structure used for fine-tuning.

include masking random contiguous spans in SpanBERT [Joshi et al. 2020] and dynamic masking proposed in RoBERTa [Liu et al. 2019a]. The NSP task is a structural prediction task that a binary classification for predicting whether two segments follow each either in the original text. With further ablation study, NSP is either removed [Joshi et al. 2020; Yang et al. 2019] due to inconsistent improvement or restricted to use sentences from a single document [Liu et al. 2019a]. More structure-aware pre-training tasks are proposed by ERNIE 2.0 [Sun et al. 2020], StructBERT [Wang et al. 2020] and ALBERT [Lan et al. 2020]. ERNIE 2.0 uses Token-Document Relation Prediction and Sentence Reordering. StructBERT strengthens BERT with both word structural objective and sentence structural objective.

Chinese is an ideographic language with no word delimiter between words in written Chinese sentences [Li and Yuan 1998]. Therefore, BERT variations with Chinese compatibility are also based on new pre-training tasks. Chinese-BERT-wwm [Cui et al. 2019a] uses Chinese Word Segmentation (CWS) tools to identify word boundaries and mask a whole word explicitly. ERNIE [Zhang et al. 2019] incorporates a multi-stage knowledge masking strategy which adds word-level mask, phrase-level mask and entity-level mask, an extension to WWM.

In this work, We pre-train an Chengyu-oriented BERT based on Chinese-BERT-wwm as it has minimum difference from BERT. Given the fact that CWS tools can handle only a small percentage of Chengyu, we believe masking in Chinese-BERT-wwm is still sub-optimal. We therefore propose new pre-training tasks by isolating each Chengyu as a token with external embeddings and using a large Chengyu corpus to perform pre-training.

3 TWO-STAGE CHENGYU RECOMMENDATION

In this section, we first give the formal definition of the Chengyu recommendation task. We then present our two-stage model.

3.1 Task Definition

The Chinese Chengyu recommendation task can be formally defined as follows. We are given a passage P, which we represent as a sequence of tokens $(w_1, w_2, \ldots, \texttt{[MASK]}, \ldots, w_n)$. Here each token is a single Chinese character, except for the special "blank" token [MASK], which represents the missing Chinese Chengyu that we need to recommend. We are also given a set of K candidate Chinese Chengyu denoted as $\mathcal{A} = \{a_1, a_2, \ldots, a_K\}$. Our goal is to select the best option $a^* \in \mathcal{A}$ that fits the context in P. We have shown a concrete example in Table 1.

To train a Chengyu recommendation model, we assume that we are given a set of training examples, where each example is a triplet containing a passage, a candidate set and the ground truth answer. The training data is denoted as $\{(P_i, \mathcal{A}_i, a_i^*)\}_{i=1}^N$. We also use $\mathcal{V}$ to denote the vocabulary of all Chinese Chengyu observed in the training data, i.e., $\mathcal{V} = \cup_{i=1}^N \mathcal{A}_i$.

3.2 Model Overview

The model consists of a *pre-training stage* and a *fine-tuning stage*. The pre-training stage uses a Chinese corpus we have collected that covers a large set of Chengyu to produce a Chengyu-oriented Chinese BERT model, which we call the Chengyu-BERT.[1] The training task for Chengyu-BERT is a Masked Language Model task where only Chengyu are masked. We can also think of the training task as essentially open-ended Chengyu recommendation. The fine-tuning stage further optimizes the pre-trained Chengyu-BERT for multiple-choice Chengyu recommendation, where the goal is to choose a Chengyu among a small set of candidates given a context. The purpose of the fine-tuning stage is to learn the subtle differences between a Chengyu and its "near synonyms", i.e., other Chengyu which have similar meanings but still cannot be used as substitutes. These "near synonyms" occur often as candidate answers in multiple-choice Chengyu recommendation such as in the ChID dataset. We will see later that the two stages share similar network structure but have some major differences due to the differences between open-ended recommendation and multiple-choice recommendation.

It is worth noting that an alternative way to use open-ended Chengyu recommendation to assist multiple-choice recommendation is *multitask learning*, where the two tasks are jointly (i.e., concurrently) rather than sequentially trained. In this paper we do not adopt the multitask learning approach because of two reasons. First, the unlabeled dataset we use for pre-training the Chengyu-BERT is very large while the specially prepared multiple-choice recommendation data used for fine-tuning is relatively small. Therefore, training the two together would lead to an imbalanced objective function. Second, by separating the training of the two sequentially, the pre-trained Chengyu-BERT can also be used directly for Chengyu recommendation without fine-tuning or even for other Chengyu-related tasks such as Chengyu emotion prediction, which we will detail in Section 4.

3.3 Pre-training Stage

Our pre-training is done on top of Chinese-BERT-wwm [Cui et al. 2019a], which is an improved version of the original Chinese version of BERT [Devlin et al. 2019]. Chinese-BERT-wwm uses Whole Word Masking [Devlin et al. 2019] in its Masked Language Model pre-training task, and is found to work better for a number of NLP tasks [Cui et al. 2019b; Duan et al. 2019; Shao et al. 2018a]. However, Chinese-BERT-wwm is not ideal for Chengyu recommendation, because we find that only a small percentage (around 1%) of Chengyu in our Chengyu vocabulary is detected as whole words in Chinese-BERT-wwm. We thus use an extended version (trained with more data) of Chinese-BERT-wwm called Chinese-BERT-wwm-ext to initialize our model but re-train the model using a special Masked Language Model task where only Chengyu are masked. This can also be seen as the open-ended Chengyu recommendation task.

Specifically, we assume that we have a large corpus of unlabeled Chinese text. Let $\mathcal{V}$ denote the Chengyu vocabulary, i.e., the set of all Chengyu found in the corpus. Let $c = (w_1, w_2, \ldots, w_c, w_{c+1},$

[1]Note that this Chengyu-BERT is not meant to be a generic BERT for any Chinese NLP task.

$w_{c+2}, w_{c+3}, \ldots, w_n$) denote a context sequence where each w_i ($1 \leq i \leq n$) is a Chinese character and ($w_c, w_{c+1}, w_{c+2}, w_{c+3}$) forms a Chengyu. We first merge ($w_c, w_{c+1}, w_{c+2}, w_{c+3}$) into a single word $v \in \mathcal{V}$ where $\mathcal{V}$ is our Chengyu vocabulary. We then mask v with the special token `[MASK]` and feed the sequence into an L-layer BERT. Following standard practice, we prepend `[CLS]` to the beginning of the sequence and append `[SEP]` to the end of the sequence. We also include position embedding. For segment embedding, we treat the sequence as a single segment.

To evaluate whether a Chengyu is suitable for the given context, ideally we need to match the Chengyu with the entire sequence of hidden vectors produced by BERT. However, because in the open-ended recommendation setting we have a large number of candidates, it would be too expensive to match each Chengyu with the entire sequence of hidden states. We therefore focus on the token `[CLS]`, which represents an aggregated representation of the entire sequence, and the token `[MASK]`, which represents the local context of the blank. Let $\mathbf{h}^L_{\text{CLS}} \in \mathbb{R}^d$ denote the hidden vector produced by the last layer of BERT representing `[CLS]`, and $\mathbf{h}^L_{\text{MASK}} \in \mathbb{R}^d$ the similarly produced hidden vector representing `[MASK]`. We define the representation of the masked sequence $\mathbf{h} \in \mathbb{R}^d$ using a fusion function f, i.e., $\mathbf{h} = f(\mathbf{h}^L_{\text{CLS}}, \mathbf{h}^L_{\text{MASK}})$. We tried a few different choices of f in our preliminary experiments and found the following form, which follows the practice of [Tai et al. 2015; Wang and Jiang 2017], to be slightly better:

$$\mathbf{h} = \mathbf{W} \begin{bmatrix} \mathbf{h}^L_{\text{CLS}} \\ \mathbf{h}^L_{\text{MASK}} \\ \mathbf{h}^L_{\text{CLS}} \odot \mathbf{h}^L_{\text{MASK}} \\ \mathbf{h}^L_{\text{CLS}} - \mathbf{h}^L_{\text{MASK}} \end{bmatrix},$$

where $\odot$ is element-wise multiplication between two vectors and $\mathbf{W} \in \mathbb{R}^{d \times 4d}$ is a matrix to be learned.

We further assume that each Chengyu $v \in \mathcal{V}$ has an embedding vector $\mathbf{e}_v$ (to be learned), which is to be compared with $\mathbf{h}$ for prediction. We use softmax to compute the probability of selecting v given the context c:

$$p(v|c) = \frac{\exp(\mathbf{e}_v \cdot \mathbf{h})}{\sum_{v' \in \mathcal{V}} \exp(\mathbf{e}_{v'} \cdot \mathbf{h})}. \tag{1}$$

It is important to note that the probability here is normalized over *all* Chengyu in $\mathcal{V}$. Assume we have N training examples. Let c_n be the context of the n-th example, and let a^*_n be the ground truth answer for the n-th example. The loss function is then defined as follows:

$$L_{\mathcal{V}} = -\sum_{n=1}^{N} \log p(a^*_n | c_n). \tag{2}$$

The left side of Figure 1 illustrates the model used for pre-training.

3.3.1 Pre-training Data. We need a large corpus with a wide coverage of Chengyu for the pre-training stage. We collect the data through the following pipeline. (1) **Chengyu Vocabulary:** We construct an initial Chengyu vocabulary of 33,237 Chengyu by merging Chengyu found in multiple online resources, including Chengyu Daquan[2], Xinhua Chengyu Dictionary[3], Chengyu Cloze Test[4] and ChID[5]. (2) **Chengyu Corpus:** We collected a large corpus of Chinese text by crawling e-books online. Then for each Chengyu from the Chengyu vocabulary we retrieve contiguous sentences as its context. We choose to discard the context if its length is less than fifteen characters. Using this

[2] http://www.guoxue.com/chengyu/CYML.htm
[3] https://github.com/pwxcoo/chinese-xinhua
[4] https://github.com/bazingagin/chengyu_data
[5] https://github.com/zhengcj1/ChID-Dataset

procedure, we are able to collect a total number of 11 million contexts covering 22,786 Chengyu. (3) **Subsampling:** Although we have built a training set in huge number, we find that the distribution of sentences is extremely skewed for different Chengyu. The imbalance may hurt our pre-training task. Following [Mikolov et al. 2013], we use a subsampling approach to counter the imbalance between rare and frequent Chengyu as follows:

$$P(v) = \begin{cases} 1 & c(v) \leq 10 \\ 1 - \sqrt{\frac{t}{f(v)}} & c(v) > 10 \end{cases}, \tag{3}$$

where v is a Chengyu, $c(v)$ is the count of contexts of v in the dataset, $f(v) \in [0, 1]$ is the relative frequency of v and t is a chosen threshold. After using the subsampling method listed above, we are able to reduce the training instances to 5.9 million.

3.4 Fine-tuning Stage

For the second stage of fine-tuning, we assume that we have a set of training data where each training instance consists of a context sequence $c = (w_1, w_2, \ldots, \texttt{[MASK]}, \ldots, w_n)$ with `[MASK]` representing the blank to be filled, a small set of candidate answers $\mathcal{A} = \{a_1, a_2, \ldots\}$, and the ground truth correct answer $a^* \in \mathcal{A}$. Note that those incorrect candidates in $\mathcal{A}$ are often "near-synonyms" of a^*. The fine-tuning model follows the same way of using BERT to encode the input sequence as in the pre-training stage. The output of the L-layer BERT is a sequence of hidden vectors $\mathbf{h}_1^L, \mathbf{h}_2^L, \ldots, \mathbf{h}_n^L$, corresponding to the n tokens in the input sequence, including the `[MASK]` token.

It is worth noting that a major difference of the fine-tuning model from the pre-training model is the probability of choosing candidate a is normalized over just the small candidate set $\mathcal{A}$. This allows us to focus on learning the subtle differences between the ground truth answer a^* and its "near-synonyms".

Formally, the probability of choosing $a \in \mathcal{A}$ given context c is

$$p(a|c) = \frac{\exp(\mathbf{e}_a \cdot \mathbf{h})}{\sum_{a' \in \mathcal{A}} \exp(\mathbf{e}_{a'} \cdot \mathbf{h})}. \tag{4}$$

Note that here the probability is normalized over the candidate set $\mathcal{A}$.

Assume that we have N training examples. Let c_n denote the context of the n-the example and a_n^* the ground truth answer of the n-th example. We can define the following objective function:

$$L_{\mathcal{A}} = -\sum_{n=1}^{N} \log p(a_n^*|c_n). \tag{5}$$

Finally, in the fine-tuning stage, the training data for multiple-choice Chengyu recommendation can also be used as open-ended recommendation training data if we ignore the candidate set. We therefore can have an objective function below that combines the probability of the ground truth answer as computed by Eqn. (4) and the probability as computed by Eqn. (1), i.e., normalized over all Chengyu in $\mathcal{V}$:

$$L = L_{\mathcal{V}} + L_{\mathcal{A}}. \tag{6}$$

The right side of Figure 1 illustrates the model used for fine-tuning.

4 EXPERIMENTS ON CHENGYU RECOMMENDATION

In this section, we present the evaluation of our two-stage Chengyu recommendation model for multiple-choice recommendation.

4.1 Data and Experiment Settings

To facilitate the study of Chengyu comprehension using deep learning models, Zheng et al. [2019] released a large-scale Chinese Idiom Dataset called **ChID**. The dataset was created in the “cloze” style. The text includes novels and essays from the Internet and news articles. To construct the candidate answer set for each masked Chengyu, the authors considered synonyms, near-synonyms and other Chengyu either irrelevant or opposite in meaning to the ground truth Chengyu. The example in Table 1 is from ChID.

	In-domain			Out	Total
	Train	Dev	Test	Out	Total
Passages	520,711	20,000	20,000	20,096	580,807
Distinct Chengyun	3,848	3,458	3,502	3,626	3,848
Total blanks	648,920	24,822	24,948	30,023	728,713

Table 2. Some statistics of the ChID-Official dataset.

We use two different versions of the ChID datasets.

- **ChID-Official:** The first version is the official release of ChID. The data was released with a training set, a development set and a few different test sets. Besides the standard test set, the authors also constructed the following test sets: (1) **Ran**: In this test set, the candidate Chengyu were randomly sampled from the vocabulary $\mathcal{V}$. No synonyms or near-synonyms were intentionally added as candidates. (2) **Sim**: In this test set, the candidates were sampled from the top-10 Chengyu most similar to the ground truth Chengyu. It is therefore more challenging than the Ran test dataset. (3) **Out**: This is an out-of-domain test dataset. The test passages come from essays (whereas the training and development data comes from news and novels). The **Test**, **Ran** and **Sim** share the same context but have different candidate sets. Some statistics of the data can be found in Table 2.
- **ChID-Competition:** ChID-Competition[6] is the data for an online competition[7] on Chinese idiom comprehension. The data is a modified version of the ChID-Official. Different from ChID-Official, for each entry in ChID-Competition, a list of passages with blanks is given, and they share the same set of candidate Chengyu. Each candidate can be used only once within each entry. Table 3 shows part of an example entry. We can see that the three Chengyu “方兴未艾", “一日千里", “日新月异" in the candidate set share similar meanings and are all suitable for the blank Q000381 in Passage 2. However, Q000382 in Passage 3 can only choose “日新月异" and Q000383 in the Passage 4 can only choose "方兴未艾". As a result, "一日千里" will be the correct answer for Q000381. The challenge here is that the ground truth answers will be similar in semantic meaning and models need to distinguish their differences while comparing similar contexts to make the correct decisions. Therefore, under this setting, some heuristic global optimization strategies can be used to improve the performance. ChID-Competition is divided into four subsets: **Train**, **Dev**, **Test** and **Out** (for out-of-domain test data).

Although ChID is a large-scale dataset for Chengyu recommendation, it actually covers only over 3000 Chengyu. We therefore consider another Chengyu recommendation dataset that covers more Chengyu.

[6] https://github.com/zhengcj1/ChID-Dataset/tree/master/Competition
[7] https://biendata.com/competition/idiom/

Passage 2: 最近十年间，虚拟货币的发展可谓Q000381。美国著名经济学家林顿·拉鲁什曾预言：到2050年，基于网络的虚拟货币将在某种程度上得到官方承认，成为能够流通的货币。现在看来，这一断言似乎还嫌过于保守……
In the last decade, the development of virtual currency can be described as Q000381. Lyndon LaRouche, a famous American economist, predicted that virtual currency based on the Internet would be officially recognized as a currency in circulation to some extent by 2050. That assertion now seems too conservative……
Passage 3: “平时很少能看到这么多老照片，这次图片展把新旧照片对比展示，令人印象深刻。”现场一位参观者对笔者表示，大多数生活在北京的人都能感受到这个城市Q000382的变化，但很少有人能具体说出这些变化，……
"It's rare to see so many old photos, but this exhibition shows old and new photos in comparison, which is very impressive." A visitor to the scene told me that most people living in Beijing can feel the Q000382 changes of the city, but few people can describe these changes in detail. ……
Passage 4: 从今天大盘的走势看，市场的热点在反复的炒作之中，概念股的炒作Q000383，权重股走势较为稳健，大盘今日早盘的震荡可以看作是多头关前的蓄势行为。……
Judging from the trend of the market today, the hot spot in the market is repeated speculation, speculation of concept stocks Q000383, the trend of the weighted stocks is relatively stable, the market today morning trading shock can be seen as the preparation before the multi-head. ……

Candidates:

□ 百尺竿头 already have a great achievement	□ 随波逐流 go with the stream; drift along
□ 方兴未艾 be in the ascendant	□ 身体力行 earnestly practise what one advocates
□ 一日千里 at a tremendous pace	□ 三十而立 be independent at the age of thirty
□ 逆水行舟 sail against the current	□ 日新月异 change with each passing day
□ 百花齐放 All flowers bloom together.	□ 沧海一粟 a drop in the ocean

Table 3. An example in ChID-Competition. We show only three passages out of the five passages in this entry.

- **CCT**: Chengyu Cloze Test (CCT) [Jiang et al. 2018][8] is also a cloze-style dataset which contains 108,987 sentences covering 7,395 unique Chengyu. CCT data is crawled from the web and shows basic usage of each Chengyu[9].

We use 6 Nvidia 2080Ti GPU cards and a batch size of 60 per card with a total 5 training epochs for pre-training and fine-tuning. We choose the best model based on the performance over Dev set of ChID. The initial learning rate is set to be $5e^{-5}$ with 10% warm-up steps. We use the optimizer *AdamW* in accordance with a linear learning rate scheduler. We choose 128 as the maximum length and we truncate passages longer than this limit by keeping only the 128 characters surrounding `[MASK]`, with `[MASK]` in the middle. Our code has been released online as ChengyuBERT[10].

4.2 Results on ChID-Official

We first conduct experiments using the ChID-Official dataset. We try to answer the following research questions using the ChID-Official dataset. **R1**: Does our two-stage model perform better than previous methods? **R2**: Are both stages of training in our model necessary? **R3**: For the objective function shown in Eqn. (6), do we need both $L_{\mathcal{V}}$ and $L_{\mathcal{A}}$?

In order to answer R1, we compare our model with the following baselines: **LM** uses a bidirectional LSTM language model to compute the hidden representation of the blank from both forward and backward directions and then concatenates the two hidden states as the final representation for the

[8] https://github.com/bazingagin/chengyu_data
[9] http://zaojv.com
[10] https://github.com/VisualJoyce/ChengyuBERT

blank. **AR** is the attentive reader model [Hermann et al. 2015] and **SAR** is the Stanford attentive reader model [Chen et al. 2016]. AR and SAR use different attention mechanisms over the context when computing the attention-based representation for the blank. All three models use Chengyu embeddings and are supervised using a loss function the same as $L_{\mathcal{A}}$. LM, AR and SAR are all methods implemented and reported in [Zheng et al. 2019].

In addition, we implemented a baseline that uses Chinese-BERT-wwm-ext directly for Chengyu recommendation. In this baseline, which we call **BERT-BL**, We first concatenate each candidate Chengyu in characters with the given context passage by a special token `[SEP]` to construct a single sequence and feed it into BERT. Then we fine-tune a linear classifier over the hidden representations of `[CLS]` of each candidate sequence to choose the best one as the choice. We also show human performance as a reference point. Finally, we refer to our complete two-stage model as **Two-Stage**.

In order to answer R2, we consider the following degenerate versions of our model: **w/o Pre-Training:** In this version of our model, we do not perform pre-training and directly use Chinese-BERT-wwm-ext for the second stage of fine-tuning. **w/o Fine-Tuning:** In this version of our model, we directly use the pre-trained Chengyu-BERT and the Chengyu embeddings for Chengyu recommendation. We first rank all Chengyu in the vocabulary $\mathcal{V}$ based on the pre-trained Chengyu-BERT, and then pick the candidate in $\mathcal{A}$ that is ranked the highest as the answer.

In order to answer R3, we consider another two degenerate versions of our model: **w/o $L_{\mathcal{V}}$:** In this version, we exclude $L_{\mathcal{V}}$ in the objective function Eqn. (6). **w/o $L_{\mathcal{A}}$:** In this version, we exclude $L_{\mathcal{A}}$ in the objective function Eqn. (6).

We use accuracy as our performance metric. Here Accuracy is defined as the percentage of test examples where the recommended Chengyu is the same as the ground truth candidate Chengyu.

Model		Dev	Test	Ran	Sim	Out
Human	[Zheng et al. 2019]	-	87.1	97.6	82.2	86.2
LM	[Zheng et al. 2019]	71.8	71.5	80.7	65.6	61.5
AR	[Zheng et al. 2019]	72.7	72.4	82.0	66.2	62.9
SAR	[Zheng et al. 2019]	71.7	71.5	80.0	64.9	61.7
BERT-BL		79.33	79.42	88.84	72.93	73.11
Two-Stage		85.43	85.36	**95.04**	78.74	82.03
w/o Pre-Training		81.87	81.75	92.87	74.13	71.97
w/o Fine-Tuning		81.12	81.26	92.52	74.06	79.94
w/o $L_{\mathcal{V}}$		**86.15**	**86.31**	94.25	**80.54**	**83.52**
w/o $L_{\mathcal{A}}$		84.76	84.62	94.83	77.69	80.84

Table 4. The experiment results in terms of accuracy on ChID-Official. The metric used in this task is accuracy for multiple-choice problems.

The results are shown in Table 4. For Human, LM, AR and SAR, the performance shown in the table is taken directly from [Zheng et al. 2019]. We can observe the following from the table. (1) Our **Two-Stage** model can substantially outperform all the baselines. This shows the effectiveness of our two-stage model and the usefulness of our collected unlabeled Chinese corpus for pre-training. (2) The performance of **Two-Stage** is also clearly higher than the two degenerate versions **w/o Pre-Training** and **w/o Fine-Tuning**. This shows that both stages of training are critical for us to achieve the optimal performance. (3) Comparing the performance of **w/o $L_{\mathcal{V}}$**, **w/o $L_{\mathcal{A}}$** and our complete model, we can see that $L_{\mathcal{A}}$ is more critical. We do observe that in most cases, whether or not to include $L_{\mathcal{V}}$ does not make any substantial difference. For the split **Sim**, which uses near-synonyms as candidate answers, using $L_{\mathcal{A}}$ can improve the performance with a significant margin than using $L_{\mathcal{V}}$ only. But for the test set **Ran**, which uses randomly selected wrong candidate answers, using **Two-Stage** performs slightly better than $L_{\mathcal{A}}$. We believe this is because when the wrong candidate answers are randomly chosen, these wrong answers are no longer near-synonyms to the correct answer, and therefore $L_{\mathcal{A}}$ is kind of similar to $L_{\mathcal{V}}$.

Overall, the experiments on ChID-Official show that our two-stage model is indeed very effective for this task, and both stages of training are critical.

4.3 Results on ChID-Competition

To further test the competency of our model, we next evaluate the model on **ChID-Competition**. There are some differences between ChID-Official and ChID-Competition, which we have detailed earlier. Because in ChID-Competition multiple contexts are considered together with the same set of candidates, we use some heuristic methods to post-process the predictions in order to globally optimize the results.

Table 5 shows the comparison between our model and the top systems on the leaderboard. In the first part of the table, we show the top-3 systems on the competition leaderboard.[11] In the second part of the table, we list several other pretrained language models extracted from the benchmark CLUE [Xu et al. 2020][12]. Because of the special settings of ChID-Competition, we find that removing $L_{\mathcal{A}}$ helps the performance on ChID-Competition, so we also show the performance of **w/o** $L_{\mathcal{A}}$. We can see that our **Two-Stage** model can still achieve consistently better performance than the top 3 systems submitted to the leaderboard, and the **w/o** $L_{\mathcal{A}}$ setting works even better. This shows again that our model indeed works better than other existing methods on the ChID dataset.

Model	Dev	Test	Out
Top-1 (wssb)	88.35	90.57	85.54
Top-2 (On The Road)	90.59	91.35	84.93
Top-3 (Beenle)	81.94	89.27	84.72
ERNIE-base	82.46	82.28	-
ALBERT-base	70.99	71.77	-
XLNet-mid	83.76	83.47	-
RoBERTa-large	85.31	84.50	-
RoBERTa-wwm-large-ext	85.81	85.37	-
Two-Stage	91.19	91.14	89.40
w/o $L_{\mathcal{A}}$	**92.41**	**91.98**	**90.22**

Table 5. Experiment results for ChID-Competition. Here we include the top submissions on the leaderboard.

4.4 Results on CCT

We further use the CCT [Jiang et al. 2018] dataset to evaluate our model. Note that the CCT dataset covers more Chengyu than ChID. Note also that although the number of Chengyu in CCT is large, CCT does not have enough contexts for each Chengyu and is thus not suitable for further fine-tuning. Therefore, here we directly use the pre-trained Chengyu-BERT for Chengyu

[11] We show the top-3 systems on the leaderboard as of the submission date of this paper.
[12] https://github.com/CLUEbenchmark/CLUE

recommendation on CCT. We also add a setting to CCT where 7 candidates are considered for each context instead of 4 (which is the original setting).

Model	Candidates	Performance
Human [Jiang et al. 2018]	4	70.0
BiLSTM [Jiang et al. 2018]	4	89.5
Pre-training	4	**93.7**
Pre-training	7	90.5

Table 6. Evaluation on CCT.

Table 6 shows the results. We can see from the table that our two-stage model again can outperform the baseline performance reported in [Jiang et al. 2018].

4.5 Error Analysis

To better understand where our method fails, we conduct a detailed error analysis over the ChID-Official dataset. Specifically, we randomly select 200 examples from the evaluation data where our predictions are different from the ground truth answers. We manually go through these examples to understand the reasons behind the wrong predictions, and we group the examples into a few categories, as shown in Table 7.

We now explain the different categories of errors that we have identified:

Violation of Syntactic Rules: Chinese Chengyu also need to follow syntactic rules. Given a particular context, some candidate Chengyu are not suitable simply because they do not syntactically fit into the context. For example, the two candidates in row *Syntactic Error* in Table 7 both refer to an unbelievable state or achievement. However, the local contextual words "______地进行" require a Chengyu that can serve as an adverb. "登峰造极" usually is not used as an adverb, making "神乎其神" the correct answer.

Inconsistency: While grammatically two Chengyu may both be suitable for the blank locally, once taking the full context into account, some Chengyu can become less suitable or even strange, causing inconsistency in meaning. Two common reasons for inconsistency are *Logical Error* and *Sentiment Error*.

For the *Logical Error* example in Table 7, when we just look at the local context of the blank, where the crow introduces itself to the cuckoo, either of the two candidates ("快人快语" and "敢作敢为") is obviously a good choice. Once the cuckoo mentions "speak" in its reply, to be consistent, "快人快语" (which is about talking) would be the more suitable answer than "敢作敢为" (which is about taking actions).

While most Chinese Chengyu are neutral, some may carry sentiment of a particular polarity. In such cases, it is important to choose an idiom whose sentiment fits the context. For the *Sentiment Error* example in Table 7, "文质彬彬" and "道貌岸然" both indicate somebody being calm and polite. However, "文质彬彬" is usually used to praise a person acting like a gentleman while "道貌岸然" is a negative idiom to describe a hypocritical person. As the context uses words such as "suddenly assumed" with cues of negative sentiment, "道貌岸然" is more suitable than "文质彬彬" here.

Synonym and Non-Synonym: For the remaining errors, we find that based on our understanding, the predicted idiom may also be suitable for the passage, and therefore they may not be considered to be real errors. We further separate these into "synonyms" and "non-synonyms",

Category	Count	%	Example
Syntactic Error	23	11.5	更有网友将“光棍节”与其他节日进行对比，______地进行日期的主题研究，从而得出”惊人“结论：“男人节是8·3，妇女节是3·8，他们相加就是11·11，光棍节就这样诞生了！Somebody online took “the singles day" and other festivals for comparison, ______ researched the date, thus came to a surprising conclusion: men’s day is 8 • 3, women’s day is 3 • 8, their sum is 11 • 11, “the singles day" was born! ● 神乎其神：magical, magically ○ 登峰造极：outstanding
Logical Error	69	34.5	乌鸦答道：“我乃乌鸦，______。”布谷鸟说：“谨向你致意，望你说话永远这样直爽。至于我，呼唤声调必须悠扬。” The crow replied, “I am a crow, ______." “With all due respect," said the cuckoo, “Salute, hope you always **speak** so straightforward. As for me, the call must be melodious." ● 快人快语：straight talk from an honest man ○ 敢作敢为：act with courage and determination
Sentiment Error	11	5.5	一见到这位警长，他便从九天之外回到地面上来了，于是他的脸上马上摆出了一副______的样子，说道，那“信我看过了，先生，您办得很对，应该把那个人逮起来。现在请你告诉我，你有没有搜有到有关他造反的材料？” The sight of the sheriff brought him back to reality, and his face **suddenly assumed** a ______ look, ... ○ 文质彬彬：be gentle ● 道貌岸然：be sanctimonious
Synonym	25	12.5	哈娜姐近来很喜欢在自己的头部造型下功夫，每次都很______。Rihanna has been working on her head lately, every time is so ______. ● 出人意表：beyond expectations ○ 出人意料：beyond expectations
Non-Synonym	56	28.0	协议规定住宿纳入他们公司统一管理，他们在其宿舍墙壁上张贴了《管理规定》，上面______地写着，严禁在宿舍内聚餐、饮酒等不健康行为。Under the agreement, accommodation is subject to the unified management of their company, and they have posted management rules on the walls of their dormitories, which ______ state that unhealthy behaviors such as sharing meals, drinking are strictly prohibited. ● 明明白白：extremely clear ○ 白纸黑字：clearly (written)
Misuse	16	8.0	院墙有的残垣断壁，有的只是用树枝夹起围成的栅子，那栅子也不知挺了多少年，______，缺胳膊断腿。Some of the courtyard walls are in ruins, some are only grids built from branches, the grids have been barely standing for years, ______, missing arms and legs. ● 前仰后合：laugh oneself into convulsions ○ 东倒西歪：lying on all sides

Table 7. Different categories of errors and their distribution. In each example, the candidate answer shown with a solid circle is the ground truth answer.

depending on whether the predicted answer is a synonym with the ground truth answer or not. In the case when the predicted answer is not a synonym of the ground truth answer, the predicted answer may still be suitable for the context because there is not sufficient context to support that the ground truth answer is a better choice.

Misuse: Finally, we also observe that in some cases the ground truth answer, which is the Chengyu used in the original text, is actually a misuse of the Chengyu. This could happen if the writer of the original text has misunderstanding of the Chengyu. Since the original text comes from the Web and we cannot guarantee the literacy level of the writers, misuse of Chengyu does happen occasionally in the original corpus. An exmaple is shown in Table 7.

Our error analysis suggests the following: (1) A significant percentage (40%) of errors may not be real errors. This suggests that the original ChID dataset could potentially be further improved by providing multiple correct answers. (2) The most common errors are logical errors, which require reasoning to correct. It is generally known that reasoning is a challenging problem in training neural network models for language understanding. For Chinese idiom comprehension, we can see that there is still much room for improvement when we deal with Chengyu that require reasoning to understand.

5 CHENGYU EMBEDDINGS FOR EMOTION AND SENTIMENT PREDICTION

We suspect that the Chengyu embedding vectors learned by our pre-training stage may be valuable for other tasks. To test this hypothesis, we choose a Chengyu emotion and sentiment prediction task. Previously, Wang and Yu [2010] attempted to use lexicons from the CIKB database to build a feature-based SVM to predict the sentiment label for a Chengyu. Since CIKB is not available online, we use Chinese Affective Lexicon Ontology (CALO) [Yu and Jianmei 2008] for our emotion and sentiment prediction task.

	Emotion			Sentiment
	Coarse-grained	Fine-grained	Intensity	
可歌可泣	good (好)	praise (赞扬)	7	appreciative (褒义)
东拼西凑	disgust (恶)	reproach (贬责)	3	derogatory (贬义)
欢天喜地	enjoyment (乐)	pleasure (快乐)	7	appreciative (褒义)
撼天动地	surprise (惊)	surprise (惊奇)	7	neutral (中性)

Table 8. Examples of emotion labels for some Chengyu in CALO.

CALO was created with the purpose of supporting textual Affective Computing (AC) in Chinese language. The construction of CALO was based on mainstream emotional classification research [Ekman 1992] in combination with conventional Chinese emotion categories. Six categories, anger (怒), fear (惧), sadness (哀), enjoyment (乐), disgust (恶) and surprise (惊), are used and consistent with [Ekman 1992]. However, enjoyment (乐) is not sufficient to describe some positive emotions like respect and belief, so an extra category, "good" (好) was added. There are therefore 7 main categories in CALO. Each main category was further classified into different numbers of subcategories according to their intensity and complexity. There are 21 subcategories in total in CALO. Each entry in CALO is a Chengyu that has an emotion label from the subcategories.

In addition, we also consider three general labels, namely, appreciative, derogatory and neutral, to indicate the general sentiment of a Chengyu. Ground truth of these labels for different Chengyu are also found in the CALO dataset.

We take those Chengyu for which we are able to train Chengyu embeddings and which have entries in CALO. This gives us 14,361 Chengyu, a comparable size with that of [Wang and Yu 2010]. The statistics are shown in Table 9. We randomly split the Chengyu from CALO into training and testing sets by keeping the testing set size to 3000. Note that the distribution of CALO is skewed to non-neutral sentiments.

We use the Chengyu embeddings learned from our pre-training to predict the Chengyu sentiments and emotions. For the baseline method, we treat each Chengyu as a "sentence" and extract features of the hidden vectors $\mathbf{h}_{[CLS]}$ of `[CLS]` using Chinese-BERT-wwm-ext. For each emotion or sentiment prediction task, we use a SVM to predict the emotion or sentiment label of each Chengyu. In order to test whether our learned Chengyu embeddings are useful for emotion detection, we concatenate

	CIKB		CALO	
	Train	Test	Train	Test
Appreciative (A)	6,967	1,011	4,937	1,305
Neutral (N)	8,216	1,100	1,731	458
Derogatory (D)	4,817	889	4,678	1,237

Table 9. The sentiment distribution on the prediction task from CIKB and CALO.

the $\mathbf{h}_{[CLS]}$ with the learned Chengyu embedding $\mathbf{e}$ and feed the vector into SVM to predict a label. We train the model and report the label accuracy (ACC) and macro average F1 scores as shown in Table 10. We can see from the table that our performance is clearly better than the baselines. This demonstrates the value of the Chengyu embeddings that we have learned.

	Emotion				Sentiment	
	Coarse-grained		Fine-grained			
	ACC	F1	ACC	F1	ACC	F1
`[CLS]`	62.2	45.2	48.2	25.5	64.9	56.8
`[CLS]` + Our Learned Embeddings	**63.9**	**46.1**	**49.0**	**26.3**	**66.3**	**57.7**

Table 10. The emotion prediction results on CALO.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a BERT-based two-stage model for Chinese Chengyu recommendation. Our model pre-trains a Chengyu-oriented BERT over a large Chinese corpus we have collected for open-ended Chengyu recommendation. It then fine-tunes the pre-trained Chengyu-BERT for multiple-choice Chengyu recommendation. Experiments showed that our proposed two-stage model could achieve the state of the art on both ChID and CCT datasets. We also conducted ablation studies to test the effectiveness of the two stages, and found both to be useful.

In the future, we plan to look into the interpretability of neural network models for Chengyu comprehension, especially to understand how neural network models are able to tell the difference between a Chengyu and its near-synonyms.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments and suggestions. This research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 2358–2367. https://doi.org/10.18653/v1/P16-1223

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019a. Pre-Training with Whole Word Masking for Chinese BERT. arXiv:1906.08101 [cs.CL]

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019b. A Span-Extraction Dataset for Chinese Machine Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 5883–5889. https://doi.org/10.18653/v1/D19-1600

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised Sequence Learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'15)*. MIT Press, Cambridge, MA, USA, 3079–3087. http://dl.acm.org/citation.cfm?id=2969442.2969583

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Xingyi Duan, Baoxin Wang, Ziyue Wang, Wentao Ma, Yiming Cui, Dayong Wu, Shijin Wang, Ting Liu, Tianxiang Huo, Zhen Hu, and et al. 2019. CJRC: A Reliable Human-Annotated Benchmark DataSet for Chinese Judicial Reading Comprehension. *Chinese Computational Linguistics* (2019), 439–451. https://doi.org/10.1007/978-3-030-32381-3_36

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion* 6, 3-4 (1992), 169–200.

Chikara Hashimoto, Satoshi Sato, and Takehito Utsuro. 2006. Japanese Idiom Recognition: Drawing a Line between Literal and Idiomatic Meanings. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Association for Computational Linguistics, Sydney, Australia, 353–360. https://www.aclweb.org/anthology/P06-2046

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 1693–1701. http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.pdf

Wan Yu Ho, Christine Kng, Shan Wang, and Francis Bond. 2014. Identifying Idioms in Chinese Translations.. In *LREC*. 716–721.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 328–339. https://doi.org/10.18653/v1/P18-1031

Zhiying Jiang, Boliang Zhang, Lifu Huang, and Heng Ji. 2018. Chengyu Cloze Test. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, New Orleans, Louisiana, 154–158. https://doi.org/10.18653/v1/W18-0516

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics* 8 (2020), 64–77. https://doi.org/10.1162/tacl_a_00300

Graham Katz and Eugenie Giesbrecht. 2006. Automatic Identification of Non-Compositional Multi-Word Expressions using Latent Semantic Analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*. Association for Computational Linguistics, Sydney, Australia, 12–19. https://www.aclweb.org/anthology/W06-1203

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1eA7AEtvS

Haizhou Li and Baosheng Yuan. 1998. Chinese Word Segmentation. In *Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation*. Chinese and Oriental Languages Information Processing Society, Singapore, 212–217. https://doi.org/2065/12081

Dekang Lin. 1999. Automatic Identification of Non-compositional Phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, College Park, Maryland, USA, 317–324. https://doi.org/10.3115/1034678.1034730

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

Yuanchao Liu, Bo Pang, and Bingquan Liu. 2019b. Neural-based Chinese Idiom Recommendation for Enhancing Elegance in Essay Writing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5522–5526. https://doi.org/10.18653/v1/P19-1552

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119. http://papers.nips.cc/

paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing (CICLing '02)*. Springer-Verlag, Berlin, Heidelberg, 1–15.

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2018a. DRCD: a Chinese Machine Reading Comprehension Dataset. arXiv:1806.00920 [cs.CL]

Yutong Shao, Rico Sennrich, Bonnie Webber, and Federico Fancellu. 2018b. Evaluating Machine Translation Performance on Chinese Idioms with a Blacklist Method. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan. https://www.aclweb.org/anthology/L18-1005

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 05 (Apr. 2020), 8968–8975. https://doi.org/10.1609/aaai.v34i05.6428

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 1556–1566. https://doi.org/10.3115/v1/P15-1150

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

Lei Wang and Shiwen Yu. 2010. Construction of Chinese Idiom Knowledge-base and Its Applications. In *Proceedings of the 2010 Workshop on Multiword Expressions: from Theory to Applications*. Coling 2010 Organizing Committee, Beijing, China, 11–18. https://www.aclweb.org/anthology/W10-3703

Shuohang Wang and Jing Jiang. 2017. A Compare-Aggregate Model for Matching Text Sequences. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. https://openreview.net/forum?id=HJTzHtqee

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BJgQ4lSFPH

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese Language Understanding Evaluation Benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 4762–4772. https://www.aclweb.org/anthology/2020.coling-main.419

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*. 5754–5764.

Xu Linhong Lin Hongfei Pan Yu and Ren Hui Chen Jianmei. 2008. Constructing the Affective Lexicon Ontology [J]. *Journal of the China Society for Scientific and Technical Information* 2 (2008), 6.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1441–1451. https://doi.org/10.18653/v1/P19-1139

Chujie Zheng, Minlie Huang, and Aixin Sun. 2019. ChID: A Large-scale Chinese IDiom Dataset for Cloze Test. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 778–787. https://doi.org/10.18653/v1/P19-1075