5-2022

Chinese idiom understanding with transformer-based pretrained language models

Minghuan TAN
Singapore Management University, mhtan.2017@phdcs.smu.edu.sg

CHINESE IDIOM UNDERSTANDING WITH TRANSFORMER-BASED PRETRAINED LANGUAGE MODELS

MINGHUAN TAN

SINGAPORE MANAGEMENT UNIVERSITY
2022

Chinese Idiom Understanding with Transformer-based Pretrained Language Models

by

Minghuan TAN

Submitted to School of Computing and Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Computer Science

Dissertation Committee:

Jing JIANG (Supervisor / Chair)
Professor of School of Computing and Information Systems
Singapore Management University

Yuan FANG
Assistant Professor of School of Computing and Information Systems
Singapore Management University

Wei GAO
Assistant Professor of School of Computing and Information Systems
Singapore Management University

Aixin SUN
Associate Professor of School of Computer Science and Engineering
Nanyang Technological University

Singapore Management University
2022

I hereby declare that this PhD dissertation is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in this dissertation.

This PhD dissertation has also not been submitted for any degree in any university previously.

Minghuan Tan

20 May 2022

Chinese Idiom Understanding with Transformer-based Pretrained Language Models

Minghuan TAN

Abstract

In this dissertation, I study the understanding of Chinese idioms using transformer-based pretrained language models. By "understanding", I confine the topics to word embeddings learning, contextualized word representations learning, multiple-choice cloze-test reading comprehension and conditional text generation.

Chinese idioms are fixed phrases that have special meanings usually derived from an ancient story. The meanings of these idioms are oftentimes not directly related to their component characters, which makes it hard to model them compared with standard phrases whose meanings are compositional.

I initiate the work with studying idiom representations derived from pretrained language models, in particular, BERT. We adopt probing-based methods to investigate to what extent BERT can encode an idiom's meaning. We design two probing tasks to test whether idiom encodings through pretrained language models can be used to (1) classify the usage of a potential idiomatic expression as either idiomatic or literal and (2) identify idiom paraphrases. Then we propose a BERT-based method to better learn Chinese idioms' embeddings and evaluate the embeddings using our newly constructed dataset of Chinese idiom synonyms and antonyms.

I further study Chinese idiom prediction based on a context. We first propose a BERT-based dual embedding model for the Chinese idiom prediction task, where given a context with a missing Chinese idiom and a set of candidate idioms, the model needs to find the correct idiom to fill in the blank. Our method is based on the observation that part of an idiom's meaning comes from a long-range context that contains topical information, and part of its meaning comes from a local context that encodes more of its syntactic usage. We use BERT to process the contextual

words and to match the embedding of each candidate idiom with both the hidden representation corresponding to the blank in the context and the hidden representations of all the tokens in the context through context pooling. We also propose to use two separate idiom embeddings for the two kinds of matching. Experiments on ChID, a recently released Chinese idiom cloze test dataset, show that our proposed method performs better than existing state of the art. Ablation experiments also show that both context pooling and dual embedding contribute to the performance improvement. Observing some of the limitations with existing work, we further propose a two-stage model, where during the first stage we retrain a Chinese BERT model by masking out idioms from a large Chinese corpus with a wide coverage of idioms. During the second stage, we fine-tune the retrained, idioms-oriented BERT on a specific idiom recommendation dataset. We evaluate this method on the ChID dataset and find that it can achieve the state of the art. Ablation studies show that both stages of training are critical for the performance gain.

I also propose a new task called Chengyu-oriented text polishing. This task is based on the hypothesis that using Chengyu properly usually can enhance the elegance and conciseness of the Chinese language. We formulate the task as a context-dependent text generation problem and construct a dataset with 1.5 million automatically generated instances for training and 4K human-annotated examples for evaluation. The study offers solid baselines built with the latest pretrained encoder-decoder transformer models.

I finally conclude the thesis by summarizing the contributions of this thesis and pointing out potential future directions to explore related to Chinese idiom understanding, namely, sentiment analysis with idioms and explaining Chinese Chengyu recommendation models.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation studies Chinese Idiom Understanding using transformer-based pretrained language models. We are motivated to study this topic from the following perspectives: (1) Idiomaticity and compositionality are two closely related terms in the context of Multiword Expressions (MWEs), which are critical for the purposes of fluency, robustness and better understanding of natural languages. (2) Despite the success of pretrained language models in a wide range of NLP tasks, it is still largely under-explored whether these models capture meanings accurately, especially idiomatic meanings, and how pretrained language models may help the construction of representations for idioms. (3) Considering the prevalent usage of Chinese idioms, we are also curious about whether or not pretrained language models can help with their representations and to what extent these representations can improve the understanding of Chinese idioms.

To achieve these goals, we base our work on pretrained language models like BERT [31] to study topics of BERT-based idiom representations and their applications on downstream tasks like Chinese idiom recommendation and intelligent writing assistance. To begin with, we conduct a probing-based empirical study on BERT encodings of idioms. Then we explore how to improve Chinese idiom representations by using BERT as the contextualized encoder. Moving to applications, we investigate optimizing idiom representations for Chinese idiom recommendation.

Finally, we look into a new task of intelligent writing assistance called Chengyu-oriented text polishing.

1.1 Multiword Expressions

In the literature, researchers give different definitions [90, 15, 26, 106, 8] for Multiword Expressions. In this work, we adopt the definition by Baldwin and Kim [7], which is further extended by Ramisch [104] for the purpose of MWEs acquisition. However, we loose their last condition to permit whether or not using it as a single unit.

Definition 1 (Multiword Expressions) *Multiword expressions (MWEs) are lexical items that:*

1. *are decomposable into multiple lexemes,*
2. *present idiomatic behaviour at some level of linguistic analysis,*
3. *can be treated as a unit at some level of computational processing.*

In the context of MWEs defined as above, we directly refer the definition of idiomaticity and compositionality from Baldwin and Kim [7] as:

Definition 2 (Idiomaticity) *Idiomaticity refers to markedness or deviation from the basic properties of the component lexemes, and applies at the lexical, syntactic, semantic, pragmatic, and/or statistical levels.*

Definition 3 (Compositionality) *Compositionality refers to the degree to which the features of the parts of a MWE combine to predict the features of the whole.*

From these definitions, an MWE can present different values with different levels of linguistic properties. Specifically, lexical level indicates word formation and inflection, syntactic level deals with word order and grammar rules, semantic level is related to meaning, pragmatic level bonds to specific situations and statistical

level reflects conventionalisation when using a language. Both Baldwin and Kim [7] and Ramisch [104] use a table to illustrate how these properties distribute in different samples, see Table 1.1. It would actually be more accurate to describe the degree of idiomaticity in a continuum, ranging from totally compositional and fully predictable [104].

	Lexical	Syntactic	Semantic	Pragmatic	Statistical
ad hoc	+	+			+
every now and then		+	+		+
social butterfly			+		+
all aboard				+	+
yellow dress					

Table 1.1: Distribution of idiomaticity in binary value across all levels for examples of expressions.

However, this definition doesn't cover expressions that are semantically compositional and contain idiomatical meanings at the same time. For example, "spill the beans" may refer to the idiomatical meaning of "disclose the secrete" or be used just literally in some context. These expressions with ambiguity are called potentially idiomatic expressions [47].

The prevalence of MWEs in all text genres poses significant challenges across different tasks in natural language processing [106]. These include but not limit to sentiment analysis [145], automatic spelling correction [57] and machine translation [59]. Despite the development of pretrained language models, there are still limited understanding of how these models may handle representation of MWEs. A series of works are proposed to investigate phrase composition from their contextualized representations. Yu and Ettinger [155] conduct analysis of phrasal representations in state-of-the-art pretrained transformers and find that phrase representation in these models still relies heavily on word content, showing little evidence of nuanced composition. Shwartz and Dagan [113] confirm that contextualized word representations perform better than static word embeddings, more so on detecting meaning shift than in recovering implicit information. Therefore, it remains a challenging

problem to resolve the idiomaticity of phrases.

To solve issues introduced by MWEs, many areas in NLP benefit from treating MWEs as single lexical units [68], including parsing [25], machine translation [25, 16], keyphrase/index term extraction [89], and language acquisition research [35]. However, these does not help reducing challenges of learning meaning representations of idiomaticitical expressions. Some recent approaches are trying to further diagnose pretrained language models using new metrics and datasets. Garcia et al. [40] analyse different levels of contextualisation to check to what extent models are able to detect idiomaticity at type and token level. Garcia et al. [41] propose probing measures to assess Noun Compound (NC) idiomaticity and conclude that idiomaticity is not yet accurately represented by contextualised models. AStitchInLanguageModels [125] design two tasks to first test a language model's ability to detect idiom usage, and the effectiveness of a language model in generating representations of sentences containing idioms.

1.2 Chinese idioms

A Chinese idiom, like all idioms in other languages, is a Multiword (character) expression which has a figurative meaning that may not be derivable from its constitute components [135]. In the literature, Chinese idioms generally include set phrases (also known as Chengyu (成语) (e.g., "高朋满座", which refers to the presence of distinguished guests), institutionalized expressions (俗语) (e.g., "滚雪球", which literally means rolling of a snow ball but also means make something bigger and bigger, such as continuously making profits), proverbs (谚语) (e.g., "亡羊补牢，为时未晚", which means it is never too late to mend the sheepfold when a sheep is lost), two-part allegorical sayings (歇后语) ("八仙过海——各显神通", which means eight immortals crossing the sea, and everybody shows his talent)), and maxims (格言) (e.g., "天行健，君子以自强不息", which means as heaven maintains vigor through movements, a gentle man should constantly strive

for self-perfection) [137].

Despite the ongoing discussions over categorizations of Chinese idioms and distinguishing of Chengyu from other types of Chinese idioms [159], Chengyu have the following properties that make it more prominent as a research topic:

1. Fixedness. Chengyu usually have fixed forms in structure that the component characters (mostly four) cannot be changed.

2. Stability. While many expressions acquire an idiomatic meaning over time [14] and new idioms come into existence on a daily basis, Chengyu are more stable than other types of idioms in their forms and meaning. Xiaobing and Lina [149] analyzed annual use of idioms in National Language Resource Monitoring Corpus (print media) in the years 2006-2008 and provided the an analysis on the stability of idioms from a synchronic-diachronic perspective.

3. Idiomaticity. The meaning of each Chengyu may not be literally understood through the composition of its characters, especially for those which are derived from historical stories or formulated using ancient Chinese grammars. For example, "一定不易" is literally interpreted as *"it must be not easy"* in modern Chinese. However, the idiom uses the ancient grammar and word sense, and the idiomatic meaning is actually *"once decided never change"*, which is not even close to the literal meaning.

The fixedness and stability properties are very helpful to make use of large-scale online resources and published dictionaries with low annotation costs. Considering idiomaticity, the usage of Chengyu still poses a challenge on language understanding not only for humans but also for artificial intelligence.

In this thesis, we focus more on the meaning representations and contextual fitness of Chengyu. We will research three major tasks on Chengyu: (1) learning static representations of Chengyu as part of the Chinese vocabulary, (2) Chinese Chengyu Recommendation [61, 80], which is trying to recommend a best Chengyu

given a specific context. (3) A special case in intelligent writing assistance called Chengyu-oriented text polishing.

1.3 Neural Network Models for Language Understanding

Language understanding has been largely improved by neural language modeling (NLM) [11] compared to statistical language modeling (SLM) [67, 18, 43]. The main advantage of NLM lies in the expressive ability of deep neural networks to automatically learn syntactic and semantic features regardless of the sparsity of data, which is incompetent for n-gram-based SLM.

The development of NLM induces better representations for natural language at word level and subsequently sentence level. Distributed word representations, or word embeddings, are typically built upon neural language models [11] which can capture semantic similarities and relatedness among different words. The topic has been explored extensively [22, 130, 88, 87, 85] to generate better dense representations about words. At sentence level, recurrent neural networks (RNNs) [56] and convolutional neural networks (CNNs) [65] can deal with text sequences and compose the meaning of sentences into vectors from embeddings of separate words.

Built upon the neural network modeling framework, language understanding thrives in a series of tasks. In neutral machine translation (NMT), the encoder-decoder framework [20, 119] and attention mechanism [4, 82] largely improve the performance. In machine reading comprehension (MRC), newer neural architectures like Match-LSTM [138, 140], Bi-Directional Attention Flow (BiDAF) [109] and Gated-Attention Reader [32] show stronger abilities in understanding passages compared to traditional retrieval-based methods. These tasks influence each other and borrow structures from each other, contributing to a large pool of available building blocks for more complex neural models.

Language model pretraining has been proven to be effective over a list of natural language tasks at both sentence level [13] and token level [128, 103]. Existing strategies of using pretrained language models include feature-based methods like ELMO [96] and fine-tuning methods such as OpenAI GPT [100] and BERT [31]. Specifically, ELMO pretrains a bidirectional language model (biLM) which can be extracted as high quality deep context-dependent features and loaded for training for other tasks. GPT and BERT are both constructed upon Transformer [131]. GPT adopts a single directional language model while BERT uses a multi-layer bidirectional language model. The BERT model is pretrained over a large corpus with self-supervised methods using Masked Language Model (MLM) task and the Next Sentence Prediction (NSP) task. Bert-based fine-tuning strategy and its extensions [27, 152, 79] are pushing performance of neural models to near-human or super-human level.

More recently, a series of variations of BERT are proposed for better performance and wider language coverage. To strengthen BERT, these models can be grouped into masking-based approaches and structural-based approaches. Masking-based approaches include whole word masking (WWM) in BERT [31], masking random contiguous spans in SpanBERT [63] and dynamic masking proposed in RoBERTa [79]. Structural-based approaches focus on the relationships among segments of sentences. For example, the NSP task is a structural prediction task that a binary classification for predicting whether two segments follow each either in the original text. Under further ablation studies, NSP is either removed due to inconsistent improvement in XLNET [152] and SpanBERT [63], or restricted to use sentences from a single document in RoBERTa [79]. More structure-aware pretraining tasks are proposed by ERNIE [116, 117], StructBERT [141] and ALBERT [71]. There are also transformer-based models supporting multiple languages, like mBERT [31] and XLM-R [24].

Besides GPT, there are also transformer-based pretrained language models with encoder-decoder structures for text generation, like T5 [102] and BART [73]. These

generative models can even match the performance of RoBERTa on GLUE [133] and SQuAD [103], and achieves new state-of-the-art results on a range of tasks like abstractive dialogue, question answering, and summarization.

In this thesis, we construct our models mainly over transformer-based pretrained language models and explore to both understand idiom representations of these models and improve performance of idiom-oriented tasks.

1.4 Neural Network Models for Chinese Idiom Understanding

Chinese idioms are catching attentions from neural network researchers but are usually treated as more challenging language units which are not modelled directly in their networks. These idioms are processed via extra procedures after querying an idiom dictionary or database. In neural machine translation (NMT), researchers will add idioms to a blacklist [111] or prepare an idiom corpus [55]. In Chinese word segmentation (CWS), common practices are replacing idioms with one single token [161] before using neural network models.

Recently, neural networks that explicitly model Chinese idioms appear in the task of Chinese Chengyu Recommendation (CCR). In the literature, Chengyu Cloze Test [61] builds the first Chengyu recommendation system to assist Chinese learners. Chinese Idiom Recommendation [80] addresses automatically recommending idioms because remembering idioms is difficult for most people. The fixedness of Chengyu are being addressed to simplify the modeling. Liu et al. [80] reformulate the CCR problem as context-to-idiom machine translation problem but set the target max length to four characters. Both Jiang et al. [61] and Zheng et al. [165] formulate the CCR task as a cloze-style test where they assume idioms in the context have been identified. Specifically, Jiang et al. [61] incorporate two BiLSTM networks to encode the definition of Chengyu and the context sentence separately followed

by computing bilinear attentions following Stanford Attentive Reader (SAR) [17]. Zheng et al. [165] constructs the first large scale Chengyu cloze-test dataset ChID and offers strong baselines using attentive readers [51, 17]. ChID is now used by many Chinese pretrained language models as a benchmark dataset in the evaluation of Chinese language understanding.

Given the prevalence usage of idioms in the Chinese language, idiom usage has been a good sign of better expressiveness and is generally considered to be effective in enhancing elegance in writing [78, 80]. However, there is still a lack of datasets and baselines in polishing a context without Chengyu into one that with. It will be an interesting idea to consider the text generation task of Chengyu-oriented text generation.

Chengyu, as the representative of Chinese idioms in this thesis, present extra challenges that many Chengyu are very similar but not identical in meanings, which are called near-synonyms [165]. To recommend the best Chengyu, the model needs to learn the nuances to differentiate these near-synonyms. Another challenge is the skewed distribution of different Chengyu which may lead to the imbalanced corpus used for training that rare Chengyu cannot get a good representation. Finally, compared to common words or phrases, a large portion of Chengyu carry emotions that may affect the meaning of the context.

1.5 Dissertation Structure

We first discuss related works that are connected to the topics of this thesis in Chapter 2. The rest of the dissertation contains two parts, followed by a chapter on our conclusions and future work.

Part I contains Chapter 3 to Chapter 5. We study representations of idioms derived from pretrained language models.

In Chapter 3, we explore to what extent a pretrained BERT model is able to encode the meaning of a potentially idiomatic expression (PIE) in a certain con-

text. We conduct two probing tasks, PIE usage classification and idiom paraphrase identification. Our experiment results suggest that BERT indeed is able to separate the literal and idiomatic usages of a PIE with high accuracy and is also able to encode the idiomatic meaning of a PIE to some extent. This work is published at RANLP2021 (International Conference Recent Advances in Natural Language Processing).

Chapter 4 is an extension of Chapter 3 but focuses on understanding MWEs' idiomaticity of multilingual pretrained language models in zero-shot and one-shot settings. This system description has been accepted to NAACL 2022 Workshop (SemEval 2022 Task 2: Multilingual Idiomaticity Detection and Sentence Embedding).

In Chapter 5, we address current issues in Chinese idiom embedding learning and evaluation. To learn embeddings for each Chengyu and conduct evaluations with new metrics, we first construct a large scale dataset of Chengyu synonyms and antonyms, then we propose two new evaluation metrics to calibrate the quality of the learned Chengyu embeddings. This work is also accepted at RANLP2021 (International Conference Recent Advances in Natural Language Processing).

Part II investigates more on neural network-based applications with Chinese idioms, ranging from Chapter 6 to Chapter 8. Chapter 6 and Chapter 7 focus on the task of Chinese Chengyu Recommendation given a context. Chapter 8 proposes a new task on Chengyu-oriented Chinese text polishing by constructing new datasets and baselines.

In Chapter 6, we present our Chengyu representation method of dual-embeddings. We address this task through constructing better representations for Chinese Chengyu. We treat them as fixed Multiword expressions and explore different embedding strategies. Specifically, we try to use a single embedding representation for each Chinese Chengyu instead of deriving its embedding from its component Chinese characters. We also explore contextualized embedding representations by using different interactions with the context. The work is published at COLING2020 (The 28th International Conference on Computational Linguistics).

In Chapter 7, we explore the possibility of pretraining Chengyu-oriented language models using large crawled corpus in a self-supervised fashion. We further explore how to improve pretrained Chinese BERT for Chinese Chengyu recommendation through pretraining on an open-ended Chengyu recommendation task using a large corpus. This helps the understanding of idioms and is helpful in the tough problem of differentiating near-synonyms in Chengyu. The work is published in the journal TALLIP (ACM Transactions on Asian and Low-Resource Language Information Processing).

Chapter 8 considers Chinese text polishing from the perspective of Chengyu usage. This work introduces the construction of the Chengyu-oriented text polishing dataset and the human annotation procedures to evaluate the quality of automatically constructed pairs. Based on the dataset, a series of baselines built over transformer-based generation models are provided. The work is in submission.

Finally in Chapter 9, I conclude the thesis by summarizing the contributions of this thesis and pointing out some potential future directions that are worth studying related to the understanding of Chinese idioms.

Chapter 2

Related Work

2.1 Probing Tasks

In natural language processing, a probing task is used to test whether the model has learned knowledge about a specific linguistic property. The notion of *probing* [36] or a *probing task* [23] refers to the use of a classification problem to reveal whether certain linguistic properties of sentences are captured in the input embedding representations of the sentences fed into the classification model. There have been studies investigating what properties of a sentence that its embedding might have contained [36, 112, 1]. The properties being probed include semantic roles [36], negation scopes [36], constituents [112], part-of-speech tags [112], sentence lengths [1], word orders [1], agreement information [42] and tense of the main clause [3]. With the emergence of contextualized embeddings such as BERT [31] and ELMO [96], researchers have also applied probing tasks to word-level contextual representations [127], attention mechanisms [21] and syntactic knowledge [97, 53]. Probing phrasal representations to study lexical composition has also attracted attention. Jawahar et al. [60] found that the compositional scheme underlying BERT mimics classical, tree-like structures. Shwartz and Dagan [113] conducted a series of experiments and concluded that lexical composition can shift the meanings of the constituent words and introduce implicit information. Yu and Ettinger [155]

reminded us that phrase representation in transformer models still relies heavily on word content, with little evidence of sophisticated composition of phrase meaning like that done by humans.

2.2 Potentially Idiomatic Expressions

Potentially Idiomatic Expressions (PIEs) originate from multiword expressions (MWEs) which have both an idiomatic interpretation and a literal interpretation, for example, *spill the beans*. Identifying the correct meaning of a PIE in a certain context is crucial for many downstream tasks including sentiment analysis [145], automatic spelling correction [57] and machine translation [59]. There has been both supervised [114] and unsupervised [46, 69] approaches to solve this problem. For example, Feldman and Peng [37] treated idiom recognition as outlier detection, which does not rely on costly annotated training data. Peng et al. [93] incorporated the affective hypothesis of idioms to facilitate the identification of idiomatic operations. Different from these studies, our objective is not to improve the performance of idiom recognition but rather to use the task as a probing task to understand the capabilities of BERT to encode idioms. With newly created large scale dataset *MAGPIE* [47], we can further investigate how contextualized word representations work for idiomatic expressions and literal ones.

2.3 Paraphrase Identification

Paraphrase identification aims to determine whether a pair of language units such as sentences have the same meaning [64] or whether a given paraphrase candidate can replace a given language unit in its context without changing overall semantic meaning of the text [153]. Idiom paraphrasing is a challenging task that has been attracting continuous attention from the community. For example, Liu and Hwa [77] investigated the effectiveness of a phrasal substitution method to replace idioms

with literal expressions, indicating that high quality paraphrasing of idiomatic expressions can be achieved. Yimam et al. [153] researched at a paraphrase-scoring annotation task and showed that the contexts have an impact on the ranking of paraphrases. Haagsma et al. [46] looks at the literal representation of the PIE's figurative sense (similar to dictionary definitions of an idiom's meaning, which can also be treated as paraphrase) to facilitate potentially idiomatic expression classification. Different from the studies above, in this thesis, we focus on understanding whether pretrained BERT models encode the semantic meanings of idioms, using idiom paraphrase identification as the probing task.

2.4 Word Embeddings

Word embedding is an important technique in NLP. It computes dense meaning representations for discrete words. It is built upon the distributional hypothesis that linguistic items with similar distributions have similar meanings. One important property for word embedding is that the vectors can show not only relatedness of words but also other linguistic regularities like the word analogy (*king* - *man* + *woman* = *queen*). Several methods have been proposed to learn non-contextualized word embeddings efficiently, including Continuous Bag-Of-Words (CBOW), Skip-Gram with Negative Sampling (SGNS) and GloVe [94]. CBOW and SGNS [85] are two most commonly used efficient log-linear prediction models forlearning non-contextualized word embeddings. CBOW tries to predict a word based on its context, where the context is represented as the average word embeddings within the contextual window. In contrast, SGNS tries to predict the contextual words of a given word, and negative sampling is used to reduce the computational cost. Word2vec [85] is a toolkit that uses a local context window to either predict nearby words from a given word (SGNS) or predict a target word from its set of context words (CBOW). GloVe [94] trains on the nonzero elements of a word-word co-occurrence matrix rather than on the entire sparse matrix or on individual context windows in a large

corpus.

Both CBOW and SGNS have been used to learn Chinese word embeddings [19, 75]. Since Chinese is an ideographic language with no explicit word delimiter between words [74], Chinese segmentation tools are usually used to identify word boundaries when learning Chinese word embeddings. Considering characters have their own semantic meanings, character information has been incorporated to improve Chinese word embeddings [19]. In addition, inspired by N-gram SGNS for English [164, 12], which predicts contextual N-grams rather than contextual words, Li et al. [75] trained Chinese word embeddings using N-gram SGNS and found that both N-gram and character features bring significant and consistent improvement. To further improve Chinese word embeddings, n-grams and characters are proven to be effective features in training word representations [164, 12]. The CBOW-based extensions for Chinese word embeddings learning models, e.g., CWE [19], GWE [115], SCWE [150], JWE [154], are trying to incorporate character-level features or even sub-character features. Character-enhanced Word Embedding (CWE) [19] addresses the importance of internal structures of words and proposes multiple-prototype character embeddings to deal with character ambiguity and non-compositional words. Similarity-based Character-enhanced Word Embedding (SCWE) [150] extends CWE by weighting the contribution of characters by measuring semantic similarity between a word and its component characters.

However, Chinese idioms are not always treated as words by Chinese segmentation tools. They are sometimes separated into multiple words. Therefore, only a subset of the idioms in our idiom vocabulary can be found as words in existing pre-trained non-contextualized Chinese word embeddings, and we are only able to perform evaluation on this subset of idioms. The approach of CWE and SCWE may be relevant to our Chengyu embedding learning. Internal structure of words are also important for Chengyu when considering the compositionality property.

Existing word embedding evaluation methods can be categorized into intrinsic and extrinsic methods [108]. Commonly used intrinsic methods include word simi-

larity and word analogy, while extrinsic methods rely on downstream NLP tasks [94]. In Chapter 5, we use an intrinsic method to evaluate Chinese idiom embeddings. Extrinsic evaluation methods adopt downstream NLP tasks using word embeddings as input features and measure their performances [94]. Intrinsic evaluation methods focus on the language regularities such as word similarity and word analogy learned by the embeddings [87, 10, 19, 147, 75].

Several benchmark datasets for evaluating Chinese word embeddings have been released [142, 38, 62, 19, 44, 58, 75]. A Chinese version of the most used dataset WordSim-353 [38, 62] (later adapted as WordSim-296 [19]) and a conventional dataset WordSim-240 [142] are widely adopted in many Chinese word embedding methods [19, 154]. Chinese Polysemous Word Similarity Dataset [44] was constructed to address the issue of polysemous words in earlier datasets. PKU-500 [147] considered more diverse criteria like domain, frequency, part-of-speech, word length, word sense and polarity. More recently, COS960 [58] is released focusing on the similarity of Multiword Expressions. Li et al. [75] released a big and balanced dataset CA8 for analogy evaluation, as well as over 100 Chinese word embeddings trained with different corpora and settings. Qiu et al. [99] researches the question whether intrinsic measures can predict the performance of downstream tasks and did the first study on the correlation between results of intrinsic evaluation and extrinsic evaluation with Chinese word embeddings.

In this thesis, we find that existing learned embeddings are still sub-optimal for Chinese idioms and the evaluation methods may rely on superficial cues like character-overlapping. We use an SGNS-based Chinese word embedding method as a representative non-contextualized word embedding method for evaluation. More importantly, contextualized word embeddings such as ELMO, GPT and BERT have been developed in recent years and shown their high effectiveness for many NLP tasks. We use two representative BERT variants, BERT-wwm and ERNIE, to evaluate Chinese idiom embeddings derived from pretrained Chinese BERT models. We propose a context-based learning method and two new evaluation metrics. We

compare the performance over idiom embeddings using all the models listed above to illustrate that the gains in performance are not consistent for the more challenging idioms case.

2.5 Transformer-based Pretrained Language Models in Chinese

Chinese is an ideographic language with no word delimiter between words in sentences [74]. Similar to WWM discussed in Section 1.3, Chinese-BERT-WWM [27] uses Chinese Word Segmentation (CWS) tools to identify word boundaries and mask a whole word explicitly. ERNIE [116] incorporates a multi-stage knowledge masking strategy which adds word-level mask, phrase-level mask and entity-level mask. It's worth noticing that ERNIE adopts the mixed corpus of Chinese Wikipedia, Baidu Baike, Baidu news and Baidu Tieba. Since most Chinese idioms have entries on Baidu Baike and they would be treated as entities by the entity-level mask, intuitively the embeddings extracted using ERNIE should be better than Chinese-BERT-WWM, whose CWS tools may not be able to recognize all the idioms. ERNIE2 [117] uses a continual pre-training framework to build and learn incrementally pre-training tasks through constant multi-task learning.

2.6 Chinese Chengyu Recommendation

Cloze-style reading comprehension is an important form in assessing machine reading abilities. Researchers created many large-scale cloze-style reading comprehension datasets like CNN/Daily Mail [51], Children's Book Test (CBT) [54] and RACE [70]. These datasets have inspired the design of various neural-based models [51, 17] and some become benchmarks for machine reading comprehension.

To facilitate the study of Chengyu comprehension using deep learning models,

Zheng et al. [165] released a large-scale Chinese Idiom Dataset called **ChID**[1]. The dataset was also created in the "cloze" style. The authors collected passages from novels and essays on the Internet and news articles from THUCTC[2]. The authors then masked Chinese idioms found in these passages using a blank. 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.

In this thesis, we use two different versions of the ChID datasets.

- **ChID-Official:** This version is the official release of ChID in their paper[165]. 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. 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). Some statistics of the data can be found in Table 2.1.

	In-domain				Out-of-domain	Total
	Train	Dev	Test	Total	Out	Total
Passages	520,711	20,000	20,000	560,711	20,096	580,807
Distinct idioms	3,848	3,458	3,502	3,848	3,626	3,848
Total blanks	648,920	24,822	24,948	698,690	30,023	728,713

Table 2.1: Some statistics of the **ChID-Official** dataset. The row of passages shows how many distinct passages are used for each split. The second row shows how many distinct idioms are covered on each split. The final row shows how many blanks are there on each split.

Table 2.2 shows an example from the training set of ChID-Official. We can

[1] https://github.com/zhengcj1/ChID-Dataset
[2] https://github.com/thunlp/THUCTC

see that among the seven candidates, grammatically, most can fit into the local context “完全不会有_____的感觉” (“you will not feel _____ at all”) well, but to select the best answer we need to read and understand the entire passage. The ChID dataset has a big size, containing more than 500K passages and more than 600K blanks, making it possible for researchers to train complex neural network models on this dataset.

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 2.2: 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.

- **ChID-Competition:** ChID-Competition[3] is the data for an online competition[4] 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 2.3 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

[3] https://github.com/zhengcj1/ChID-Dataset/tree/master/Competition

[4] 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 2.3: An example in ChID-Competition. We show only three passages out of the five passages in this entry.

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).

Despite the importance of Chengyu in Chinese language understanding, there have been only a few pieces of work on Chengyu using neural models [61, 80, 165]. Chinese Chengyu Recommendation (CCR) has been addressed in recent years [80, 61, 165].

2.7 Text Polishing

Text polishing is closely related to intelligent writing assistance [49], paraphrase generation [83, 160] and text style transfer [98].

Intelligent Writing Assistance Intelligent Writing Assistance [49] is provided by computer systems to writers for analyzing text for errors in grammar or style, and more generally, for outline construction, plot construction, or even for automatically generating text. There are also works of intelligent writing assistance in Chinese. WINGS [30] is a Chinese input method extended on IBus-Pinyin providing writing suggestions for writers. Given the prevalence usage of Chengyu in the Chinese language, neural-based Chinese Chengyu recommendation systems are proposed for enhancing elegancy in essay writing [78, 80].

Text Style Transfer Text style transfer is the task of rephrasing the text to contain specific stylistic properties without changing the intent or effect within the context. Meaning preservation has been one of the common objectives in style transfer as literal meaning is likely to change when the transfer occurs [98]. For example, it would be hard to modify sentiment while preserving meaning or intent. In the text polishing, the models have stronger requirements to keep the original intent of the author unchanged.

Back-translation Back-translation, as a data augmentation method, has been adopted in a wide range of tasks, like text style transfer [98] and paraphrase generation [39, 84, 144]. Back-translation can not only rephrase the source sentence with reduced effect of the original style [98], but also provide multiple candidates for selection. In this work, we use back-translation to automatically generate samples for constructing the text polishing dataset.

Part I

Idiom Representations Derived from Pretrained Language Models

Chapter 3

Does BERT Understand Idioms? A Probing-Based Empirical Study of BERT Encodings of Idioms

In this chapter, we study to what extent a pretrained BERT model is able to encode the meaning of a potentially idiomatic expression (PIE) in a certain context. We refer readers to Section 2.2 for details about PIE. We make use of a few existing datasets and perform two probing tasks: PIE usage classification and idiom paraphrase identification. Our experiment results suggest that BERT indeed is able to separate the literal and idiomatic usages of a PIE with high accuracy. It is also able to encode the idiomatic meaning of a PIE to some extent.

3.1 Introduction

Understanding idiomatic expressions is important for NLP tasks such as sentiment analysis [6, 145] and machine translation [59, 111]. However, due to the non-compositionality of idioms, it remains a challenge to model the semantic meanings of idioms effectively [106, 113].

BERT is a contextualized pretrained language model that has been widely used

and proven to be highly effective for many NLP tasks [31]. To better understand how BERT works, recently the community has adopted the approach of *probing*, where a *probing task* is designed to test whether BERT encodings contain sufficient information to perform the task well. Examples of probing tasks include POS tagging and parsing [52, 148] as well as semantic reasoning tasks such as understanding numbers [132].

It is therefore also natural to ask whether BERT encodes any knowledge about the usage and meanings of idioms, given that BERT was trained on huge corpora, which must contain many idiomatic expressions. However, this problem has not been well explored. To the best of our knowledge, the closest existing work is by Shwartz and Dagan [113], who studied whether pretrained (static and contextualized) word embeddings can detect meaning shift and implicit information of phrases, with the help of several probing tasks. However, we believe there is a need for further exploration. We note that Shwartz and Dagan [113] did not specifically focus on idioms; only one of the six probing tasks was directly related to idioms, and only idiomatic noun compounds were studied. Since English idioms have different syntactic structures, it would be useful to experiment with a higher coverage of different types of idioms.

In this chapter, we focus on probing BERT to understand whether BERT embeddings can encode the meanings of a diverse range of different types of idioms. We propose two probing tasks to test whether BERT understands idioms. First, given a context containing a potentially idiomatic expression (PIE), the task is to decide whether the meaning of the PIE is literal or idiomatic, based on the BERT-encoded contextualized embedding of the PIE. We hypothesize that if pretrained BERT could perform the task well, it would indicate that BERT knows the difference between literal and idiomatic usages of the same expression based on its context. For this task, we use a large dataset recently released by Haagsma et al. [47], which covers 1756 unique idioms and 50K contextual sentences, much larger and more diverse than the idiomatic noun compounds dataset used by Shwartz and Dagan [113]. However, this

task is not sufficient to show whether BERT truly understands the *idiomatic meaning* of a PIE. In order to test this, we design a second probing task based on existing idiom paraphrase datasets. The task is to select the correct paraphrase of an idiom among a set of candidate phrases based on the cosine similarity between the idiom's BERT embedding and these candidate phrases' BERT embeddings. We hypothesize that if the correct paraphrase could be ranked higher than other irrelevant phrases, it would indicate that BERT indeed understands the idiomatic meaning of the idiom.

It is important to note that our objective is not to improve the performance of the two tasks by designing effective learning methods; rather, the objective is to use these two tasks to probe pretrained BERT in order to understand how much BERT encodes the meanings of idioms. Therefore, the models for the two probing tasks are simple models without many parameters to be learned.

Through our empirical study using both the original BERT and ERNIE2 [117] (an improved version of BERT, see Section1.3 for more information), we find that compared with non-contextualized embedding representations of PIEs, contextualized BERT and ERNIE2 embeddings of PIEs can clearly achieve higher accuracy for PIE usage classification, with an accuracy level around 90%, suggesting that BERT can use the context to accurately guess whether an expression is used literally or idiomatically. For paraphrase identification, we find that BERT and ERNIE2 perform significantly better than a random baseline, although the absolute performance is still considered low. Since paraphrase identification is itself challenging, to put things in perspective, we also compare with paraphrase identification for general multi-word expressions (MWEs). Contrary to our expectation, we find that identifying paraphrases for general MWEs does not necessarily fare better than for idioms. Further analysis reveals that this is because BERT contextualization actually hurts paraphrase identification for general MWEs but not so for idioms.

3.2 Probing Tasks

We design two probing tasks to answer two research questions: (1) Can BERT distinguish the idiomatic usage of a PIE from its literal usage? (2) Can BERT understand the idiomatic meaning of an idiom? Both questions are related to the capabilities of BERT to understand idioms, but the second task is more demanding than the first. The two tasks also share similar objectives as the probing tasks designed by [113], which aimed to test whether pretrained word embeddings can detect the shift of meaning of a phrase from its component words, and whether pretrained word embeddings understand the implicit meaning of a phrase. However, they are conducting probing at word level, which focuses on whether the meaning of a word in a noun compound (NC) is literal. The dataset [105] used by them only has 90 noun compounds. Although they try to augment the dataset using data released by Tratz [129], the dataset is still limited to 3K. The paraphrase identification task used by them also uses compounds and addresses whether the paraphrase describes the semantic relation between two words of a noun compound [50].

In this chapter, we use a much larger dataset called MAGPIE [47] that covers much more potentially idiomatic expressions for phrase-level literal-idiomatic classification. To make the task more challenging, we choose to split the data such that the idiomatic expressions in the training, development, and test sets do not overlap. We further adapt several paraphrase datasets [77, 153, 95] to compare phrasal semantic relatedness for idioms. We compare the effect of BERT encodings at different layers for the two probing tasks to better understand the effect of contextualization.

3.2.1 PIE Usage Classification

Many MWEs can be interpreted either literally or idiomatically. In some literature, these expressions are defined as potentially idiomatic expressions (PIEs) [114, 46, 47]. For example, "spill the beans" can either be used literally to refer to the action of spilling beans or in its idiomatic sense to refer to disclosing some secrete. However,

current approaches are investigating this problem with the limitation to one or more syntactic patterns. In this chapter, we propose to use the latest large scale dataset MAGPIE to probe how BERT is capturing the difference of literal and non-literal usage of a PIE.

Task Definition. Given a piece of context denoted as $(w_1, w_2, \ldots, w_n)$ containing a PIE with m words, $w_i, \ldots, w_{i+m-1}$, the task is to decide whether the PIE is used with its *literal* meaning or its *idiomatic* meaning. Performance is measured by accuracy. It is important to note that since our goal is to test whether pretrained BERT can already encode such knowledge, we do not train a classifier *per idiom*. Instead, we train a single binary classifier using a set of training PIEs and their labeled contexts, and test the classifier on a separate set of different test PIEs and their contexts.

Context: Think of a sunflower turning its flower head towards a source of light — and therefore of energy. The sunflower does not learn by experience to **turn its head** more effectively as it matures, or not to turn at all if it is repeatedly electrically shocked every time it does so.	
Annotation:	
Label: literal	PIE: turn head
Confidence: 0.75	Genre: W nonAc: nat science
Judgment Count: 4	Variant Type: combined-inflection
Label Distribution: {'idiomatic': 0.25, 'literal': 0.75}[1]	

Table 3.1: An example from MAGPIE dataset with details of annotations.

Data. We use the *MAGPIE* dataset [47], which is the largest-to-date corpus of English PIEs and labeled instances of both their literal and idiomatic usages in different contexts. The corpus comprises 1756 unique PIEs and more than 50K contexts, an order of a magnitude larger than previous similar resources. Annotations of *MAGPIE* included various aspects: annotation (dis)agreement, distribution of idiom types, sense distributions across types, composition of the 'other'-category, and influence of genre. An example of *MAGPIE* is given in Table 3.1. In this chapter,

[1]For the other labels that are not used in this chapter, we refer the reader to the original paper for details.

	Size	Example	
		Sentence	Paraphrase
Idioms-MWEs	171	If only I could **soup up** this computer to run just a little faster.	increase the power of
MWEs-MWEs	176	She constantly complains of boredom as her presence **at home** is merely decorative, while her husband is heavily involved in his scholarly interests.	in her house
Idioms-Idioms	158	This Cuban Black Bean recipe is pretty much as **easy as beans** get and they are SO delicious.	piece of cake

Table 3.2: Paraphrase evaluation datasets. We select one example from each dataset. The source phrase is highlighted in bold font in the sentence.

we further analyse what might be the reason of BERT's advantage in connection with annotation agreement.

3.2.2 Idiom Paraphrase Identification

In this chapter, to further understand whether BERT has learned the idiomatic meaning of phrases, we propose the Idiom Paraphrase Identification probing task to check whether contextualized representations of PIEs encoded by BERT have shifted meanings that are closer to their paraphrases.

Task Definition. Given a piece of context denoted as $(w_1, w_2, \ldots, w_n)$ containing a PIE $w_i, \ldots, w_{i+m-1}$ where the PIE is known to be used idiomatically, and given a set of candidate phrases $\mathcal{P} = \{p_1, p_2, \ldots, p_L\}$, where each $p_l \in \mathcal{P}$ is a MWE and one of them is a paraphrase of the given idiom, the task is to identify the correct paraphrase from $\mathcal{P}$. We cast this task as a ranking problem and use Mean Reciprocal Rank (MRR) to measure the performance.

Data. We combine different resources described below to create the data needed to perform this paraphrase identification task. Specifically, we create three datasets: (1) **Idioms-MWEs**, (2) **MWEs-MWEs**, and (3) **Idioms-Idioms**, Details of the collection of these three datasets are listed below:

- **Idioms-MWEs:** We use the idiom paraphrase dataset created by Liu and Hwa [77]. Each instance in this dataset is a context sentence containing an idiom together with a phrase that can substitute the idiom in the context. The dataset was created by shortening the definitions of these idioms from a dictionary and performing appropriate grammatical and referential transformations to ensure that the idiom substitution fits seamlessly into the original context. The paraphrases have also been verified and refined by human annotators. This gives us a dataset with high quality paraphrases of idiomatic expressions. The dataset contains 171 unique idioms, each with a single context sentence and a paraphrase.

- **MWEs-MWEs:** Since paraphrase identification itself is likely a challenging task even for non-idiomatic MWEs, in order to put things in perspective, we also make use of another paraphrase dataset that contains pairs of MWEs that are paraphrases. Yimam et al. [153] investigated the impact of context for the paraphrase ranking task using both multi-word expressions and single words. The dataset covers 17k data points (2k MWEs and 15k single word) annotated through crowd-sourcing. The 2k MWEs are of particular interest to us in this probing task. We processed the original dataset by retaining only those paraphrase pairs with a human agreement score of 4, which gives us a final set of 176 entries of a MWE in a context as well as their paraphrases. We find that these 176 entries do not overlap with the PIEs in the MAGPIE dataset, suggesting that these MWEs are likely all non-idiomatic expressions. By performing paraphrase identification on this dataset, we can get a sense of the expected performance for paraphrase identification on phrases that are not

idiomatic.

- **Idioms-Idioms:** Pershina et al. [95] presented idiomatic expressions as a new domain for short-text paraphrase identification and released a dataset of 1.4K annotated idiom paraphrase pairs and 2.4K idioms with definitions. However, no context is provided for each idiom. We use this dataset jointly with MAGPIE to construct an evaluation dataset where each entry has an idiom usage label and a definition of the PIE if it is used idiomatically. We use the 91 Idiom-Idiom paraphrase pairs to construct a more challenging split to check if BERT can perceive these paraphrases. By switching the order of each idiom pair, we obtain 192 candidate entries. We retrieve contexts with *idiomatic* label for each idiom pair from MAGPIE to construct the evaluation dataset. For those entries that do not exist in MAGPIE, we retrieve online examples like Wiktionary manually. We filter out some of the entries which share duplicate contexts or have the source idiom being only a naive variation of the target. At the end of the process, we get 158 entries.

For each dataset, we list its size and one example in Table 3.2.

To create the set of candidate paraphrases, we simply pool the paraphrases of all the entries of the three datasets together as the set of candidate paraphrases for all instances.

3.3 Experiments

For each of the two probing tasks above, we use pretrained BERT[2] and ERNIE2[3] to process each context $(w_1, w_2, \ldots, w_n)$. Following standard practice, we prepend the `[CLS]` token to the beginning of the sequence and append the `[SEP]` token to the end. The sequence is then fed into an L-layer BERT. Let $\mathbf{h}_i^k \in \mathbb{R}^d$ denote the hidden vector produced by the kth layer of BERT representing w_i. When $k = 0$, $\mathbf{h}_i^0$ denotes

[2] `huggingface.co/bert-base-uncased`
[3] `huggingface.co/nghuyong/ernie-2.0-en`

the combined representation of the word embedding, the position embedding and the token type embedding before it is fed into the transformer-based encoder.

For each PIE, we get a sequence of hidden vectors at the kth layer for the m tokens inside this PIE as follows: $\mathbf{p}^k = (\mathbf{h}_i^k, \mathbf{h}_{i+1}^k, \ldots, \mathbf{h}_{i+m-1}^k)$. We will use these contextualized BERT embeddings of the PIE as input to the model for the probing tasks. Note that when training the model for a probing task, BERT is not fine-tuned.

For both probing tasks, we experiment with both the original BERT [31] and ERNIE2 [117], which supports phrase masking by using lexical analysis and chunking tools to get the boundary of phrases in the sentences. Our code and data are released on github [4].

3.3.1 PIE Classification

After we get the hidden representation $\mathbf{p}^k = (\mathbf{h}_i^k, \mathbf{h}_{i+1}^k, \ldots, \mathbf{h}_{i+m-1}^k)$ of the PIE, we further encode the sequence into a single vector using a bidirectional LSTM encoder. We then treat this vector as input to train the binary PIE usage classifier using a linear classifier.

We show the accuracy of the trained PIE usage classifier on both the development set and the test set in Table 3.3. We include a baseline BL-majority that always predicts the usage to be idiomatic. This is because we observe that there are more instances in this dataset labeled as idiomatic than literal. We also include another baseline BL-GloVe, which uses the static GloVe word embeddings [94] to replace the BERT encoded representations. For BERT embeddings, we include the results using the bottom layer (Layer-0) and the results using the final layer (Layer-12). Including Layer-0 is for us to observe how the static embeddings of BERT have performed.

From the table, we can draw the following conclusions:

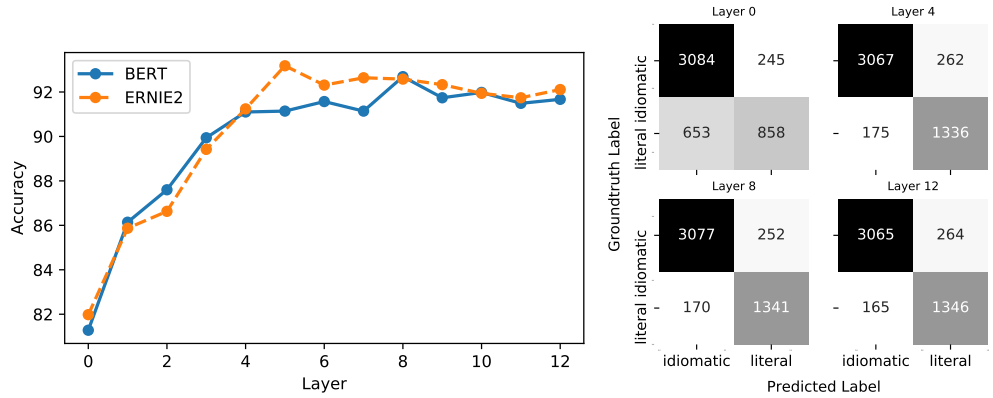
1. The baseline method **BL-majority** achieves an accuracy above 50%. This shows that the dataset is not balanced, with more instances of idiomatic usage.

[4] `https://github.com/VisualJoyce/CiYi`

2. Using Layer-0 of BERT and ERNIE2, i.e., using only static word embeddings, we can see that the performance is always above 80% and is very close to **BL-GloVe**. This suggests that even the static word embeddings contain some prior knowledge about whether the expression is literal or idiomatic.

3. Using Layer-12 of BERT and ERNIE2, we can see that the accuracy of PIE usage classification significantly increased compared with using Layer-0. In fact, the absolute accuracy level is quite high, reaching 90%. This confirms that with BERT contextualization, the embeddings of the PIE better reflect the usage of the PIE, allowing the classifier to easily predict whether the PIE is used literally or idiomatically.

This shows that BERT can indeed encode the knowledge about the usage of a PIE.



(a) PIE usage classification accuracy on test data with different Transformer layers.

(b) Confusion matrix for Layer-0, Layer-4, Layer-8 and Layer-12.

Figure 3.1: PIE usage classification.

Given the large gap between the classification accuracy using Layer-0 and Layer-12, next we experiment with other intermediate layers of the Transformer architecture for BERT and ERNIE2. The results are shown in Figure 3.1a. From the figure we find that starting from around Layer-4 the performance stabilizes and the last layer is not necessarily the one with the best performance. This shows that BERT requires just a few rounds of contextualization to encode the idiom usage information.

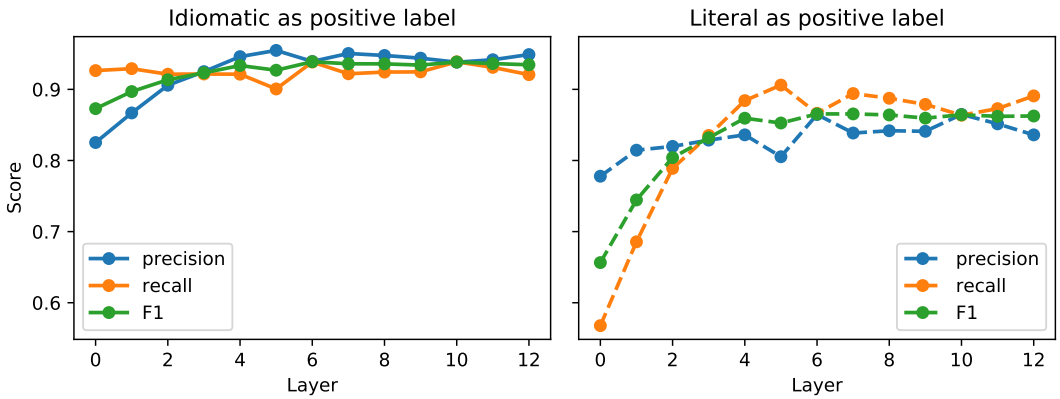


Figure 3.2: F1 score, precision and recall curve for different layers in BERT. We list both cases that either choosing *idiomatic* or *literal* as the positive label.

		Dev	Test
BL-majority		71.76	68.78
BL-GloVe		80.52	82.05
BERT	Layer-0	83.90	81.28
BERT	Layer-12	**90.33**	91.67
ERNIE2	Layer-0	84.65	81.98
ERNIE2	Layer-12	89.03	**92.11**

Table 3.3: PIE classification accuracy.

To better understand how BERT contextualization improves PIE usage classification, we further zoom into the two different types of errors: (1) literal usage mistakenly classified as idiomatic usage, and (2) idiomatic usage mistakenly classified as literal usage. We show the numbers of these error cases in four confusion matrices in Figure 3.1b (one confusion matrix for one of Layer-0, Layer-4, Layer-8 and Layer-12), where the lower-left corner shows the first type of errors and the upper-right corner shows the second type of errors. In Figure 3.2, we further show the precision, recall and F1 scores across all the layers by either choosing *idiomatic* or *literal* as the positive label.

We observe that interestingly the error reductions from Layer-0 to Layer-12 comes mostly from the group *literal-idiomatic* where literal expressions are wrongly predicted to be idiomatic. We hypothesize that this is because without contextualization, some of the words in these PIEs tend to indicate that the PIEs are used

idiomatically, probably because these words have appeared often in other idiomatic expressions in the training data; but after considering the specific contexts these PIEs are placed in, i.e., with BERT contextualization, the model recognizes that these contexts are semantically similar to the literal meanings of the tokens inside these PIEs, and therefore predict the usage as being literal. This shows that with more contextualization, BERT embeddings help the most in recognizing literal usages of PIEs.

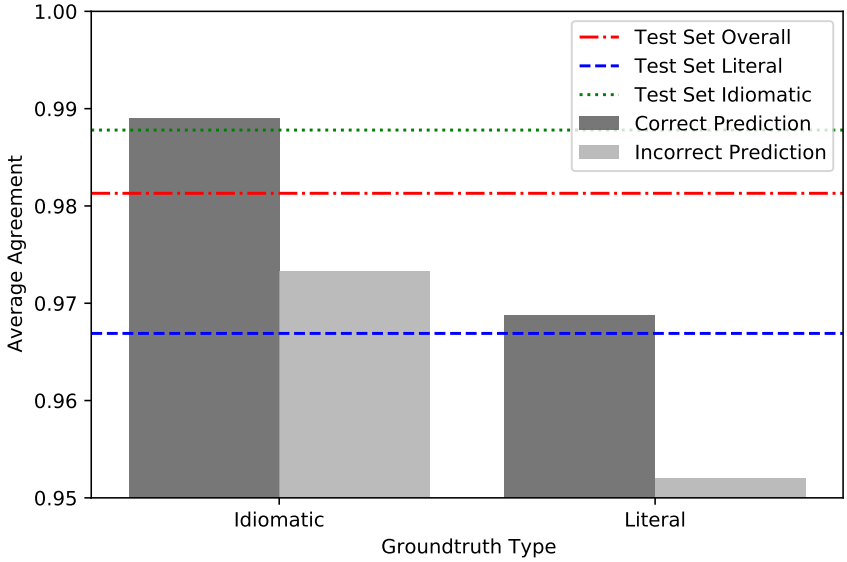


Figure 3.3: Average agreement score for predictions in Layer-12. Horizontal lines are average annotation agreement scores over test set: (1) Idiomatic cases, (2) Literal cases, (3) Overall.

We further ask the question whether those instances where BERT embeddings did not do well for the PIE usage classification task are those instances where human annotators' agreement is also low. To answer this question, we show the average annotation agreement scores on the test set for correctly predicted instances and incorrectly predicted instances. The statistics are shown in Figure 3.3. The red line shows the average agreement score over *all* test instances, the green line shows the average agreement score over those instances whose ground truth labels are "idiomatic", and the blue line shows the average agreement score over those instances with the ground truth label "literal". We can see that human annotations have a clearly higher degree of agreement on those idiomatic usages of PIEs, but a lower

agreement when a PIE is likely used literally. The four bars in Figure 3.3 shows the average agreement scores of correctly and incorrectly predicted instances, grouped by the ground truth labels. We can see that clearly those incorrectly predicted instances (shown in light gray bars) have clearly lower human agreement scores compared with the correctly predicted ones. This verifies our hypothesis that the model tends to make mistakes on those instances which humans also find hard.

3.3.2 Paraphrase Identification

For the paraphrase identification task, after we get the hidden representation $\mathbf{p}^k$ of the PIE in its context, we take the average of these vectors to obtain a single vector. For each candidate paraphrase, we perform the same encoding, without any context, and then take the average of the produced hidden vectors. Finally, we rank the candidates based on the cosine similarity between the PIE's embedding and the candidate's embedding.

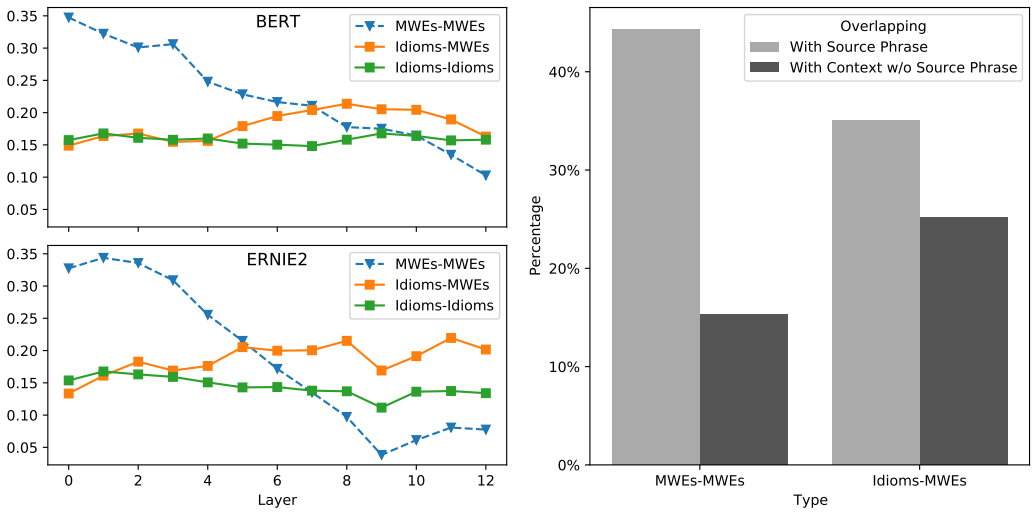
	Idioms-MWEs	MWEs-MWEs	Idioms-Idioms
BL-random	0.013	0.013	0.013
BERT	0.163	0.104	0.154
ERNIE2	0.202	0.078	0.136

Table 3.4: MRR scores for paraphrase ranking.

The Mean Reciprocal Rank (MRR) scores are listed in Table 3.4. For comparison, we consider a baseline that randomly ranks the candidates. We can observe the following from the table:

1. BERT and ERNIE2 can perform better than the random baseline on Idioms-MWEs, although the absolute values of MRR are low. This shows that BERT contextualized embeddings can still encode the idiomatic meanings of idioms to some extent.

2. We also observe that identifying paraphrases for MWEs-MWEs, which are likely not idiomatic, is not easier than for idioms. This is counter-intuitive and



(a) Paraphrase ranking MRR across layers for different splits.

(b) Percentage of pairs with word overlap.

Figure 3.4: Paraphrase identification.

we will show further investigation below.

3. Identifying paraphrase idioms of idioms (Idioms-Idioms) is a bit harder than identifying MWEs-MWEs paraphrases. This maybe because the candidate idioms are not contextualized, and therefore their embeddings do not reflect their idiomatic meanings.

To better understand why paraphrase identification for general MWEs has even lower performance than for idioms, we again test the performance using different layers of BERT/ERNIE2 embeddings. The results are shown in Figure 3.4a. Now it is clear that with non-contextualized embeddings (i.e., Layer-0), paraphrase identification for general MWEs is actually much easier than for idioms. This is intuitive because the meaning of non-idiomatic MWEs can be derived from their component words and therefore contextualization is not needed. The figure also shows that with more contextualization, performance of paraphrase identification for general MWEs is largely hurt, but this is not the case for idioms. It's also interesting that, for Idioms-Idioms, the MRR scores do not change much with layers. We think this may due to both an idiom and its idiomatic paraphrase share less overlap with the context.

Noticing that the performance of paraphrase identification for **Idioms-MWEs**

surpasses **MWEs-MWEs** at Layer-8, i.e., when there is some degree of contextualization, we conduct some further analysis to understand why. Specifically, given a query idiom (or query MWE) q, its context c, and its ground truth paraphrase MWE p, we would like to check if p tends to have common words with q and c, respectively. Our hypothesis is that if p shares common words with c, then contextualized word embeddings are helpful because they encode the context c.

We show our analysis in Figure 3.4b. In the left hand side of the figure, the light gray bar shows the percentage of test instances in the MWEs-MWEs dataset where the query MWE q shares at lease one common word with the ground truth paraphrase p, and the dark gray bar shows the percentage of test instances in MWEs-MWEs where the context c shares at least one common word with the ground truth paraphrase p. The right hand side of the figure shows the same percentages for the Idioms-MWEs dataset. We can see that for MWE-MWE paraphrase pairs, it is less common for the ground truth paraphrase to share a common word with the context of the query phrase, compared with Idiom-MWE paraphrase pairs. This is reasonable because for an idiom, its idiomatic meaning is often not directly linked to the semantic meanings of their component words, and therefore words in the idiom itself may not overlap with words in its paraphrase; on the other hand, the context where an idiom appears may imply the idiom's idiomatic meaning, and therefore may have word overlap with the paraphrase. The statistics shown in Figure 3.4b shows that because for MWEs, their paraphrases are less likely to share common words with the contexts where the MWEs appear, contextualization done by BERT therefore not only is not so useful but also may harm the performance of paraphrase identification.

3.4 Conclusion

In this chapter, we use two probing tasks to study whether BERT understands English idioms. In conclusion, we find that BERT is able to detect idiomatic usages of a PIE

with a high accuracy, and with more contextualization as the layer increases, BERT helps the most in recognizing literal usages of PIEs. However, this only proves that BERT is effective in detecting meaning shift for idiomatic expressions. To further probe if the shifted meanings are closer to their paraphrases, we adopt the paraphrase identification task by gathering three different types of paraphrase pairs, MWEs-MWEs, MWEs-Idioms and Idioms-Idioms. Our experiments show that BERT is able to encode the idiomatic meaning to some extent. However, contextualization may have different effects for MWEs and idioms, which still requires further exploration to fully explain.

Chapter 4

HiJoNLP at SemEval-2022 Task 2: Detecting Idiomaticity of Multiword Expressions using Multilingual Pretrained Language Models

Last chapter investigated idiomaticity of monolingual pretrained language models. This chapter describes an approach to detect idiomaticity only from the contextualized representation of a MWE over mutltilingual pretrained language models. Our experiments find that larger models are usually more effective in idiomaticity detection. However, using a higher layer of the model may not guarantee a better performance. In the multilingual scenario, the convergence of different languages are not consistent and rich-resource languages have big advantages over other languages.

4.1 Introduction

Due to the limited understanding of how pretrained language models may handle representation of phrases, a series of works are proposed to investigate phrase composition from their contextualized representations. Yu and Ettinger [155] conduct

analysis of phrasal representations in state-of-the-art pre-trained transformers and find that phrase representation in these models still relies heavily on word content, showing little evidence of nuanced composition. Shwartz and Dagan [113] confirm that contextualized word representations perform better than static word embeddings, more so on detecting meaning shift than in recovering implicit information. Therefore, it remains a challenging problem to resolve the idiomaticity of phrases.

Specifically on idiomaticity, recent approaches are trying to further diagnose pretrained language models using new metrics and datasets. Garcia et al. [40] analyse different levels of contextualisation to check to what extent models are able to detect idiomaticity at type and token level. Garcia et al. [41] propose probing measures to assess Noun Compound (NC) idiomaticity and conclude that idiomaticity is not yet accurately represented by contextualised models. AStitchInLanguageModels [125] design two tasks to first test a language model's ability to detect idiom usage, and the effectiveness of a language model in generating representations of sentences containing idioms.

In last chapter, we conduct two probing tasks, PIE usage classification and idiom paraphrase identification, suggesting that BERT indeed is able to separate the literal and idiomatic usages of a PIE with high accuracy and is also able to encode the idiomatic meaning of a PIE to some extent. However, there's still much more to explore in idiomaticity.

Based upon AStitchInLanguageModels [125], SemEval-2022 Task2 [126] is proposed with a focus on multilingual idiomaticity. The task is arranged consisting the two subtasks:

1. Subtask A: A binary classification task aimed at determining whether a sentence contains an idiomatic expression.

2. Subtask B: Pretrain or finetune a model which is expected to output the correct Semantic Text Similarity (STS) scores between sentence pairs, whether or not either sentence contains an idiomatic expression.

In this chapter, we focus on Subtask A and investigate how the span representation of a MWE can tell about its idiomaticity. We extend one of the monolingual idiomaticity probing method [122] to multilingual scenario and compare multiple settings using multi-lingual BERT (mBERT) [31] and XLM-R [24]. Following Yu and Ettinger [155], we also consider variations of phrase representations across models, layers, and representation types. Different from them, we use more representation types to conduct the experiments.

Our main conclusion from these experiments are two folds:

1. Larger models are usually more effective in idiomaticity detection. However, a higher layer may not contribute more to the idiomaticity detection task, or more contextualization does not guarantee a better performance.

2. For multilingual scenario, the convergence of different languages are not consistent. Rich resource languages have initiative advantages over other languages.

4.2 System Overview

4.2.1 Subtask A

For Subtask A, to test models' ability to generalise, both zero-shot and one-shot settings are considered.

1. zero-shot: PIEs in the training set are completely disjoint from those in the test and development sets.

2. one-shot: one positive and one negative training examples for each MWE in the test and development sets

Note that the actual examples in the training data are different from those in the test and development sets in both settings.

Data Each row of the data of Subtask A has attributes like language and the potentially idiomatic MWE. The ”Target” is the sentence that contains this MWE. The previous and next sentences for context are also provided. The label provides the annotation of that row, and a label of 0 indicates ”Idiomatic” and a label of 1 indicates ”non-idiomatic”, including proper nouns.

Baseline The baseline model [126] is based on mBERT. In the zero-shot setting, the model uses the context (the sentences preceding and succeeding the one containing the idioms) and does not add the idiom as an additional feature (in the “second input sentence”). In the one shot setting, the model is trained on both the zero-shot and one-shot data, but exclude the context (the sentences preceding and succeeding the one containing the idioms) and add the idiom as an additional feature in the “second sentence”.

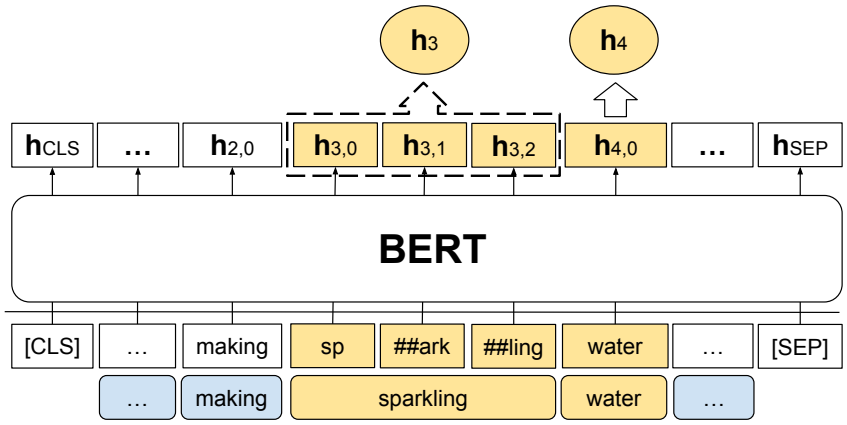


Figure 4.1: Mismatched transformer-based span representation.

4.2.2 Span-based Model

While the common practice for classification tasks using pretrained language models usually needs concatenation of text sequences, this does not tell us enough information how representations of MWEs may lead to the change of performance. Therefore, in this work, we focus on the contextualized representations of MWEs to predict its idiomaticity.

Problem Formulation Consisting with the definition in [122], given a sentence denoted as $(w_1, w_2, \ldots, w_n)$, which contains a MWE with m words denoted as $(w_i, \ldots, w_{i+m-1})$, The task is to decide whether the MWE is used with its *literal* meaning or its *idiomatic* meaning, or if a sentence contains an idiomatic expression as describe in the task.

Span Identification In this work, our method requires a pair of span indices of the target MWE to extract their hidden representation from the encoded sequence. However, in this task, no such indices is offered explicitly from the dataset. We empirically find these indices by using editing distances in characters between the MWE and the sentence. This method works for most of the cases.

Span Representation For each MWE, we have a pair of span offsets in the original context. We use an L-layer BERT to process the tokenized context by prepending `[CLS]` to the beginning and appending `[SEP]` to the end. Let $\mathbf{h}_i^k \in \mathbb{R}^d$ denote the hidden vector produced by the kth layer of BERT representing w_i. We extract the hidden representations of the span to get its contextualized representations. For each MWE, we get a sequence of hidden vectors at the k-th layer for the m tokens inside this MWE as follows: $\mathbf{p}^k = (\mathbf{h}_i^k, \mathbf{h}_{i+1}^k, \ldots, \mathbf{h}_{i+m-1}^k)$.

In transformer-based models, a word might be tokenized into several pieces. We adopt the mismatched tokenization trick offered by Allennlp [1] to reconstruct its hidden vector. The hidden vector will be the average embeddings of constituent pieces. The mismatched encoding is illustrated in Figure 4.1.

We represent the target MWE using the span by six different kinds of combinations of the span's words. The first four of them are only using their endpoints. We use $\mathbf{x} = \mathbf{h}_i^k$ to denote the start of the span and $\mathbf{y} = \mathbf{h}_{i+m-1}^k$ to denote the end of the span.

1. **x,y** The span is represented by a direct concatenation of two endpoints.

[1] `https://github.com/allenai/allennlp`

2. **x,y,x-y** The span is represented by a direct concatenation of two endpoints and the difference of them.

3. **x,y,x*y** The span is represented by a direct concatenation of two endpoints and the elementwise product of them.

4. **x,y,x*y,x-y** The span is represented by a direct concatenation of two endpoints, the elementwise product and the difference of them.

5. **SelfAttentive** We firstly compute an unnormalized attention score for each word in the document. Then we compute spans representations with respect to these scores by normalising the attention scores for words inside the span.

6. **MaxPooling** A span is represented through a dimension-wise max-pooling operation. Given a span, the resulting value of a dimension is using the maximum value of this dimension across all the span tokens.

Span Classification We use a binary linear classifier upon the span representation.

4.3 Experiments

In this paper, we want to test how the pretrained model, the transformer layer and the representation type, affect performance of idiomaticity detection.

Model	Type	Layer	EN	PT	GL	Avg
mBERT [126]	-	12	70.70	68.03	50.65	65.40
mBERT	x,y,x-y	12	76.24	72.27	64.27	72.85
XLM-R	x,y	8	**77.62**	71.61	64.88	72.68
XLM-R-L	x,y,x-y	24	75.22	**75.80**	**69.01**	**74.66**

Table 4.1: Experiment results of zero-shot setting for different multilingual pretrained models, in macro F1 score.

Model	Type	Layer	EN	PT	GL	Avg
mBERT [126]	-	12	88.62	86.37	81.62	86.46
mBERT	MaxPooling	8	86.59	85.82	85.77	86.63
XLM-R	MaxPooling	8	89.49	83.71	82.19	86.17
XLM-R-L	x,y,x*y,x-y	24	**91.26**	**86.96**	**89.06**	**89.79**

Table 4.2: Experiment results of one-shot setting for different multilingual pretrained models, in macro F1 score.

4.3.1 Settings

This subtask is evaluated using the Macro F1 score between the gold labels and model predictions (see the details in the evaluation script).

All the multilingual pretrained langauge models are hold by Huggingface, including mBERT[2], XLM-R[3] and XLM-R-L[4].

Since we are focusing on comparison of span representation across different layers and representation types, we conduct experiments with the 4-th, 8-th and 12-th layer of mBERT and XLM-R and the 8-th, 12-th and 24-th layer of XLM-R-L. All six representation types are considered for each layer-based models.

We run most of our experiments with an NVIDIA 1080ti GPU with 11GB memory, and use a NVIDIA A100 for XLM-R-L-based experiments. We finetune each experiment for 10 epochs with the learning rate set to 5e-5. We notice that the training process converges with training accuracy 1 in a short period. To reduce the effect of overfitting, we use a dropout probability of 0.5 before the classification layer. Our code is built over Allennlp2 and will be released on Github[5].

[2]BERT multilingual base (cased) : `https://huggingface.co/bert-base-multilingual-cased`

[3]XLM-RoBERTa (base-sized model): `https://huggingface.co/xlm-roberta-base`

[4]XLM-RoBERTa (large-sized model): `https://huggingface.co/xlm-roberta-large`

[5]`https://github.com/VisualJoyce/CiYi`

4.3.2 Results and Analyses for Subtask A

We list the overall experiment results in Table 4.3 in the Appendix. The table contains three main parts with each part showing the detailed experiment results for a multilingual pretrained language model. In each part, we test all six combinations of span representations using encoded sequences from different layers. To better illustrate our major conclusions, we select the best settings for each multilingual model from Table 4.3, and rearrange the zero-shot results to Table 4.1 and one-shot results to Table 4.2.

Table 4.1 shows us that using only endpoints of the span can be effective in predicting its idiomaticity and representation type **x,y,x-y** is a good choice for the zero-shot setting. We think representation using only endpoints is working well might due to most of the MWEs in current dataset consist of two words.

Table 4.2 shows us that representation type **MaxPooling** is a good choice for the one-shot setting and the best performance may be achieved using middle layers.

Combining both zero-shot setting and one-shot setting, we find that larger models are usually more effective in idiomaticity detection. For a specific pretrained model, using contextualized representation from a higher layer may not guarantee a better performance. For example, from the perspective of overall score for the One Shot scenario, the highest scores are all reached at the 8-th layer. However, we didn't observe a consistent advantage of using a specific representation type across different models and layers.

From the perspective of language, span-based models are achieving relative larger gains in both settings for GL. On one hand, the corpus used for training pretrained language models is not balanced across different languages. For example, in XLM-R, data from EN is several times than that of PT and hundrands times than that of GL. The data for GL may just surpass a minimal size for learning a BERT model and restricts performance in both settings for GL compared with PT and EN. On the other hand, this tells us that better span representation still help in detection of idiomaticity.

4.3.3 Endpoints-based Representation

This work focuses on the contextualized representation of the span of a target MWE. As pointed out by others, phrase representations, especially idioms, are not always compositional and rely more than the constituent words in the span. Not to mention, it is a much easier case which only uses the endpoints of the span. However, in both zero-shot setting and one-shot setting, we notice that endpoints-based methods works almost as well. We suspect this may due to the following reasons: (1) Endpoints of MWEs are highly correlated with these MWEs and can be very indicative about their representation. (2) Most of the MWEs covered in this dataset contain two words.

4.4 Conclusion

In conclusion, our experiments find that larger models are usually more effective in idiomaticity detection. And for a specific pretrained model, using contetualized representation from a higher layer may not guarantee a better performance. As the data used for multilingual pretrained language models is not well-balanced, rich resource languages have significant advantages over other languages. In the future, with the community contributing stronger language models with more balanced language distribution and more multilingual idiom-annotated datasets, idiomaticity detection still has large potentials to be explored from more angles.

Model	Type	Layer	Zero Shot				One Shot			
			EN	PT	GL	Avg	EN	PT	GL	Avg
mBERT	-	12	70.70	68.03	50.65	65.40	88.62	86.37	81.62	86.46
mBERT	x,y	4	75.11	69.63	64.20	72.49	86.32	85.17	76.50	83.84
mBERT	x,y,x-y	4	73.69	71.69	57.96	70.31	86.51	85.68	77.04	84.25
mBERT	x,y,x*y	4	76.76	70.67	60.27	71.69	87.76	86.15	80.16	85.93
mBERT	x,y,x*y,x-y	4	75.54	73.56	60.18	71.62	89.28	85.16	80.21	86.17
mBERT	SelfAttentive	4	72.13	73.19	62.16	70.79	85.48	82.86	78.76	83.50
mBERT	MaxPooling	4	71.27	73.11	58.46	69.49	85.23	83.40	76.56	82.93
mBERT	x,y	8	75.95	68.49	65.07	72.21	85.87	84.91	81.21	84.97
mBERT	x,y,x-y	8	72.45	66.88	61.95	69.11	86.86	83.95	82.47	85.41
mBERT	x,y,x*y	8	75.14	67.73	61.81	70.49	86.55	81.82	81.29	84.23
mBERT	x,y,x*y,x-y	8	72.59	73.18	61.91	70.87	86.09	84.21	81.30	84.79
mBERT	SelfAttentive	8	73.84	68.60	62.22	69.78	89.69	82.72	83.65	86.46
mBERT	MaxPooling	8	77.24	68.59	62.16	71.89	86.59	85.82	85.77	86.63
mBERT	x,y	12	76.31	70.77	58.80	70.36	86.45	84.06	79.69	84.47
mBERT	x,y,x-y	12	76.24	72.27	64.27	72.85	85.91	85.19	82.60	85.47
mBERT	x,y,x*y	12	74.04	71.76	64.24	71.65	87.58	84.27	79.92	85.00
mBERT	x,y,x*y,x-y	12	78.63	69.01	62.91	72.62	86.66	85.24	79.05	84.75
mBERT	SelfAttentive	12	75.03	69.71	60.75	70.32	86.31	82.62	83.69	85.14
mBERT	MaxPooling	12	75.33	71.08	59.00	69.90	88.37	85.38	81.19	86.07
XLM-R	x,y	4	80.70	65.29	54.57	68.82	89.26	80.22	73.15	82.48
XLM-R	x,y,x-y	4	79.49	67.51	54.98	69.20	89.27	82.26	74.10	83.47
XLM-R	x,y,x*y	4	78.66	70.77	57.66	71.19	88.38	79.47	70.03	80.79
XLM-R	x,y,x*y,x-y	4	74.10	67.11	56.48	68.49	88.30	81.13	73.96	82.73
XLM-R	SelfAttentive	4	**81.51**	70.99	55.49	71.91	88.46	80.84	74.82	82.78
XLM-R	MaxPooling	4	78.57	67.48	59.38	70.69	88.97	81.83	79.99	84.81
XLM-R	x,y	8	77.62	71.61	64.88	72.68	89.18	80.98	78.43	84.22
XLM-R	x,y,x-y	8	76.86	66.31	60.52	69.38	86.57	81.33	72.91	81.51
XLM-R	x,y,x*y	8	73.51	62.41	55.22	65.02	87.58	81.31	75.57	82.73
XLM-R	x,y,x*y,x-y	8	77.70	70.43	65.19	72.43	87.74	78.24	80.44	83.36
XLM-R	SelfAttentive	8	78.43	68.01	61.40	71.08	89.03	83.19	75.55	83.82
XLM-R	MaxPooling	8	76.61	68.45	64.50	71.35	89.49	83.71	82.19	86.17
XLM-R	x,y	12	76.95	66.35	57.73	68.54	86.66	81.73	77.73	83.15
XLM-R	x,y,x-y	12	75.98	63.26	55.31	66.31	86.12	79.82	73.51	80.92
XLM-R	x,y,x*y	12	77.70	70.18	61.05	71.05	87.65	79.54	74.17	81.83
XLM-R	x,y,x*y,x-y	12	78.07	71.51	59.04	70.91	88.19	82.09	76.30	83.45
XLM-R	SelfAttentive	12	76.16	70.54	62.92	71.36	90.05	79.77	77.26	83.82
XLM-R	MaxPooling	12	74.98	75.23	63.80	72.31	85.67	81.03	75.50	81.88
XLM-R-L	x,y	8	79.10	72.78	61.68	72.82	91.89	85.56	75.63	85.86
XLM-R-L	x,y,x-y	8	76.96	59.09	57.83	68.44	89.08	85.94	76.44	85.25
XLM-R-L	x,y,x*y	8	73.51	62.41	55.22	65.02	87.58	81.31	75.57	82.73
XLM-R-L	x,y,x*y,x-y	8	80.19	71.09	62.12	73.45	91.92	81.79	72.77	84.06
XLM-R-L	SelfAttentive	8	77.25	71.92	59.94	70.56	**92.66**	84.59	77.57	86.37
XLM-R-L	MaxPooling	8	77.83	70.92	61.18	71.40	89.85	81.79	69.14	81.71
XLM-R-L	x,y	12	76.92	70.40	60.11	70.17	90.26	85.19	82.76	87.10
XLM-R-L	x,y,x-y	12	77.48	67.52	60.25	69.85	92.24	81.30	81.00	86.30
XLM-R-L	x,y,x*y	12	80.54	65.49	55.46	68.72	91.43	83.91	78.78	86.00
XLM-R-L	x,y,x*y,x-y	12	79.77	69.66	60.84	71.54	90.28	84.42	83.46	87.14
XLM-R-L	SelfAttentive	12	78.13	74.44	61.92	72.63	90.48	86.23	78.90	86.43
XLM-R-L	MaxPooling	12	80.68	71.01	62.90	73.06	92.46	86.03	77.26	86.62
XLM-R-L	x,y	24	78.55	74.83	65.72	74.46	90.15	85.58	85.98	88.10
XLM-R-L	x,y,x-y	24	75.22	**75.80**	**69.01**	**74.66**	90.27	85.40	85.50	87.94
XLM-R-L	x,y,x*y	24	80.55	67.54	63.38	73.08	87.66	81.48	79.72	84.08
XLM-R-L	x,y,x*y,x-y	24	76.63	73.76	64.52	72.65	91.26	86.96	**89.06**	**89.79**
XLM-R-L	SelfAttentive	24	73.17	71.93	62.14	69.99	88.64	**87.81**	80.86	86.73
XLM-R-L	MaxPooling	24	75.39	72.33	66.15	72.26	89.30	85.47	85.39	87.55

Table 4.3: Experiment results for different multilingual pretrained models, in macro F1 score. We use bold font to highlight the maximum score across all settings and underline to highlight the maximum score in each part.

Chapter 5

Learning and Evaluating Chinese Idiom Embeddings

In this chapter, we study the task of learning and evaluating Chinese idiom embeddings. Considering that existing datasets for evaluating Chinese word embeddings have a low coverage of idioms, we first construct a new evaluation dataset that contains idiom synonyms and antonyms. Based on our observation that existing Chinese word embedding methods may not be suitable for learning idiom embeddings, we further present a BERT-based method that directly learns embedding vectors for individual idioms. We empirically compare representative existing methods and our method on our constructed evaluation dataset. We find that our method substantially outperforms existing methods on the evaluation dataset we have constructed. With extensive analysis using antonyms, we also find that our method is able to better distinguish idiom antonyms from synonyms than existing methods.

5.1 Introduction

As we know, the semantic meanings of Chengyu are often non-compositional and sometimes metaphoric. For example, the Chengyu 瓜田李下 literally means "melon field, beneath the plums," but its idiomatic meaning is to warn people to avoid

situations where a person may be easily suspected of wrongdoing. Chengyu are commonly used in modern Chinese language, and using computational methods to understand Chengyu plays an important role in Chinese language understanding. For example, a recent work studied how to improve essay writing with recommending Chinese idioms [80], and others studied how to improve reading comprehension by correcting usage of Chinese idioms [143] and differentiating synonyms of Chinese idioms [81]. In this chapter, we refer to Chengyu as Chinese idioms, although there are also other types of idioms in Chinese.

Recent years have witnessed the success of deep neural networks for many NLP tasks. A central idea behind deep neural networks for NLP is to use dense embedding vectors to represent language units including words, phrases and sentences, and such embeddings have been shown to be useful for many tasks such as sentiment analysis [156], question answering [48] and machine translation [166]. We therefore believe that it is also desirable to derive embedding vectors for Chinese idioms that can accurately capture their semantic meanings. However, it is not clear whether existing methods for Chinese word embeddings are effective in deriving good Chinese idiom embeddings, and there are at least two reasons for this.

First, existing Chinese word embedding evaluation datasets do not have sufficient coverage of idioms. For example, in the commonly used WordSim-240 [142] and WordSim-296 [19] datasets for Chinese word relatedness, no idiom is found. More recently, Huang et al. [58] released a COS960 dataset with similarities of Multiword Expressions (MWEs). Although COS960 covers 150 Chinese idioms, this is still a relatively small number, and only 20 MWE pairs in COS960 consist of both idioms. For the word analogy task, another commonly used evaluation task, Chen et al. [19] created the first Chinese dataset with 1,125 analogies, but no idiom is included. Li et al. [75] released a large and balanced dataset CA8 for word analogy. Although CA8 has 400 entries that contain idioms, they only cover 32 unique idioms and no idiom pairs are included. With this lack of coverage of idioms in existing evaluation datasets, we cannot judge whether existing Chinese word embedding methods work

well for Chinese idioms.

Second, it is reasonable to suspect that existing word embedding methods for Chinese have limitations that make them less suitable for Chinese idioms. For non-contextualized word embedding methods such as Continuous-Bag-Of-Words (CBOW) and Skip-Gram with Negative Sampling (SGNS), they treat contexts as bags of words, but given the complex meanings of Chinese idioms, learning their embeddings from bag-of-word representations of contextual words without considering the order and interactions between these contextual words may not be sufficient. Existing pre-trained non-contextualized Chinese word embeddings are also usually trained with a relatively small context window, but the semantic meaning of a Chinese idiom is often based on a larger context where the idiom appears. In fact, it has been observed that larger context windows result in more topicality [72, 9], and we suspect that for learning Chinese idiom embeddings a larger context window helps. Therefore, existing pre-trained non-contextualized Chinese word embeddings may not capture the semantic meanings of Chinese idioms well. On the other hand, recent contextualized word embedding methods such as BERT [31] and its variants (e.g., ERNIE [163]) consider longer contexts and use attention mechanism to model interactions between words, but since they do not focus on learning word embeddings, they do not learn a single embedding vector for each Chinese idiom. Although we can aggregate the character-level representations of the characters inside an idiom and treat the aggregated representation as the idiom embedding, since many Chinese idioms' semantics are non-compositional, this simplified approach is likely not ideal.

In this chapter, we study the problem of learning and evaluating Chinese idiom embeddings. To overcome the first challenge stated above, i.e., the lack of suitable evaluation dataset for Chinese idiom embeddings, we construct an evaluation dataset that contains Chinese idiom synonyms and antonyms. We also define two evaluation metrics to measure how close the ground truth idiom synonyms are in an embedding space in order to quantify the quality of the embedding space. To overcome the second challenge stated above, i.e., the potential limitations of existing word

embedding methods for Chinese idioms, we propose to adapt a method by Tan and Jiang [121] for Chinese idiom recommendation to learn idiom embeddings. This method learns a single embedding vector directly for each idiom and encodes the contextual information using BERT.

With the evaluation dataset we have created, we empirically compare a SGNS-based non-contextualized word embedding method for Chinese, two variants of BERT for Chinese, and our Chinese idiom embedding method. We find that based on the two metrics we have defined to measure closeness of synonyms in an embedding space, our method performs substantially better than existing methods. We also find that our method can better distinguish idiom antonyms from idiom synonyms than existing embedding methods. We also conduct further analysis to demonstrate that embedding methods that rely more on Chinese character information show advantages only when the synonyms share many common characters.

The contributions of our work are twofold: (1) We construct an evaluation dataset to facilitate the evaluation of Chinese idiom embeddings. Code and data are released on github[1]. (2) We present a BERT-based method that directly learns Chinese idiom embeddings, and we empirically compare this method with existing Chinese word embedding methods to demonstrate both the importance of learning a single embedding vector for an entire idiom and the importance of using BERT to encode the context when learning these idiom embeddings.

5.2 Construction of the Evaluation Dataset

A standard intrinsic task for evaluating word embeddings is word similarity [5, 134]. For Chinese idioms, a natural choice of idiom pairs that are semantically similar are synonyms or near-synonyms[2]. Although previously Wang et al. [136] constructed a

[1] `https://github.com/VisualJoyce/ChengyuBERT`

[2] We use near-synonyms to refer to idioms that do not have exactly the same meaning but their meanings are highly similar. It is not common to have Chinese idioms that are complete synonyms, except for those that are variants of the same basic form.

Chinese idiom knowledge base that contains idiom synonyms, this knowledge base is not publicly available. On the other hand, there exist online resources containing synonyms and near-synonyms of Chinese idioms. We choose two websites, kxue.com (快学网)[3] and Baidu Baike (百度百科)[4], as the sources from which to crawl idiom synonyms and near-synonyms. We also collect idiom antonyms from these two websites because an antonym of an idiom is often topically related to that idiom and therefore may be also close to that idiom in an embedding space. However, we expect a good idiom embedding method to be able to separate antonyms from synonyms.

Idiom Vocabulary: According to Wang et al. [136], there are in total around 38K Chinese idioms, among which around 3.5K are commonly used. In order to obtain a vocabulary of Chinese idioms with high coverage, we merge the idioms found in the following four resources: (1) Chengyu Daquan[5], (2) Xinhua Chengyu Dictionary[6], (3) Chengyu Cloze Test[7], and (4) ChID. [8]. This gives us a Chinese idiom vocabulary with 33,237 idioms.

ChIdSyn: As we have pointed out earlier, we believe idiom synonyms can help us evaluate idiom embeddings. To construct a large dataset of Chinese idiom synonyms, we crawled synonyms from two websites: (1) Kxue.com is an online Chinese thesaurus. It has a dedicated page where Chinese idiom synonyms are listed. Each entry in this list consists of a key and a value, where the key is a Chinese idiom and the value is one or more other Chinese idioms that are near-synonyms of the key. We crawled all the entries from this idiom synonym page on kxue.come[9]. Baidu Baike is an online encyclopedia in Chinese. For each idiom, there is a section called 成语辨

[3] http://chengyu.kxue.com/
[4] https://baike.baidu.com/
[5] www.guoxue.com/chengyu/CYML.htm
[6] github.com/pwxcoo/chinese-xinhua
[7] github.com/bazingagin/chengyu_data
[8] https://github.com/zhengcj1/ChID-Dataset
[9] We crawled the data from http://chengyu.kxue.com/list/jinyici.html before October 19, 2020.

析 (Chengyu Differentiation) that lists its synonyms and antonyms.[10] We crawled the synonyms of those idioms in our vocabulary that can be found on Baidu Baike. In total, we obtained around 30k entries of Chinese synonyms. We then removed those idioms in the data that are not in our idiom vocabulary as described earlier. In the end we obtained a total of around 21K entries in our synonym dataset, where each entry consists of a *query idiom* and a set of other idioms that are the query idiom's synonyms or near-synonyms.

We observe that a significant portion of the synonyms share common characters with the query idioms. For example, 山盟海誓 (oath of eternal love) and 海誓山盟 are treated as near-synonyms in our dataset, but these two idioms contain exactly the same set of Chinese characters. In fact, they are variants of the same basic form. Another example is 挨家挨户 (door to door) and 挨门挨户, which share three common characters. In general, it is not uncommon for Chinese idioms to have such variants due to historical reasons such as misuse (including literary malapropism). Although these are valid near-synonyms, we suspect that they may affect the evaluation of idiom embeddings. This is because those idiom embeddings that rely more on character-level information are likely to gain advantages when evaluated on these near-synonym pairs sharing common characters. For example, if an idiom embedding is obtained by averaging the character embeddings of its component characters, then it is very easy for this type of idiom embeddings to recognize that 山盟海誓 and 海誓山盟 are near-synonyms (because they would have the same average character embedding), but we would not be able to know whether such embeddings truly capture the semantic meanings. We also suspect that for those idioms that have near-synonyms sharing common characters, their semantic meanings are more likely to be compositional and thus less idiomatic. For example, for the idiom 挨家挨户, the character 挨 means "in sequence" and both 家 and 户 mean "household." The meaning of the idiom, which is "door to door," can be directly inferred from the meanings of the characters. Therefore, when the character

[10]For example, for the idiom "一马平川", see https://baike.baidu.com/item/一马平川.

家 (household) is replaced with the character 门 (door), the meaning of the idiom remains the same.

Consequently, we move those synonyms that share at least two common characters with the query idioms into a separate dataset, which we will not use as the main evaluation dataset. The remaining synonyms always have no more than one common character with their query idioms. We refer to this cleaned synonym dataset as *ChIdSyn*, and the separate dataset containing synonyms sharing two or more common characters is referred to as *ChIdSyn-com*. We will use *ChIdSyn-com* for additional analysis in our experiments. Statistics of *ChIdSyn* and *ChIdSyn-com* can be found in Table 5.1.

	Before Filtering		After Filtering	
	#Idioms	#Entries	#Idioms	#Entries
Crawled	33,524	30,354	21,745	20,753
ChIdSyn	11,387	8,897	8,125	6,822
ChIdSyn-com	28,622	24,147	18,498	15,836
ChIdAnt	11,263	9,733	7,939	7,316

Table 5.1: Statistics of the crawled datasets. *Crawled* refers to synonyms and near-synonyms. We list antonyms separately in the last line of the table.

ChIdAnt: From the same two websites, we have also collected around 10K entries in an antonym dataset which we refer to as *ChIdAnt*. Similarly, each entry in this dataset consists of a query idiom and its antonyms. Although antonyms are idioms having opposite meanings, they are often topically closely related. For example, the idiom 饱学之士 means "a scholarly man," and its antonym 胸无点墨 means "uneducated." We can see that their meanings are topically closely related. We therefore suspect that they are still close in an embedding space, but ideally a good idiom embedding method should be able to distinguish the synonyms of a query idiom from its antonyms. Table 5.1 gives some statistics of *ChIdAnt*.

5.3 Learning Chinese Idiom Embeddings

Existing Chinese word embedding methods can be used to derive idiom embeddings. However, as we have discussed in Section 5.1, they may not be ideal for learning Chinese idiom embeddings. In this section, we refer readers to Section 2.4 for existing Chinese word embedding methods. We then present a method to learn Chinese idiom embeddings based on BERT. Our proposed method is adapted from a method for Chinese idiom recommendation [121].

The original Chinese-BERT starts from embeddings of individual Chinese characters at the bottom layer. When BERT-wwm or ERNIE is applied to Chinese, although words are identified and masked using Chinese segmentation tools, the model still does not learn embedding vectors directly for entire words. Therefore, to obtain an embedding for an idiom, we need to aggregate the component characters' embeddings.

In this chapter, we take the vector representations of individual characters at the top layer of BERT, and average these character representations as the embedding for the entire idiom.[11]

5.3.1 Learning Idiom Embeddings with BERT

As we have pointed out earlier, existing non-contextualized Chinese word embedding methods model contextual words in a bag-of-word manner, which is suboptimal for encoding the contextual information. Chinese-BERT and its variants can better encode the contextual information using the Transformer architecture, but they do not learn a single embedding vector for an entire Chinese idiom, and therefore they are not ideal either because idioms often have non-compositional semantics. We propose to combine BERT contextual encoding with single embedding vectors for Chinese idioms.

[11] We have also experimented with another setting where we use the `[CLS]` token's representation at the top layer as the idiom representation. We found this to perform worse than using average character embedding.

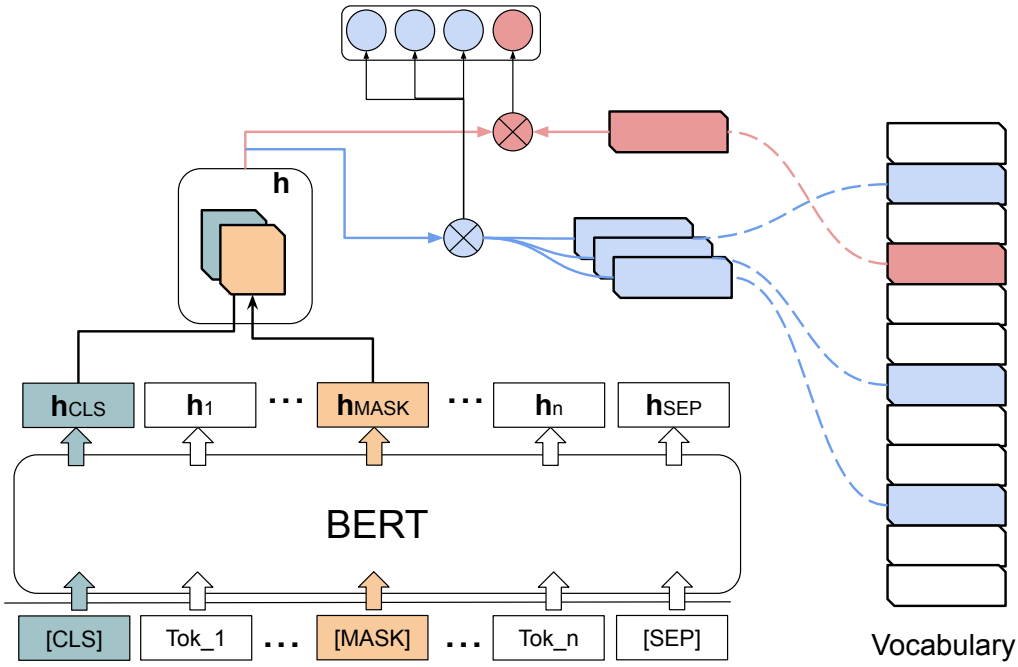


Figure 5.1: Model structure for BERT with SGNS. The red flow shows the path for the target idiom while the light blue flows show paths for negative sampled idioms used for the learning.

Specifically, to train idiom embeddings, we perform the task of idiom prediction based on its context. Given an idiom v appearing in a context window $c = (w_{-k}, \ldots, w_{-2}, w_{-1}, \texttt{[MASK]}, w_1, w_2, \ldots, w_k)$, where w_i are the contextual words and `[MASK]` replaces the idiom v in the original text, the task aims to predict v based on c. To do so, our idea is to assume that v has an embedding vector $\mathbf{e}_v$ to be learned. We then use BERT to derive a hidden representation $\mathbf{h}$ that represents c and use $\mathbf{h}$ and $\mathbf{e}_v$ to derive a log-linear score to indicate how likely v fits into the context c.

Note that the task described above is similar to the prediction task used by CBOW, but instead of simply using the average word embedding to represent the context c, our method uses BERT to encode c. The task described above is also similar to the Masked Language Model task of BERT, but we mask and predict whole idioms rather than individual characters.

Concretely, to use BERT to encode the sequence c, following standard practice, we prepend the token `[CLS]` to the beginning of c and append `[SEP]` to the end

of c. We also include position embeddings. For segment embeddings, we treat the sequence c as a single segment. Let $\mathbf{h}_{\texttt{[CLS]}} \in \mathbb{R}^d$ denote the hidden vector produced by the last layer of BERT representing `[CLS]`, and $\mathbf{h}_{\texttt{[MASK]}} \in \mathbb{R}^d$ the similarly produced hidden vector representing `[MASK]`. We then define the following vector $\mathbf{h}$ to combine $\mathbf{h}_{\texttt{[CLS]}}$ and $\mathbf{h}_{\texttt{[MASK]}}$ into a single vector representation because both are important for representing the context c, $\mathbf{h} = \mathbf{W}[\mathbf{h}_{\texttt{[CLS]}}; \mathbf{h}_{\texttt{[MASK]}}; \mathbf{h}_{\texttt{[CLS]}} \odot \mathbf{h}_{\texttt{[MASK]}}; \mathbf{h}_{\texttt{[CLS]}} - \mathbf{h}_{\texttt{[MASK]}}]$, 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 then use a standard log-linear model based on the dot product between $\mathbf{h}$ and $\mathbf{e}_v$ to train our model. To use the hidden representation $\mathbf{h}$ of the context to predict the idiom v, we take its idiom embedding $\mathbf{e}_v$, apply Layer Normalization [2] LN on it. We also adopt negative sampling to select negative Chengyu. The learning objective is defined as

$$-(\log \sigma(LN(\mathbf{e}_v)^\intercal \mathbf{h}) + \sum_{v' \in \mathcal{N}_v} \log \sigma(-LN(\mathbf{e}_{v'})^\intercal \mathbf{h})), \tag{5.1}$$

where $\mathcal{N}_v$ contains a fixed number of negative samples for each Chinese idiom, and $\sigma(\cdot)$ is the sigmoid function. Besides the transformation $\mathbf{W}$ and LN, during the training process, the BERT layers will be finetuned and the whole vocabulary will be learned from random initialization. The model structure is illustrated in Figure 5.1.

5.4 Experiments

5.4.1 Experiment Setup

Evaluation metrics: Recall that our main evaluation dataset is the *ChIdSyn* dataset that contains entries of query idioms and their near-synonyms, where these near-synonyms share at most one common character with the query idiom. We design two evaluation metrics to measure whether near-synonyms in *ChIdSyn* are close to each other in an embedding space. (1) ***Recall@K***: Given a query idiom v_n, we rank all idioms based on their idiom embeddings' cosine or Euclidean distances with the

	Recall@K								Coherence@K					
	Cosine				Euclidean				Cosine			Euclidean		
	1	3	5	10	1	3	5	10	3	5	10	3	5	10
SGNS	0.054	0.102	0.132	0.178	0.031	0.056	0.071	0.092	0.038	0.043	0.045	0.027	0.031	0.036
SGNS+C	0.030	0.084	0.127	0.198	0.009	0.022	0.030	0.048	0.032	0.038	0.043	0.023	0.029	0.038
SGNS+B	0.067	0.127	0.159	0.210	0.043	0.080	0.101	0.131	0.047	0.051	0.053	0.034	0.038	0.042
SGNS+B+C	0.051	0.128	0.184	0.271	0.017	0.046	0.063	0.089	0.043	0.055	0.059	0.030	0.041	0.047
BERT-wwm	0.031	0.084	0.117	0.170	0.030	0.078	0.111	0.163	0.028	0.034	0.037	0.026	0.030	0.034
ERNIE	0.037	0.109	0.161	0.238	0.036	0.110	0.163	0.244	0.038	0.048	0.058	0.037	0.049	0.060
Ours-16	0.145	0.282	0.357	0.451	0.142	0.275	0.348	0.433	0.107	0.113	0.113	0.105	0.109	0.110
Ours-32	**0.164**	**0.327**	**0.411**	**0.519**	**0.163**	**0.322**	**0.404**	**0.503**	**0.126**	**0.137**	**0.142**	**0.123**	**0.136**	**0.139**

Table 5.2: *Recall@K* and *Coherence*@K on *ChIdSyn*, where ranking is based on either cosine or Euclidean distance.

query idiom's embedding. Let $\mathcal{R}_{v_n}^{(K)}$ represent the top-K ranked idioms. Let $\mathcal{S}_{v_n}$ denote the set of ground truth near-synonyms of v_n. *Recall@K* is defined as

$$Recall@K = \frac{1}{N}\sum_{n=1}^{N}\frac{|\mathcal{S}_{v_n} \cap \mathcal{R}_{v_n}^{(K)}|}{|\mathcal{S}_{v_n}|}, \tag{5.2}$$

where N is the total number of query idioms in *ChIdSyn*. (2) ***Coherence@K***: However, it is not guaranteed that all near-synonyms of a query idiom v are identified in the online resources we crawled, i.e., some of the top-K ranked idioms may be indeed near-synonyms but are not found in the ground truth near-synonym set. To overcome this limitation, we can measure whether a query idiom and its ground truth near-synonyms share many common "similar" idioms. In this way, even if a real near-synonym u of idiom v is missed from the ground truth, if u is found to be similar to both v and its ground truth near-synonyms, it will contribute positively to the metric. We therefore define the following metric, which we call *Coherence@K*:

$$Coherence@K = \frac{1}{N}\sum_{n=1}^{N}\frac{|\cap_{u\in\mathcal{S'}_{v_n}}\mathcal{R}_u^{(K)}|}{|\cup_{u\in\mathcal{S'}_{v_n}}\mathcal{R}_u^{(K)}|}, \tag{5.3}$$

where v_n is a query idiom, N is the total number of query idioms, $\mathcal{S'}_{v_n} = \{v_n\} \cup \mathcal{S}_{v_n}$ (i.e., v_n together with its ground truth near-synonyms), and $\mathcal{R}_u^{(K)}$ is the top-K similar idioms to u, where similarity can be based on either cosine or Euclidean distance.

Methods to be compared: We empirically compare the following embedding methods: (1) **SGNS** and its variants: We use Chinese word embeddings released

by Li et al. [75], which are trained using the Skip-Gram with Negative Sampling method. There are a few variations of these embeddings. **SGNS+B** uses bigram prediction, **SGNS+C** incorporates character information, and **SGNS+B+C** uses both bigram prediction and character information. Li et al. [75] also experimented with different genres of text for training. In this chapter, we use their pre-trained word embeddings trained on the literature genre because this provides fair comparison with our method, which is also trained on Chinese text in the literature genre. (2) **BERT-wwm**: This refers to averaging the top-layer character representations after using the pre-trained Chinese-BERT-wwm [27] to process an idiom. (3) **ERNIE**: This refers to averaging the top-layer character representations after using Chinese ERNIE [163] to process an idiom. (4) **Ours-16**: This is our method where we set the context window size to be 16 characters. (5) **Ours-32**: This is also our method with a larger context window of 32 characters.

Training data: We collect online ebooks from the literature domain with a size comparable to that of the training corpus used by Li et al. [75]. We extract sentences from our crawled corpus and keep only those sentences containing idioms. Since the average word length for Chinese is around 1.6 characters, we use a window size of 8 characters on each side, i.e., 16 characters in total, which is comparable to the SGNS method that used a window size of 5 words on each side. To test how context length may affect the results, we also train our model using a larger window size of 16 characters on each side, i.e, 32 characters in total. The two versions of our model are named **Ours-16** and **Ours-32**, respectively. To ensure fair comparison, we use only the subset of the entries from *ChIdSyn* where we have idiom embeddings from all methods. This results in a subset of 3,716 entries from *ChIdSyn* for our experiments, which is still a relatively large number. Similarly, for some further analysis we do using *ChIdSyn-com*, we also use only a subset of the data, which contains 2,342 entries. A subset of *ChIdSyn-Ant* with 3940 entries is also used for further analysis.

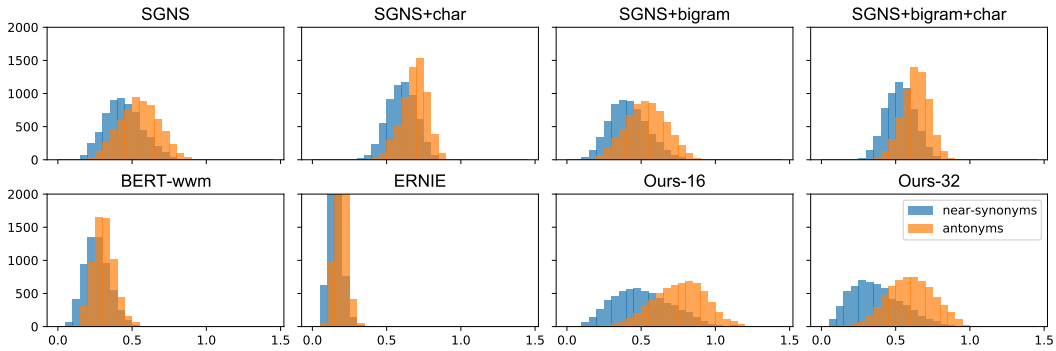


Figure 5.2: Cosine distance distribution of near-synonym and antonym pairs.

5.4.2 Main Results

We first present the results of all the methods we compare using the metrics *Recall@K* and *Coherence@K* on *ChIdSyn*, see Table 5.2. We can draw the following major conclusions from the table: (1) If we compare **Ours-16** with the **SGNS** methods, we can see that Ours-16 clearly outperforms these SGNS methods. Recall that we use a similar context window size as the SGNS methods. The main difference of Ours-16 from the SGNS methods is that we use Chinese-BERT to encode the context whereas the SGNS methods do not model the interactions between the contextual words. This implies that when learning Chinese idiom embeddings, it is important to model the order of and interactions between the contextual words. (2) Comparing **Ours-16** with **BERT-wwm** and **ERNIE**, we can see that Ours-16 also substantially outperforms these two BERT-based methods. Recall that the main difference of our method and these BERT methods is that we directly learn a single idiom embedding vector whereas for these BERT methods we need to aggregate character embeddings to derive idiom embeddings. The results suggest that many Chinese idioms' semantic meanings cannot be simply derived from their character embeddings and therefore it is important to associate a Chinese idiom with a single embedding vector and to learn this embedding vector from the contexts of this idiom. (3) **Ours-32** performs clearly better than **Ours-16**. This suggests that a larger context window is very useful for learning Chinese idiom embeddings, which have not been found to be the case for word embeddings [76].

Besides the major conclusions drawn above, we can also see from the two tables that: (1) For the SGNS methods, adding character information may actually either hurt the performance or improve the performance very little. In other words, there is no consistent observation that character information helps for Chinese idiom embeddings, which is not the case for Chinese word embeddings [19, 164]. This verifies our hypothesis that existing conclusions drawn from evaluating Chinese word embeddings may not apply to idiom embeddings. (2) For the two BERT-based methods, we can see that ERNIE performs clearly better than BERT-wwm. It is worth noticing that ERNIE uses Baidu Baike in which most idioms have entries and would be treated as entities by the entity-level mask. Intuitively, the embeddings extracted using ERNIE should be better than BERT-WWM, whose CWS tools may not be able to recognize all the idioms.

5.4.3 Further Analysis

In this section, we conduct some further comparison and analysis using *ChIdSyn-com* and *ChIdAnt*.

Synonyms with Common Characters: Recall that we identified a set of near-synonyms that share two or more common characters. We suspect that these idiom synonyms are easier to be identified if the idiom embeddings rely more on character-level information. To verify this hypothesis, we compare the various methods using *Recall@K* based on cosine distance on *ChIdSyn-com*. The results are shown in Table 5.3. We can see that indeed those existing methods that rely more on character-level information, namely, SGNS+C, SGNS+B+C, BERT-wwm and ERNIE generally perform better than the other methods, including our methods. This verifies our hypothesis above. Note that because the synonyms in *ChIdSyn-com* share many common characters, being able to identify them does not imply that the embeddings truly capture the semantic meanings of the idioms. Since SGNS+C, SGNS+B+C, BERT-wwm and ERNIE actually do not perform well on *ChIdSyn*, we

K	1	3	5	10
SGNS	0.130	0.223	0.270	0.334
SGNS+B	0.175	0.287	0.341	0.404
SGNS+C	0.518	0.775	**0.857**	**0.924**
SGNS+B+C	**0.526**	**0.776**	0.846	0.908
BERT-wwm	0.467	0.662	0.714	0.786
ERNIE	0.531	0.760	0.825	0.880
Ours-16	0.380	0.555	0.612	0.675
Ours-32	0.449	0.655	0.722	0.786

Table 5.3: $Recall@K$ on *ChIdSyn-com*.

argue that they are effective only for synonyms sharing many common characters, and this implies that they rely on superficial patterns to encode idioms.

Antonyms: Recall that earlier we raised the hypothesis that good idiom embedding methods should be able to distinguish antonyms from synonyms, although both can be topically related to the query idioms. In fact, a previous study by Samenko et al. [107] also found that embeddings contain information that distinguishes synonyms and antonyms. Inspired by them, we think that the separability of near-synonyms and antonyms may reflect the quality of the learned embeddings. We therefore visualize the distributions of cosine distances (i.e, 1 minus cosine similarity) of idiom near-synonym pairs and antonym pairs in Figure 5.2, using *ChIdSyn* and *ChIdAnt*. We can see from the figure that our methods **Ours-16** and **Ours-32** clearly has a distinguishable cosine distance distribution for antonyms compared with synonyms, whereas for the other methods the two distributions are less distinguishable. This again demonstrates the advantage of our idiom embedding methods.

5.5 Conclusion

In this chapter, we constructed a new evaluation dataset that contains Chinese idiom synonyms and antonyms to facilitate the evaluation of Chinese idiom embeddings. We presented a method that learns Chinese idiom embeddings by predicting idioms based on BERT-encoded contexts. We also propose two metrics to measure closeness

of synonyms in the embedding space. Our method performs substantially better than existing methods.

Part II

Neural Network-based Applications for Chinese Idioms

Chapter 6

A BERT-based Dual Embedding Model for Chinese Idiom Prediction

Starting from this chapter, we will study neural network-based applications in Chinese idioms. The Chinese idiom prediction task is to select the correct idiom from a set of candidate idioms given a context with a blank. We propose a BERT-based dual embedding model to encode the contextual words as well as to learn dual embeddings of the idioms. Specifically, we first match the embedding of each candidate idiom with the hidden representation corresponding to the blank in the context. We then match the embedding of each candidate idiom with the hidden representations of all the tokens in the context thorough context pooling. We further propose to use two separate idiom embeddings for the two kinds of matching. Experiments on a recently released Chinese idiom cloze test dataset show that our proposed method performs better than the existing state of the art. Ablation experiments also show that both context pooling and dual embedding contribute to the improvement of performance.

6.1 Introduction

In this chapter, we propose a BERT-based dual embedding model for the Chinese idiom prediction task. We first present two baseline models that use BERT to process

and match passages and candidate answers in order to rank the candidates. Observing that these baselines do not explicitly model the global, long-range contextual information in the given passage for Chinese idiom prediction, we propose a context-aware pooling operation to force the model to explicitly consider all contextual words when matching a candidate idiom with the passage. Furthermore, we propose to split the embedding vector of each Chinese idiom into two separate vectors, one modeling its local properties and the other modeling its global properties. We expect the embedding for local properties to capture the syntactic properties of an idiom, while the embedding for global properties to capture its topical meaning.

To evaluate the effectiveness of the BERT-based dual embedding model, we conduct experiments on the ChID dataset. Our experiments show that our method can outperform several existing methods tested by ChID [165] as well as our baseline methods. We also find that both context-aware pooling and dual embedding contribute to the performance improvement. To prove the competency of our model, we also evaluate it against a public leaderboard of ChID Competition. The results show that our model is competitive compared to the top-ranked systems. We can also achieve better performance with a large margin compared with several methods using pretrained language models. We also conduct further analysis using a gradient-based attribution method to check if our method can indeed capture global information to make correct predictions. Some case studies show that indeed our method makes use of more global contextual information to make predictions.

6.2 Method

6.2.1 Task Definition and Dataset

We formally define the Chinese idiom prediction task as follows. Given a passage P, represented as a sequence of tokens $(p_1, p_2, \ldots, p_n)$, where each token is a Chinese character and one of the tokens is a special "blank" token `[MASK]`, and given a set

of K candidate Chinese idioms $\mathcal{A} = \{a_1, a_2, \ldots, a_K\}$, our goal is to select an idiom $a^* \in \mathcal{A}$ that best fits the blank in P. See the example in Table 2.2.

We assume that a set of training examples in the form of triplets, each containing a passage, a candidate set and the ground truth answer, is given. We denote the training data as $\{(P_i, \mathcal{A}_i, a_i^*)\}_{i=1}^N$. We use $\mathcal{V}$ to denote the vocabulary of all Chinese idioms observed in the training data, i.e., $\mathcal{V} = \cup_{i=1}^N \mathcal{A}_i$.

To facilitate the study of Chinese idiom comprehension using deep learning models, [165] released the ChID dataset. The dataset was created in the "cloze" style. The authors collected diverse passages from novels and essays on the Internet and news articles from THUCTC [45]. The authors then masked Chinese idioms found in these passages using the blank token. To construct the candidate answer set for each blank, the authors considered synonyms, near-synonyms and other idioms either irrelevant or opposite in meaning to the ground truth idiom. See Table 2.2 for examples of candidate answers.

6.2.2 BERT Baselines

We first present two BERT-based baseline solutions. Given the widespread use of BERT for many NLP tasks, these baselines can be regarded as standard ways to solve the Chinese idiom prediction problem. We also present a heuristic using enlarged candidate set that can be applied to the second BERT baseline.

BERT Baseline with Idioms as Character Sequences: A straightforward way to apply BERT for Chinese idiom prediction is as follows. Given a passage $P = (p_1, p_2, \ldots, \texttt{[MASK]}, \ldots, p_n)$ and a candidate answer $a_k \in \mathcal{A}$, we first concatenate them into a single sequence $(\texttt{[CLS]}, p_1, p_2, \ldots, p_n, \texttt{[SEP]}, a_{k,1}, a_{k,2}, a_{k,3}, a_{k,4}, \texttt{[SEP]})$, where $a_{k,1}$ to $a_{k,4}$ are the four Chinese characters that idiom a_k is composed of. We can then directly use BERT to process this sequence and obtain the hidden representation for `[CLS]` on the last layer, denoted by $\mathbf{h}_{k,0}^L \in \mathbb{R}^d$. To select the best answer idiom, we first use a linear layer to process $\mathbf{h}_{k,0}^L$ for $k = 1, 2, \ldots, K$

and then use standard softmax to obtain the probabilities of each candidate. To train the model, we use standard negative log likelihood as the loss function.

BERT Baseline with Idiom Embeddings: Many Chinese idioms are metaphors and therefore their meanings should not be directly derived from the embeddings of its four individual characters, as the baseline above does. E.g., "狐假虎威" literally means a fox assuming the majesty of a tiger, but it is usually used to describe someone flaunting his powerful connections. Therefore, learning a single embedding vector for the entire idiom can help the understanding of idioms.

In this BERT baseline, instead of concatenating the passage and a candidate answer into a single sequence for BERT to process, we keep them separated. We only use BERT to process the passage sequence $(\texttt{[CLS]}, p_1, p_2, \ldots, \texttt{[MASK]}, \ldots, p_n, \texttt{[SEP]})$. Afterwards, we use the hidden representation of `[MASK]`, denoted as $\mathbf{h}_b^L$, to match each candidate answer. In this way, no matter how many candidate answers there are, BERT is used to process the passage only once. On the other hand, each Chinese idiom has a hidden embedding vector, which is to be learned.

We use $\mathbf{a}_k$ to denote the embedding vector for candidate $a_i \in \mathcal{A}$. The hidden representation $\mathbf{h}_b^L$ is fused with each candidate idiom via element-wise multiplication. Then the probability to selection a_k among all the candidates $\mathcal{A}$ is defined as follows:

$$p_k = \frac{\exp(\mathbf{w} \cdot (\mathbf{a}_k \odot \mathbf{h}_b^L) + b)}{\sum_{k'=1}^{K} \exp(\mathbf{w} \cdot (\mathbf{a}_{k'} \odot \mathbf{h}_b^L) + b)}. \tag{6.1}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are model parameters, and $\odot$ is element-wise multiplication. To train the model, we again use negative log likelihood as the loss function.

Heuristic with Enlarged Candidate Set: The ChID dataset uses only a small set of negative answers in each candidate set. It is reasonable to expect that most of the other Chinese idioms not in the candidate set are also negative answers and including

them in the training data may help. We therefore use a heuristic that considers an enlarged candidate set to further boost the performance.

To apply this heuristic, we define a candidate set $\mathcal{A}'$ to be the same as $\mathcal{V}$ (i.e., the vocabulary containing all Chinese idioms observed in the training data), and then define a second term in the loss function that is the negative log likelihood of selecting the correct answer from this enlarged candidate set.

Note that because $\mathcal{A}'$ is large, this heuristic is not feasible to be applied to the character sequence-based BERT baseline, because it would require inserting each candidate into the passage for BERT to process, which would be computationally too expensive. Therefore, this enlarged candidate set heuristic is only applied to the idiom embedding-based BERT baseline. Specifically, we can define the probability of selecting answer $a \in \mathcal{A}'$ as follows:

$$q_a = \frac{\exp(\mathbf{a} \cdot \mathbf{h}_b^L)}{\sum_{c \in \mathcal{A}'} \exp(\mathbf{c} \cdot \mathbf{h}_b^L)}. \tag{6.2}$$

Let q_i^* denote the probability of selecting the ground truth idiom among all candidates in $\mathcal{A}'$ for the i-th training example, and p_i^* denote the probability of selecting the correct answer among the original candidate set $\mathcal{A}$ for the i-th training example. Our training loss function is then defined as follows:

$$L = -\sum_{i=1}^{N} (\log(p_i^*) + \log(q_i^*)). \tag{6.3}$$

6.2.3 Our Dual Embedding Model

The BERT baselines presented above are reasonable baselines, but they have a potential problem. We observe that in order for an idiom to fit into a passage well, it has to not only grammatically (i.e., syntactically) fit into the local context surrounding the `[MASK]` token but also show semantic relevance to the whole passage. In the example shown in Table 2.2, a correct answer has to first be an adjective rather than, say, a noun or a verb. In addition, given the global context of the entire passage, it is understood that the correct answer should convey the meaning of "disorganized".

Based on the observation above, we introduce the following two changes to the second BERT baseline, i.e., the idiom embedding-based BERT baseline, introduced in Section 6.2.2.

Context-aware Pooling

As we have pointed out earlier, oftentimes Chinese idioms have metaphorical meanings, and to evaluate whether a Chinese idiom is suitable in a passage, we need to understand the semantic meaning of the entire passage. Therefore, it is important for us to not only try to match an idiom with the local context it is to be placed in (which can roughly be modeled by $\mathbf{h}_b^L$) but also to match it with the entire passage. Let us use $\mathbf{a}_k$ to denote the embedding for idiom a_k. Recall that $\mathbf{H}^L = (\mathbf{h}_0^L, \mathbf{h}_1^L, \ldots, \mathbf{h}_n^L)$ represents the hidden states of the last layer of BERT after it processes the passage sequence. Our method with context-aware pooling can be represented as follows:

$$p_k = \frac{\exp(\mathbf{a}_k \cdot \mathbf{h}_b^L + \max_{i=0}^{n}(\mathbf{a}_k \cdot \mathbf{h}_i^L))}{\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'} \cdot \mathbf{h}_b^L + \max_{i=0}^{n}(\mathbf{a}_{k'} \cdot \mathbf{h}_i^L))}. \tag{6.4}$$

Dual Embeddings

Because we need to match an idiom with both $\mathbf{h}_b^L$ and the entire passage, the second idea we propose is to split the embedding of an idiom into two "sub-embedding" vectors, which we refer to as "dual embeddings". Let us use $\mathbf{a}_k^u$ and $\mathbf{a}_k^v$ to denote the two embeddings for idiom a_k.

We then calculate the probability of selecting candidate a_k as follows:

$$p_k = \frac{\exp(\mathbf{a}_k^u \cdot \mathbf{h}_b^L + \max_{i=0}^{n}(\mathbf{a}_k^v \cdot \mathbf{h}_i^L))}{\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'}^u \cdot \mathbf{h}_b^L + \max_{i=0}^{n}(\mathbf{a}_{k'}^v \cdot \mathbf{h}_i^L))}. \tag{6.5}$$

We also adopt the heuristic of enlarged candidate set from Section 6.2.2. With the candidate set $\mathcal{A}'$ to be the same as $\mathcal{V}$, we still use dual embeddings to represent each idiom, but when we match the dual embeddings with the passage, we use both $\mathbf{a}^u$ and $\mathbf{a}^v$ to match $\mathbf{h}_b^L$ only. This is because it would be too expensive to match $\mathbf{a}^v$ of each candidate with the entire sequence of hidden states $\mathbf{H}^L$ as we now have many

candidates. So we define the probability of selecting answer $a \in \mathcal{A}'$, i.e., selecting the ground truth answer from the entire vocabulary of Chinese idioms, as follows:

$$q_a = \frac{\exp(\mathbf{a}^u \cdot \mathbf{h}_b^L + \mathbf{a}^v \cdot \mathbf{h}_b^L)}{\sum_{c \in \mathcal{A}'} \exp(\mathbf{c}^u \cdot \mathbf{h}_b^L + \mathbf{c}^v \cdot \mathbf{h}_b^L)}. \tag{6.6}$$

Similarly, to train the model, we use negative log likelihood as shown before.

6.3 Experiments

In this section, we first evaluate our proposed dual embedding method using the ChID Official dataset. We will compare the results of our proposed method with the models of earlier literature as well as the two BERT baselines presented earlier. Then we report our performance on the leaderboard of ChID Competition against several high ranked systems to further illustrate the competency of our method. Details of the ChID dataset can be found in Section 2.6.

6.3.1 Experiment Settings

Methods Compared: We compare the following different methods. The first three are baselines adopted by [165]:

Language Model (LM): This method is based on standard bidirectional LSTM (BiLSTM) [56, 167]. It uses BiLSTM to encode the given passage and compares it with the embedding vector of each candidate idiom in order to select the best idiom.

Attentive Reader (AR): This method also uses BiLSTM but augments it with attention mechanism. It is based on the Attentive Reader model by [51].

Standard Attentive Reader (SAR): This is an altered version of Attentive Reader, where attention weights are computed using a bilinear matrix. It is based on [17].

BL-CharSeq: This is the first BERT baseline treating idioms as character sequences.

BL-IdmEmb (w/o EC): This is the second BERT baseline using idiom embeddings. In this version, we do not use enlarged candidate set.

BL-IdmEmb: This baseline is the same as BL-CharSeq-IdmEmb (w/o EC) but incorporates the heuristic of enlarged candidate set.

Ours-CP: This is our method with context pooling as presented in Section 6.2.3. This method also incorporates the enlarged candidate set heuristic.

Ours-Full(CP+DE): This is our method with both context pooling and dual embedding, as presented in Section 6.2.3. This method also uses the enlarged candidate set heuristic.

Evaluation Metrics: For most of our results, we use accuracy to measure performance, i.e., the percentage of examples where our predicted idiom is the same as the ground truth idiom. As a second metric, we also report the performance of ranking all the Chinese idioms in the vocabulary. For this setting we use Mean Reciprocal Rank (MRR), a well-established metric for ranking problems, as the evaluation metric.

Other Settings: We use pre-trained BERT for Chinese with Whole Word Masking (WWM) [27][1] and pre-trained RoBERTa for Chinese[2]. As BERT has a limit on the input sequence length, 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.

We use 4 Nvidia 1080Ti GPU cards and a batch size of 10 per card with a total 5 training epochs. The initial learning rate is set to $5e^{-5}$ with 1000 warm-up steps. We use the optimizer *AdamW* in accordance with a learning rate scheduler *WarmupLinearSchedule*.

[1] `https://github.com/ymcui/Chinese-BERT-wwm`
[2] `https://github.com/brightmart/roberta_zh`

6.3.2 Main Results

We show the comparison of the performance of the various methods in Table 6.1. We also show the human performance. For Human, LM, AR and SAR, the performance shown in the table is taken directly from [165]. We can observe the following from the table.

Comparing **Ours-Full(CP+DE)** with other models, we can see that **Ours-Full(CP+DE)** consistently outperforms the other methods, including **Ours-CP**, for all evaluation splits in terms of both accuracy and MRR. This shows that our full model using dual embeddings coupled with context-aware pooling does make the model more expressive and captures the underlying meanings of Chinese idioms better. It is also worth noting that on the **Out** split, **Ours-Full (CP+DE)** achieves significant improvement over **Ours-CP**, showing a better generality of using dual embeddings.

Using context-aware pooling, **Ours-CP** achieves significant gain over **BL-IdmEmb** on the more challenging split **Sim** and **Out**. **Ours-CP** also shows the competency in comparison with **BL-CharSeq** that without merging the passage and a candidate answer into a single sequence, we could still achieve competitive results. It is important to note that **Ours-CP** is computationally much lighter than **BL-CharSeq** and enables us to train models by considering all idioms in the vocabulary as candidates, which is not feasible for **BL-CharSeq**.

Overall, we can see that the experiment results demonstrate that both contextual-aware pooling and dual embedding are effective, and our proposed full method generally can outperform all the other methods we consider that represent the state of the art.

		Dev		Test		Ran		Sim		Out	
		ACC	MRR	ACC	MRR	ACC	MRR	ACC	MRR	ACC	MRR
Human	[165]	-	-	87.1	-	97.6	-	82.2	-	86.2	-
LM	[165]	71.8	-	71.5	-	80.7	-	65.6	-	61.5	-
AR	[165]	72.7	-	72.4	-	82.0	-	66.2	-	62.9	-
SAR	[165]	71.7	-	71.5	-	80.0	-	64.9	-	61.7	-
BL-CharSeq		79.33	-	79.42	-	88.84	-	72.93	-	73.11	-
BL-IdmEmb (w/o EC)		73.59	0.017	73.31	0.017	81.05	0.017	68.13	0.017	63.82	0.012
BL-IdmEmb		80.24	0.433	79.76	0.429	91.87	0.429	71.93	0.429	72.17	0.332
Ours-CP		81.19	0.429	81.13	0.425	91.84	0.425	73.60	0.425	73.80	0.321
Ours-Full (CP+DE)		**82.79**	**0.450**	**82.64**	**0.446**	**93.46**	**0.446**	**75.46**	**0.446**	**76.44**	**0.349**

Table 6.1: The experiment results on ChID.

6.3.3 Evaluation on ChID-Competition

ChID-Competition[3] is the data for an online competition[4] on Chinese idiom comprehension. Different from ChID, for each entry in ChID-Competition, a list of passages is provided with the same candidate idiom set, and therefore some heuristic strategies can be used (for instance, the exclusion method). The challenge is that ground truth answers will be similar in semantic meanings, and prediction models need to focus on their differences while comparing similar contexts to make the correct predictions. Similar to ChID-Official, ChID-Competition is divided into *Train*, *Dev*, *Test* and *Out* splits for different evaluation stages.

To further test the competency of our model, we evaluate the full model **Ours-Full** on **ChID-Competition**. Considering differences between ChID-Official and ChID-Competition, we use some heuristic methods to post-process the predictions in order to globally optimize the results.

The comparison between our method and previous methods is listed in Table 6.2. In the first section of the table, we list the top-ranked competitors from the competition leaderboard. Then we show the results using several pre-trained language models found on the CLUE leaderboard.[5]. Finally, we list our own full model **Ours-Full**,

[3] https://github.com/zhengcj1/ChID-Dataset/tree/master/Competition

[4] https://biendata.com/competition/idiom/

[5] We show representative systems on the leaderboard as of the submission date of this chapter. https://github.com/CLUEbenchmark/CLUE

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
BERT-base	82.20	82.04	-
ERNIE-base	82.46	82.28	-
RoBERTa-large	85.31	84.50	-
RoBERTa-wwm-large-ext	85.81	85.37	-
Ours-Full	89.68	89.55	84.43

Table 6.2: Experiment results on ChID-Competition.

which used a larger pre-trained RoBERTa model. The experiment results show that our full model achieves competitive results compared with the top ranked systems of the competition.

6.3.4 Further Analysis Through Attribution Method

To better understand how our models achieve consistent improvement, we adopt the gradient based attribution method, Integrated Gradients (IG) [118], to visualize how each character contributes to the final prediction. Essentially, given a trained model, for each example, the IG method can assign an attribution value to each input unit (a single Chinese character in our case) that indicates how much this input unit contributes to the prediction based on this trained model. Without loss of generality, we focus on comparing three models **BL-IdmEmb**, **Ours-CP** and **Ours-Full**[6]. To make the visualization more readable, we apply Chinese word segmentation tools to merge characters into words. The attribution value of a word is the highest absolute value of all merged characters.

We show some cases in Figure 6.1, where red color represents positive correlation with the prediction and blue color represents negative correlation with the prediction. On the left, both "供不应求" (in great demand) and "大名鼎鼎" (famous) are positive idioms with a sense of "abundant", but the correct answer is "大名鼎鼎"

[6]To simplify the application of the IG method, here we do not use enlarged condidate set for these three methods.

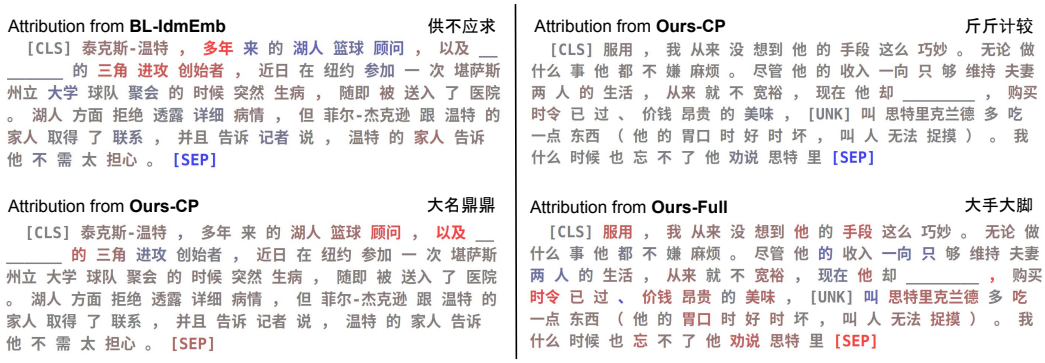


Figure 6.1: Example cases with attribution values of words shown in red and blue. Red indicates positive correlation with the prediction while blue indicates negative correlation with the prediction.

based on the global context. We hypothesize that **BL-IdmEmb** may have learned the correlation between "多年" (for many years) and "供不应求," and thus makes a wrong prediction solely based on this signal. On the other hand, **Ours-CP** chooses "大名鼎鼎" to be consistent with the "顾问" (consultant) before the conjunction word "以及" (and), suggesting that context-aware pooling may have helped with the understanding of the context.

On the right of the figure, the two candidates "斤斤计较" (mean) and "大手大脚" (over generous) are antonyms which means different attitudes to money. Both suit the context very well locally. However, the context has the adversative relation "却" (but) and the word "价钱昂贵" (expensive), showing the person is too generous with money, making "大手大脚" the correct candidate. This example shows that for more complex contextual understanding, **Ours-Full** shows advantages over **Ours-CP**.

6.4 Conclusion

In this chapter, we proposed a BERT-based dual embedding method to study Chinese idiom prediction. We used a dual-embedding to not only capture local context information but also match the whole context passage. Our experiments showed that our dual-embedding design can improve the performance of the base model,

and both the idea of context-aware pooling and the idea of dual embedding can help improve the idiom prediction performance compared to the baseline methods on the ChID dataset.

Chapter 7

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

In this chapter, we continue the study of the task 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.

7.1 Introduction

One limitation with existing studies is that the corpus used by other researchers do not have a high coverage of Chengyu. The ChID [165] dataset covers 3,848 Chengyu and the Chengyu Cloze Test (CCT) dataset [61] covers 7000 Chengyu. However, Chinese Chengyu dictionaries typically include around 20,000 Chengyu entries.

To address this problem, we collect a large corpus of Chinese text covering a

much wider range of Chengyu and propose a two-stage Chengyu recommendation model. Our model consists of a pretraining stage and a fine-tuning stage. The pretraining 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.

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 pretraining 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 pretraining can also be used for Chengyu emotion prediction and achieve decent performance.

7.2 Two-Stage Chengyu Recommendation

In this section, we present our two-stage Chengyu recommendation model. The model consists of a *pretraining stage* and a *fine-tuning stage*. The pretraining 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

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

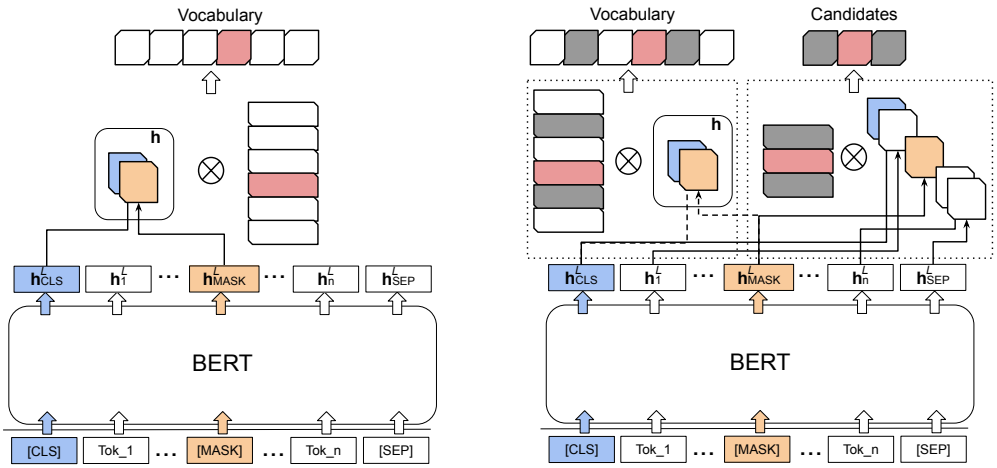


Figure 7.1: Left: The network structure used for pretraining. Right: The network structure used for fine-tuning.

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 chapter we do not adopt the multitask learning approach because of two reasons. First, the unlabeled dataset we use for pretraining 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 7.3.

7.2.1 Pretraining Stage

Our pretraining is done on top of Chinese-BERT-wwm [27], which is an improved version of the original Chinese version of BERT [31]. Chinese-BERT-wwm uses Whole Word Masking [31] in its Masked Language Model pretraining task, and is found to work better for a number of NLP tasks [110, 28, 33]. 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 $c = (w_1, w_2, \ldots, w_c, w_{c+1}, w_{c+2}, w_{c+3}, \ldots, w_n)$ 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 $\mathbf{h}^L_{\texttt{CLS}} \in \mathbb{R}^d$ denote the hidden vector produced by the last layer of BERT representing `[CLS]`, and $\mathbf{h}^L_{\texttt{MASK}} \in \mathbb{R}^d$ the similarly produced hidden vector representing `[MASK]`. Following the practice of [120, 139], We define a vector $\mathbf{h} \in \mathbb{R}^d$ as follows:

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

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:

$$p(v|c) = \frac{\exp(\mathbf{e}_v \cdot \mathbf{h})}{\sum_{v' \in \mathcal{V}} \exp(\mathbf{e}_{v'} \cdot \mathbf{h})}. \tag{7.1}$$

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:

$$L_{\mathcal{V}} = -\sum_{n=1}^{N} \log p(a^*_n|c_n). \tag{7.2}$$

The left side of Figure 7.1 illustrates the model used for pretraining.

Pretraining Data

We need a large corpus with a wide coverage of Chengyu for the pretraining 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 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 pretraining task. Following [86], we use a subsampling approach to counter the imbalance between rare and frequent Chengyu as follows:

$$P(v) = \begin{cases} 1 & c(v) \leq 10 \\ 1 - \sqrt{\frac{t}{f(v)}} & c(v) > 10 \end{cases}, \tag{7.3}$$

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.

7.2.2 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 pretraining stage. The output of the L-layer BERT is a

[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`

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 pretraining 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".

Also, because we now have a smaller candidate set, we can afford to also consider matching each candidate Chengyu with not only the representation of `[MASK]` but also its contextual words, i.e., words surrounding `[MASK]`. Specifically, we still define vector $\mathbf{h}$ in the same way as in Section 7.2.1. We further take the dot-product between the embedding of candidate a with each hidden vector $\mathbf{h}_i^L$ that is within a window of size S surrounding `[MASK]`, and then use an aggregation function f to aggregate these dot products to obtain the matching result. Then the matching between v and the context will be based on $\mathbf{e}_a \cdot \mathbf{h} + f(\mathbf{e}_a \cdot \mathbf{h}_j^L, \mathbf{e}_a \cdot \mathbf{h}_{j+1}^L, \ldots, \mathbf{e}_a \cdot \mathbf{h}_{j+S-1}^L)$, where j is the index of the first token in the context window of size S. In our experiments, we choose max-pooling as the aggregation function f and experiment with different values of S.

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

$$p(a|c) = \frac{\exp(\mathbf{e}_a \cdot \mathbf{h} + \max_{i=j}^{j+S-1}(\mathbf{e}_a \cdot \mathbf{h}_i^L))}{\sum_{a' \in \mathcal{A}} \exp(\mathbf{e}_{a'} \cdot \mathbf{h} + \max_{i=j}^{j+S-1}(\mathbf{e}_{a'} \cdot \mathbf{h}_i^L))}. \tag{7.4}$$

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:

$$L_{\mathcal{A}} = -\sum_{n=1}^{N} \log p(a_n^*|c_n). \tag{7.5}$$

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. (7.4) and the probability as computed by Eqn. (7.1), i.e., normalized over all Chengyu in $\mathcal{V}$:

$$L = L_{\mathcal{V}} + L_{\mathcal{A}}. \tag{7.6}$$

The right side of Figure 7.1 illustrates the model used for fine-tuning.

7.3 Experiments on Chengyu Recommendation

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

7.3.1 Data and Experiment Settings

We refer readers to Section 2.6 for the details of the ChID dataset. 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.

- **CCT**: Chengyu Cloze Test (CCT) [61][6] 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[7].

We use 4 Nvidia 1080Ti GPU cards and a batch size of 60 per card with a total 5 training epochs for pretraining and fine-tuning. 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. The training is done via half precision supported by apex[8].

[6] https://github.com/bazingagin/chengyu_data
[7] http://zaojv.com
[8] https://github.com/NVIDIA/apex.git

7.3.2 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. (7.6), 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** is a bidirectional LSTM language model method. **AR** is the attentive reader model [51] and **SAR** is the Stanford attentive reader model [17]. LM, AR and SAR are all methods implemented and reported in [165]. In addition, we implemented a baseline that uses Chinese-BERT-wwm-ext directly for Chengyu recommendation. We refer to this baseline as **BERT-BL**. We also show the human performance as a reference point. 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 pretraining 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. (7.6). **w/o $L_{\mathcal{A}}$:** In this version, we exclude $L_{\mathcal{A}}$ in the objective function Eqn. (7.6).

The results measured in accuracy are shown in Table 7.1. For Human, LM, AR and SAR, the performance shown in the table is taken directly from [165]. 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

Model		Dev	Test	Ran	Sim	Out
Human	[165]	-	87.1	97.6	82.2	86.2
LM	[165]	71.8	71.5	80.7	65.6	61.5
AR	[165]	72.7	72.4	82.0	66.2	62.9
SAR	[165]	71.7	71.5	80.0	64.9	61.7
BERT-BL		79.33	79.42	88.84	72.93	73.11
Two-Stage		**85.61**	**86.23**	**95.41**	79.37	**83.36**
w/o pretraining		80.00	80.01	89.40	73.80	72.22
w/o fine-tuning		80.14	80.54	92.10	72.69	78.76
w/o $L_{\mathcal{A}}$		84.70	84.87	95.22	77.40	81.81
w/o $L_{\mathcal{V}}$		85.49	85.66	93.45	**79.57**	82.47

Table 7.1: The experiment results in terms of accuracy on ChID-Official.

and the usefulness of our collected unlabeled Chinese corpus for pretraining. (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 the difference is not substantial, suggesting that it may not be critical whether we use **w/o** $L_{\mathcal{V}}$, **w/o** $L_{\mathcal{A}}$ or both. We do observe that in most cases, $L_{\mathcal{A}}$ is slight more important. For the split **Sim**, which uses near-synonyms as candidate answers, using $L_{\mathcal{A}}$ only performs the best of all. But for the test set **Ran**, which uses randomly selected wrong candidate answers, using $L_{\mathcal{V}}$ only performs better than using $L_{\mathcal{A}}$ only. 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.

7.3.3 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-

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 7.2: Experiment results for ChID-Competition. Here we include the top submissions on the leaderboard.

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 7.2 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.[9] In the second part of the table, we list several other pretrained language models extracted from the benchmark CLUE [151][10]. 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.

[9] We show the top-3 systems on the leaderboard as of the submission date of this chapter.

[10] `https://github.com/CLUEbenchmark/CLUE`

7.3.4 Results on CCT

We further use the CCT [61] 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 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). Table 7.3 shows the results. We can see from the table that our two-stage model again can outperform the baseline performance reported in [61].

Model	Candidates	Performance
Human [61]	4	70.0
BiLSTM [61]	4	89.5
Pretraining	4	**93.7**
Pretraining	7	90.5

Table 7.3: Evaluation on CCT.

7.3.5 Further Analysis

The Effect of Context Window Size: To check whether the size of the contextual window matters, we set the window size to 1, 3, 10, 20 and 30. We show the experiment results below in Table 7.4.

Model	Dev	Test	Ran	Sim	Out
Two-Stage (Window Size 1)	85.61	85.93	95.36	78.90	83.15
Two-Stage (Window Size 3)	85.58	86.03	95.38	79.07	83.11
Two-Stage (Window Size 10)	85.69	85.97	95.45	79.15	83.22
Two-Stage (Window Size 20)	85.62	85.99	95.43	79.12	83.19
Two-Stage (Window Size 30)	85.63	86.03	95.41	79.09	83.28

Table 7.4: The experiment results for window size in terms of accuracy on ChID-Official.

From the table, we observe that when we vary the window size, the performance did not change substantially. This suggests that during the fine-tuning stage, matching

the embedding of an answer candidate with $\mathbf{h}$ only but not with the contextual words can already work well. We suspect that this is because the vector $\mathbf{h}$ includes both the representation of `[MASK]` and of `[CLS]`, and therefore already contains some contextual information.

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

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 7.5.

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

Violation of Syntactic Rules

Chinese idioms also need to follow syntactic rules. Given a particular context, some candidate idioms are not suitable simply because they do not syntactically fit into the context. For example, the two candidates in row *Syntactic Error* in Table 7.5 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 idioms may both be suitable for the blank locally, once taking the full context into account, some idioms 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 7.5, 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 idioms 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 7.5, "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.

Synonyms and Non-Synonyms

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”, 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 7.5.

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 idioms that require reasoning to understand.

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

Table 7.6: Examples of sentiment labels for some Chengyu in CALO.

7.4 Chengyu Embeddings for Emotion Prediction

We suspect that the Chengyu embedding vectors learned by our pretraining stage may be valuable for other tasks. To test this hypothesis, we choose a Chengyu emotion prediction task.

Specially, emotion prediction of Chengyu [135] attempts to use lexicons from the CIKB database as a source to build a feature-based SVM to predict the emotion label for a Chengyu. Since CIKB is not available online, we use Chinese Affective Lexicon Ontology (CALO) [157] as a substitution of annotated source.

CALO is created with the purpose of supporting textual Affective Computing (AC) in Chinese language. The construction of CALO is based on mainstream emotional classification research [34] and also combines conventional Chinese emotion categories. Six categories, anger (怒), fear (惧), sadness (哀), enjoyment (乐), disgust (恶), surprise (惊) are consistent with [34]. However, enjoyment (乐) is not sufficient to describe some positive emotions like respect and belief, so an extra category, "good" (好) is added. There are therefore 7 main categories. Each main category is further classified into different numbers of subcategories according to their intensity and complexity. There are 21 subcategories in total.

Each entry in CALO has a sentiment label from the subcategories. We take those Chengyu from our trained Chengyu embeddings which have entries in CALO. This gives us 14,361 Chengyu, a comparable size with that of Wang and Yu [135].

We use the Chengyu embeddings learned from our pretraining to predict the Chengyu emotions. Since CALO has no contexts, we treat each Chengyu as a

	Coarse-grained		Fine-grained		Polarity	
	ACC	F1	ACC	F1	ACC	F1
BERT-BL	71.68	59.36	59.17	40.86	71.90	49.57
w/o embeddings	73.31	61.68	60.95	41.72	73.08	**52.21**
w/ embeddings	**73.52**	**62.11**	**61.27**	**42.75**	**73.39**	51.97

Table 7.7: The emotion prediction results on CALO.

"sentence". The baseline method we compare with uses Chinese-BERT-wwm-ext and adds a classification layer over the hidden vectors of `[CLS]`. For our method, there are two ways to train an emotion prediction model using our pretraining model. The first one is done in the same way as the baseline. The second one uses a classifier over the contextualized representation of the `[CLS]` token and the embeddings of Chengyu. Specifically, **w/o embedding** means we treat the input Chengyu as a sequence of characters and then use our pre-trained Chinese BERT model to process this sequence to obtain the representation of the `[CLS]` token. This token is used as input for emotion detection. **w/ embedding** means we make use of our pre-trained Chengyu embedding and concatenate the Chengyu embedding with the hidden representation of `[CLS]`, and use the concatenated vector as input for emotion classification.

We randomly split the Chengyu from CALO into training and testing sets by keeping the testing set size to 3000. We try to predict the sentiment of the Chengyu in terms of both coarse-grained and fine-grained categories as well as their polarities. We choose ten random splits to train the model and report the average scores as shown in Table 7.7. 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.

7.5 Conclusions and Future Work

In this chapter, 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.

Chapter 8

Chengyu-oriented Text Polishing for Chinese: Datasets and Baselines

This chapter presents the task of text polishing, which generates a sentence that is more graceful than the input sentence while keeping its semantic meaning. Text polishing has great values in real use and is important for modern writing assistance systems. For example, users of text-polishing models can select one segment of a sentence and get a refined version of the segment. This is useful for both ordinary writers and students who are learning Chinese as a second language. These models also have potential applications in post-editing of machine translated sentences to make them more native and intriguing. However, the task is still not well studied in the literature. There is a lacking of formal task definitions, benchmark datasets, and powerful models. In this work, we formulate the task as a context-dependent text generation problem and conduct a case study on the Chinese language. Specifically, we adopt the hypothesis that using Chengyu properly in Chinese language presents higher fluency and elegancy in the mastering of the language. We construct a Chengyu-oriented dataset for text polishing. The dataset contains 1.5 million automatically generated instances for training and four thousand human-annotated examples for evaluation. On top of the Transformer structure, we build many baseline systems using different configurations and initialization strategies. Automatic evaluation in terms of BLEU

indicates that the T5-style pretrained model obtains about 7.0 absolute gains. The results from human evaluation further reveal the polishing ability of the system.

8.1 Introduction

Intelligent writing assistance can accelerate the writing process for humans and has made remarkable progress in recent years. One example is Grammatical Error Correction (GEC) which aims to automatically detect and correct grammatical errors in written sentences. With incorporation of BERT [31], performance of GEC models has been largely improved. There are also intelligent writing systems like Grammarly which has been used by millions of users [91]. Another example is text completion. Language models like GPT [101] show promising results in generating coherent texts given prompts.

In this work, we study text polishing, an important component of modern writing assistance yet rarely studied in the literature. Given a sentence as the input, the task is to generate a sentence that is more graceful than the input while keeping the semantic meaning unchanged. We study the problem in a context-dependent configuration which we believe is closer to the real scenario. An example is given in Table 8.1. Given a sentence "阿宝惊呆了" ("*Bao is stunned*", underlined in Table 8.1) and its surrounding context as the input, the target is to polish the input sentence into "阿宝听得瞠目结舌" ("*Bao's jaw dropped as he listened"*) while keeping the context unchanged. By properly using the Chinese Chengyu "瞠目结舌" ("*somebody's jaw dropped*"), the polished sentence vividly depicts a person's surprised expression.

Text polishing is different from other text rewriting tasks, including paraphrasing, text infilling, and grammatical error correction. Paraphrasing only requires the output to be semantically equivalent to the input, while text polishing further requires the generated text to be more elegant. Text infilling aims to fill the missing portions of a sentence or a paragraph based on the surrounding context, with no objective in elegancy. GEC maps bad sentences into good ones, while text polishing aims to

Original Paragraph	其实科技发达到一定的程度，我们外形的转变是非常的容易的。阿宝惊呆了。他在童话世界看到的一切在这里成为了现实。 *In fact, when technology develops to a certain extent, our appearance can be easily changed. Bao is stunned. Everything he sees in the fairy tale world becomes reality here.*
Polished Paragraph	其实科技发达到一定的程度，我们外形的转变是非常的容易的。阿宝听得瞠目结舌。他在童话世界看到的一切在这里成为了现实。 *In fact, when technology develops to a certain extent, our appearance can be easily changed. Bao's jaw dropped as he listened. Everything he sees in the fairy tale world becomes reality here.*

Table 8.1: An example of text polishing for Chinese. The underlined text is the sentence that needs to be polished. The surrounding contexts keep unchanged.

convert good sentences to great ones.

As a case study, we study text polishing for the Chinese language and construct a dataset to foster research on this area[1]. The data construction pipeline includes elegant expression collection, back-translation supported by machine translation systems, and data filtering. We finally build a dataset including 1.5 million automatically generated examples for model training and four thousand human-labeled examples for model evaluation.

We build Transformer-based sequence-to-sequence baselines with different configurations. Our major findings are as follows. First, handling the problem in a text infilling manner (i.e., regarding the input sentence as a blank) performs worse than the standard-setting where both the input and context are considered. Second, pretraining in a T5 [102] manner brings significant improvements in terms of BLEU score. However, pretraining through replacing $\langle mask \rangle$ tokens with synonyms does not bring improvements. Lastly, the human evaluation indicates that the model can produce semantically related and more elegant sentences.

In summary, the main contributions of this chapter include the following:

- We present the new task of text polishing. The task aims to rewrite sentences

[1]The pipeline is language-agnostic and we plan to expand the research to other languages in the future.

to be more elegantly while maintaining the original semantics.

- We develop a semi-automatic data annotation pipeline and construct the dataset for training and evaluating text polishing systems for Chinese.
- We develop baseline systems and implement different T5-style pretrain models. Our results shed light on future directions.

8.2 Problem Formulation

The context-dependent text polishing task can be defined as follows. Given a tuple of input texts $\{C_{prev}, S_{orig}, C_{next}\}$. Text polishing aims to polish the S_{orig} sentence considering the context C_{prev} and C_{next}, which form the polished text $\{C_{prev}, S_{polish}, C_{next}\}$. The polished text retains the primary meaning and is more elegant than the original text.

8.3 Dataset Construction

Text polishing is to make a piece of readable text with better expression. However, it's very challenging to define a better expression. Better expressions may include appropriate use of rhetorical methods, sentences with variable forms, or aphorisms and idiomatic expressions. In this work, we consider text polishing from the perspective of word usage. Given the prevalence usage of Chengyu in the Chinese language, Chengyu usage has been a good sign of better expression and is generally considered to be effective in enhancing elegancy in writing [78, 80].

In this section, we introduce the process of constructing the text polish dataset for Chinese. We first carefully collect a corpus containing sentences with elegant expression S_{polish}. Then we use back-translation to translate S_{polish} into English then translate back into Chinese again to get S_{orig}. We pair S_{orig} and S_{polish} with context as one sample of text polishing dataset. Finally, We perform human annotation to

evaluate the quality of automatically constructed pairs and keep the eligible ones as the test sets.

8.3.1 Chengyu Collection

Although Chengyu usually consist of only four characters, Chengyu can reveal complex meaning and enhance the conciseness and elegance of writing if properly used. For example, in the text segment "在这个大城市里找一个人无异于大海捞针。" (*Searching for one man in such a big city is like looking for a needle in a haystack.*), the Chengyu "大海捞针" (*look for a needle in a haystack*) elegantly describes the difficulty of finding a thing that is almost impossible to find, which is more vivid than a common expression like "在这个大城市里找一个人非常困难。" (*Searching for one man in such a big city is very difficult.*). Therefore, we can take the sentences including Chengyu as elegant expressions. We use an Chengyu list in ChengyuBERT [123], which collects Chengyu from the web and contains 33,237 common Chengyu.

8.3.2 Corpus Preparation

We use two corpora that contain Chengyu as our sources. The first one comes from ChengyuCorpus[2] used in ChengyuBERT [123]. The corpus contains about 10 million Chinese essays from e-books crawled online and has a wide coverage of Chengyu. Each essay has three sentences and the middle sentence contains an idiom. To expand the domain of the dataset, we also use the United Nations Parallel Corpus v1.0 (UN6Ways) [168]. This corpus is composed of official records and other parliamentary documents of the United Nations that are in the public domain. These documents are mostly available in the six official languages of the United Nations and organized in sentence-level alignments. We extract the Chinese sentences containing Chengyu from the corpus as S_{polish} and retrieve the previous sentence C_{prev} and next

[2]https://github.com/VisualJoyce/ChengyuBERT

sentence C_{next} of the S_{polish} by using their index in the corpus.

Data Selection Considering the difficulty of text polishing, we only keep essays whose extracted sentence contains an Chengyu and has a length between 7 and 30. Since most Chengyu have four characters, this range of length secures that phrases and sentences contain an idiom. To avoid unbalanced data and promote the diversity of Chengyu, we limit the maximum number of instances for each Chengyu to 500.

Back-translation Machine translation models are trained on large-scale data among which Chengyu only account for a small percentage. So the translation model tends to generate an ordinary expression that is less elegant. By translating the Chinese sentence containing Chengyu to English then translating the generated English sentences back to Chinese again, we can get the sentences with the same meaning but less elegance. Note that since United Nations Corpus has human-translated English parallel sentences, we only need translate its English counterpart to Chinese in one stage.

We use the translation services provided by TranSmart[3] to translate the extracted sentence S_{polish} containing Chengyu to get the S_{orig}. We observe the results of translation and find some translated sentences still contain Chengyu. For these sentences, one can hardly tell whether the expression before or after translation is more elegant. Therefore, we discard the samples whose translated sentences contain Chengyu by searching in the Chengyu list.

Data Filtering We notice that the meanings of some translated sentences are inconsistent with their original sentences and the problem gets more serious for sentences containing Chengyu whose literal meanings are inconsistent with the idiomatic meaning [55, 111]. From our observation, the usage of Chengyu in our data follows a long-tailed distribution, and Chengyu at the end of the long tail are rarely seen in the training corpus of the translation models. Therefore, we discard

[3]https://transmart.qq.com

Chengyu with a frequency below 100. Upon filtering, we get 1,532,867 paragraph pairs from ChengyuCorpus and 3,621 from UN6Ways.

8.3.3 Human Judgement

To verify our hypothesis that machine translation can get S_{orig} with the same meaning as S_{polish} but with degraded elegance, we employ human annotators to evaluate the quality of the paragraph pairs from ChengyuCorpus and UN6Ways.

Context	Extracted Sentence	Translated Sentence	Is the meaning retained	Is the elegance degraded
其实科技发达到一定的程度，我们外形的转变是非常的容易的。#polish#。他在童话世界看到的一切在这里成为了现实。	阿宝听得瞠目结舌	阿宝惊呆了	Yes	Yes
In fact, when technology develops to a certain extent, our appearance can be easily changed. #polishing#. Everything he sees in the fairy tale world becomes reality here.	Bao's jaw dropped as he listened	Bao is stunned		
她抱着猫关了店门上了楼。#polish#，唯独她这儿冷冷清清，却毫不在乎。	别的店铺门庭若市	还有很多其他的店	No	Yes
She closed the store door and carried the cat upstairs. #polish#, only she is deserted here, but she doesn't care.	Other stores are crowded	There are many other stores		

Table 8.2: The questionnaire for human annotation with two examples from the annotated datasets. The last two columns are the questions to be annotated. Chengyu are highlighted with bold font in extracted sentences.

The questionnaire used for the annotation is listed in Table 8.2. Each segment of the context to be polished is replaced by the special token "#polish#" and the segment will be listed as "Extracted Sentences", denoted as S_{polish}. Accordingly, S_{orig} is translated from S_{polish} and listed as "Translated sentence". We ask the annotators to answer two questions: (1) whether the translated sentence has the same meaning with extracted sentence, (2) whether the elegance of translated sentence is lower than that of the extracted sentence.

In total, we sample 5,000 pairs of ChengyuCorpus and use all pairs (3,621) of UN6Ways for the construction of the questionnaire. The result of annotation is shown in Table 8.3. There are 2,211 pairs of ChengyuCorpus and 1,846 pairs of UN6Ways receive the answer "Yes" to both questions. They are eligible pairs that

Corpus	ChengyuCorpus	UN6Ways
Num. of annotated	5000	3621
Num. of eligible	2211	1846
Prop. of eligible	44.2%	51.0%
Av. length of extracted	13.2	23.6
Av. length of translated	10.5	22.8

Table 8.3: Statistic of the annotated dataset.

satisfy our need, namely S_{polish} has the same meaning with S_{orig} but express more elegantly. The proportion of eligible pairs for ChengyuCorpus and UN6Ways are 44.2% and 51.0%, respectively. UN6Ways has about 6.8% higher proportion than ChengyuCorpus. [4]

The results show that our method of constructing a dataset for text polishing is reasonable. The statistics of the text polishing dataset are shown in Table 8.4. The dataset will be made publicly available to the community.

	Train Set	Dev Set	Test Set	Av. len of S_{polish}
P-Book	1,507,867	20,000	2,211	13.1
P-MultiUN	–	–	1,846	23.5

Table 8.4: Statistics of the dataset for text polishing task.

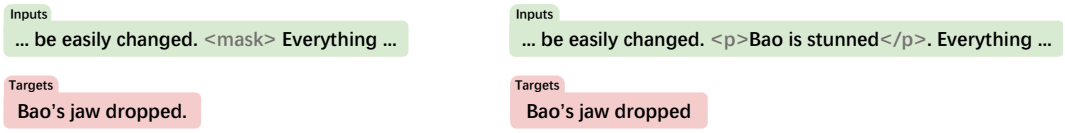
8.4 Models

8.4.1 Infilling Objective and Paraphrasing Objective

There are two choices for the objectives of sequence-to-sequence model: the infilling objective and the paraphrasing objective.

Infilling Objective The infilling objective is the same with that of T5 [102], shown in Figure 8.1(a). We naturally choose T5 as the pretrained model for the text

[4]This may be the former only need to translate parallel English sentences to Chinese while the latter need to translate Chinese to English then back to Chinese in two stages.



(a) Infilling objective (b) Paraphrase objective

Figure 8.1: Schematics of two Seq2Seq task objectives. We take English as an example to illustrate, the same is for Chinese.

polishing task. As shown in Figure 8.1(a), we mask a text span that needs polishing with $\langle mask \rangle$ token. The model predicts the masked text span with the polished sentence based on the context.

Paraphrasing Objective For paraphrasing objective in Figure 8.1(b), we use the same model architecture as T5, but use the input style like MacBERT [29] to alleviate the discrepancy between pretraining and finetuning stage. Specifically, we first tokenize sentences into words, then randomly select some words and substitute each word with a synonym. A similar word is obtained by using the Synonyms toolkit[5]. As shown in Figure 8.1(b), we use $\langle p \rangle$ and $\langle /p \rangle$ to enclose the sentence that needs to be reconstructed.

8.4.2 Pretraining

Considering the superiority of pretrained models on many tasks, we also construct our baseline models for text polishing over pretrained models. We use different span lengths for the infilled text when pretraining T5. We explore the following pretraining configurations for Chinese T5, **SPAN** stand for span length and **PP** for paraphrase:

- **T5 (SPAN=3 char, PP=False)**: This is the same as the original T5 for English. We use a fixed span length of 3 Chinese characters.
- **T5 (SPAN=1-6 char, PP=False)**: We use variable span length for pretraining to enhance the model's generalization. The span length is sampled from 1-6

[5]https://github.com/chatopera/Synonyms

with the probabilities of [0.05, 0.15, 0.3, 0.3, 0.15, 0.05]. The random span-corruption teaches the model to predict how many tokens are missing from a span.

- **T5 (SPAN=subsent, PP=False)**: We randomly choose sub-sentences in input sequences as corrupted span. The sub-sentences are any text segmentation split by "。", "；", "，", "？", or "！". sub-sentence span teaches the model to predict longer text with relative complete semantics than randomly chosen tokens.

- **T5 (SPAN=1 word, PP=True)**: Following MacBERT [29], we randomly select words and replace them with synonyms. By this means, we hope to lower the discrepancy between pretraining and finetuning objectives.

8.4.3 Finetuning for Text Polishing

We treat the text polishing task as a sequence-to-sequence finetuning problem over pretrained language models. For each pretrained models from Section 8.4.2, we may use either the infilling objective or the paraphrasing objective to do the finetuning. The polished sentence is the target sequence conditioned over the context and the original sentence.

8.5 Experiments

8.5.1 Experimental Settings

Pretraining Details We use the same model architecture as the T5-base model, which has 220M parameters, 12 Transformer layers, 12 attention heads, 768 hidden sizes. Our implementation is based on Hugging Face Transformers [146]. The original T5 is pretrained on massive English data crawled from the web that cannot be used for Chinese directly. So we pretrain T5 on Chinese data from scratch ourselves.

We collect Chinese news and blog articles from the web as the pretraining data, which has about 800 Gigabytes after data processing. We use a maximum sequence length of 512 and a batch size of 4096 sequences. We pack multiple sentences into each entry of the batch as much as possible to promote GPU utilization. We train different variants of the T5 model for 500k updates on 32 Tesla V100 GPUs. For optimization, we use the Adam optimizer [66] ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e^{-6}$) with a weight decay of 0.01. The learning rate is warmed up over the first one-tenth steps to a peak value of $1e^{-4}$, and then linearly decayed.

Finetuning Details We finetune the text polishing task on the constructed dataset **P-Book**. We finetune each pretrained model by using 8 Tesla V100 GPUs for 100k steps with a maximum length of 512 and a batch size of 256. We use the Adam as the optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e^{-8}$. The initial learning rate for Adam is $1e^{-4}$ with 10k warmup steps during which it increases linearly from 0, following a linear decrease to 0. The beam size is 4 for generation decoding and the maximum length of the sequence to be generated is set to 50. We use BLEU [92] as the metric to evaluate the performance of models on the text polishing task. BLEU is a popular automatic metric for machine translation. It uses a modified form of *precision* measure to compare a candidate translation against multiple reference translations. We save a checkpoint every 2,000 steps and report results on the model checkpoint corresponding to the highest validation performance.

Vocabulary The original T5 model employs a SentencePiece [92] tokenizer, which will cause two issues if it is adopted directly to Chinese. Firstly, it will add numerous superfluous white space tokens "_" to tokenized sequences, lengthening them significantly. Secondly, the SentencePiece tokenizer will change certain full-width symbols to half-width symbols, which will impair the downstream task's evaluation. So We use BERT's WordPiece tokenizer to tackle these issues. To be consistent with T5 in the definition of special tokens, we extend the original Chinese BERT [31]'s

vocabulary with one eos_token $\langle \backslash s \rangle$ and 100 sentinel tokens $\langle extra\_id\_num \rangle$, which change the length of BERT vocabulary from 21128 to 21229.

8.5.2 Comparison of Finetuning Objectives

Finetuning Objectives	P-Book	P-MultiUN
Infilling	11.26	1.82
Paraphrasing	**48.06**	**14.93**

Table 8.5: The automatic evaluation results on two test sets using infilling objective and paraphrasing objective to finetune text polishing. The pretrained model is **T5 (SPAN=1-6 char, PP=False)**.

We compare the two objectives (Figure 8.1) for the text polishing task via finetuning on the pretrained model **T5 (SPAN=1-6 char, PP=False)**. The results are shown in Table 8.5. The model using paraphrasing objective outperforms that using infilling objective with a large gap on both test sets. We find that the generated text of the model using the infilling objective is not able to retrain the meaning of the original text. Therefore, the model finetuned with the infilling objective gets lower scores in terms of BLEU which is an overlap-based metric. The result implies that the paraphrasing objective is a better choice for the text polishing task. So we use the paraphrasing objectives in the following experiments.

8.5.3 Automatic Evaluation Results

The automatic evaluation results on the text polishing test set with all T5 variants are shown in Table 8.6.

1. Considering the similarity between the pretraining objectives of T5 and the text polishing task, we list the zero-shot performance of T5 for the text polishing task in the first group of Table 8.6. All BLEU scores are nearly zero, indicating that pretrained models cannot be applied to the text polishing task directly.

	Models	P-Book	P-MultiUN
zero-shot	T5 (SPAN=3 char, PP=False)	0.03	0.00
	T5 (SPAN=1-6 char, PP=False)	1.28	0.14
	T5 (SPAN=subsent, PP=False)	**1.76**	**3.27**
	T5 (SPAN=1 word, PP=True)	0.11	0.00
supervised	T5 (Randomly initialized weights)	41.36	10.78
	T5 (SPAN=3 char, PP=False)	47.59	15.97
	T5 (SPAN=1-6 char, PP=False)	**48.06**	14.93
	T5 (SPAN=subsent, PP=False)	47.93	**20.85**
	T5 (SPAN=1 word, PP=True)	47.37	19.71

Table 8.6: The automatic evaluation results on the test sets with all T5 variants. We use bold to mark the best results in each group.

2. The second group of Table 8.6 is the supervised results of models finetuned on **P-Book** training data.

   (a) All baseline models achieve much higher performance on **P-Book** than on **P-MultiUN**. There is about a 30 point difference between the two data sets. This means that text polishing models cannot be generalized easily across different domains.

   (b) On **P-Book** test set, all baseline models achieve the comparable scores. There is only 0.69 points difference between the lowest and the highest scores. But on the **P-MultiUN** test set, the difference between the lowest and highest scores is 5.92.

   (c) The **T5 (SPAN=subsent, PP=False)** and **T5 (SPAN=1 word, PP=True)** perform much better than **T5 (SPAN=3 char, PP=False)** and **T5 (SPAN=1-6 char, PP=False)**. We conjecture that the former two models introduce the external knowledge, their span is a sentence for **T5 (SPAN=subsent, PP=False)** or word for **T5 (SPAN=1 word, PP=True)**, while the latter two models only span on arbitrary characters without semantics. This kind of knowledge improves the model's generalization on out-of-domain data.

8.5.4 Effect of Text Length for Text Polishing

Models	P-Book		
	All	$\|S_{polish}\| \leq 9$	$\|S_{polish}\| > 9$
T5 (SPAN=3 char, PP=False)	47.59	55.01	46.03
T5 (SPAN=1-6 char, PP=False)	**48.06**	55.19	**46.55**
T5 (SPAN=subsent, PP=False)	47.93	55.01	46.34
T5 (SPAN=1 word, PP=True)	47.37	**55.31**	45.75

Table 8.7: The automatic evaluation results on two test sets constructed by dividing **P-Book** test set according the length of S_{polish}. 'All' column is the original **P-Book** test set given as the reference.

To better compare the ability of the model in text polishing with different lengths, we further divide the **P-Book** test set into two parts according to the length of S_{polish}. We inspect the **P-Book** dataset and find that most phrases containing one Chengyu are not greater than 9 in length. For example, the length of the phrase "争风吃醋的人们" (jealous people) is 7. Therefore, we divide the test set of **P-Book** (2,211) into two parts according the length of S_{polish}, namely, $|S_{polish}| \leq 9$ (642) and $|S_{polish}| > 9$ (1,569).

We run our baseline models on the two separated test sets and the results are shown in Table 8.7. The text polishing models for shorter texts outperform those for longer texts by about 8-10 points, which is obvious that baseline models are good at polishing shorter text.

8.5.5 Human Evaluation Results

Text polishing aims to polish a sentence to get the elegant expression retaining the original meanings. The BLEU is not enough to evaluate the consistency of meanings and elegance of expression. Therefore we perform human evaluation from these aspects. We randomly choose 100 samples from **P-Book** and **P-MultiUN**, respectively. There are 50 samples in **P-Book** with $|S_{polish}| \leq 9$, the other half with $|S_{polish}| > 9$. We use **T5 (SPAN=1-6 char, PP=False)** model to generate

	Model Input	Model Output (S_{polish})	Ground Truth
P-Book	好久以后，不少人仍忘不了十月第一周的那两天。<polish>我还记得过去</polish>，回忆令人辛酸。 *After a long time, many people still cannot forget those two days in the first week of October. <polish>I remember the past</polish>, the memories are poignant.*	往事至今仍**历历在目** *The past is still vivid in my mind*	往事还**历历在目** *The past is still vivid*
	只听见外面有粗沙嗓子说话的声音。接着，<polish>我看到栅栏早就不见了</polish>，屋穴的出口四敞大开。手枪，还在我手里。 *The only thing I could hear was a gruff voice talking outside. Then, <polish>I saw that the fence was long gone</polish>, and the exits of the cave were wide open. The pistol is still in my hand.*	我看见栅栏早已**不翼而飞** *I saw that the fence had long since disappeared*	我看见那道防栅早已**不翼而飞** *I saw that the fence had long since disappeared*
P-MultiUN	裁军是一个非常崇高的目标，不应该放弃。<polish>我们必须以精力、奉献和毅力继续前进。</polish> 我们不能放弃希望，因为那如同放弃对人类未来的希望。 *Disarmament is a very noble goal that should not be abandoned. <polish>We must keep going with energy, dedication and perseverance. </polish> We cannot give up hope because that would be like giving up hope for the future of humanity.*	我们不能因此就放弃希望而**停滞不前** *We should not give up hope and stagnate because of this*	我们必须**勇往直前、一心一意、坚韧不拔**。 *We must be courageous, single-minded and resilient.*
	部落和宗族至今仍在巴基斯坦社会、尤其是农村地区发挥一定的作用。<polish>世仇在他们中间并不少见。</polish> 有时宗族之间借通婚达成妥协以保和平。 *Tribes and clans still play a role in Pakistani society, especially in rural areas. <polish> Feuds are not uncommon among them. </polish> Sometimes clans use intermarriage to reach a compromise to keep peace.*	世仇之争在他们中间**屡见不鲜** *Feuds are common among them*	其中，累世宿仇**司空见惯**。 *Among them, the accumulated feuds are common.*

Figure 8.2: The cases of text polishing on **P-Book** and **P-MultiUN**. **P-Book** cases are generated by **T5 (SPAN=1-6 char, PP=False)** model and **P-MultiUN** cases are generated by **T5 (SPAN=subsent, PP=False)** model. The polished text in model input is surrounded by $\langle polish \rangle$ and $\langle /polish \rangle$. The elegant expressions in model output and ground truth are bold.

polished sentences for **P-Book** samples and **T5 (SPAN=subsent, PP=False)** model for **P-MultiUN** samples. We ask three annotators to judge from **consistency** and **elegance**, namely, whether the meanings of generated sentences are consistent with the original sentence and whether its expression is more elegant than the original sentences. Each case receives three labels from three annotators. The final decision is made by a majority vote. Then we count the numbers of samples meeting conditions and calculate their percentage out of the total number.

The results are shown in Table 8.8. In column elegance, almost all the elegance of generated sentences are improved than original sentences. The last column is the percentage of samples that are polished, namely, the generated sentence has the same meanings as the original one and its expression is more elegant. Human evaluation also shows that model performer better on shorter text ($|S_{polish}| \leq 9$) than longer text ($|S_{polish}| > 9$). This is consistent with the automatic evaluation results in Table 8.7. But different from the big gap of models' performance between **P-Book** and **P-MultiUN**, they are very close in human evaluation (0.69 vs. 0.66). This may be that BLEU is a precision metric and not enough to reflect the semantics and elegance of sentences. We leave finding a better evaluation method for future research.

		Consistency	Elegance	Polishing
P-Book	All	0.73	0.96	0.69
	$\|S_{polish}\| \leq 9$	0.80	0.96	0.76
	$\|S_{polish}\| > 9$	0.66	0.96	0.62
P-MultiUN		0.71	0.94	0.66

Table 8.8: The human evaluation results on two test sets. The values represent the percentage of eligible cases out of the total cases.

8.5.6 Case Study

To better understand the baseline models' ability for text polish task, we give some cases from model generated sentences in human evaluation in Figure 8.2 and ground truths as the reference. The generated sentences of 4 cases all contain one Chengyu (bold in text), which is usually considered a more elegant expression than ordinary sentences. For **P-Book**, the generated sentence has the same Chengyu with ground truth. But for **P-MultiUN**, although the Chengyu in the generated sentence and ground truth are different, they have the same meanings. For example, in the last case, "屡见不鲜" and "司空见惯" both mean "commonplace". Models can properly generate polished sentence that is suitable for the given context. This also explains why the result in automatic and human evaluation for **P-MultiUN** is not consistent.

8.6 Conclusion

In this work, we present a new task of Chengyu-oriented text polishing in Chinese language. The task aims to polish the text to enhance its elegance with the original meaning reserved. We elaborate process of data construction for text polishing, which can be used for the study of text polishing for other languages. We build baselines with different T5-style pretraining models and evaluate them through automatic evaluation and human evaluation. The results can serve as baselines for future studies.

Chapter 9

Conclusions and Future Work

In this thesis, we conduct a lines of work to understand Chinese idioms using neural network models. While we admit that Chinese language may present uniqueness in many aspects, we always keep in mind that the ubiquitous existence of idiomaticity across all langauges. That's why this thesis spent a large portion of pages on idiomaticity of other languages. We adopt probing-based approaches to not only scrutinize if pretrained language models can detect idiomaticity in the context, but also tried to analyze, from the perspective of transformer layers, the trade-off between contextualization and shifted meaning. The focus of this thesis is Chinese idioms, especially Chengyu. We address current issues of low coverage of Chengyu when learning word representations and propose new evaluation metrics of the learned Chengyu representations. Speaking of applications, our studies are built over latest large-scale pretrained language models and we spend much effort to scale the problem with a large corpus and focused pretraining. Specifically, we explored how to make use of idiomaticity of Chengyu to learn separate embeddings with respect to local context and global context. Then we consider Chengyu-oriented pretraining. Results on open benchmarks indicate that our method is effective. In addition, we find potential applications of Chengyu in intelligent writing assistance systems and propose the new task Chengyu-oriented text polishing. Our explorations over Chengyu prove that to enable a generalized pretrained language models with

idiom-aware abilities, both corpus and modelling are critical for the performance gain.

This thesis has certainly not comprehensively studied all aspects of Chinese idiom understanding. Below I will point out two potential future directions related to Chinese idiom understanding that I think are important to study. The first one is motivated by the observation that a large portion of Chengyu carry sentiments, e.g., "欢天喜地" has a positive sentiment while "哀痛欲绝" has a negative sentiment. Therefore, it is reasonable to believe that the choice of a Chinese idiom in a certain context is also affected by the sentiment polarity of the context and of the candidate idiom. The second direction is to explain how Chinese idiom recommendation models work, which may help explain those failed cases and suggest improvement to the model. This is closely related to explainability of neural network models.

9.1 Sentiment Analysis with Idioms

There are two sub-tasks that can be done considering sentiment of Chengyu. One is idiom-aware sentiment classification (ISC) of sentences. The other is sentiment-aware idiom recommendation (SIR). In Chapter 7, we explored emotion prediction of Chengyu without context. The task shows that with better pretraining, Chinese idiom understanding models can capture the emotions of Chengyu at multi-granularity. It remains a research question how idioms and context can affect each other from the point view of sentiment.

Idiom-aware Sentiment Classification (ISC): Given a piece of text that contains a Chengyu, we are required to predict the sentiment label for the text.

The challenge is that there's no public datasets focusing on idiom-aware sentiment classification. However, researchers have created plenty of datasets for general sentiment classification [124, 158, 162]. Because of the wide usage of Chinese idioms, we can construct an idiom-aware sentiment classification dataset by filtering those entries from original dataset. Using this constructed dataset, we can try to

Sentiment: 喜悦 **Sentence:** :凌晨的比赛很干净，一张黄牌都没有。昨晚裁判没捣乱。西班牙得冠军该是众望所归。
Sentiment: 厌恶 **Sentence:** : 遇到如此烂片，不来评个分，我觉得对不起我看片所花的时间和金钱，只怪最低是一星，请忽略星级，这片给负分是众望所归！！！
Sentiment: 厌恶 **Sentence:** 懂的！孩子们辛苦了！国庆的第一天，人家都欢天喜地的去外面趴趴走、跟家人团聚，姐我就只能坐在这个让我每天都情绪亢奋的牢笼里对着这电脑浏览着那些已经灰了的头像，也时不时的抬头看看窗外，感叹下外面的世界真美好，而我却如此下场，好悲惨.....

Table 9.1: Examples for sentiment classification containing Chengyu.

improve the performance of sentiment analysis using idioms. In Table 9.1, we list several examples from the dataset.

Sentiment-aware Idiom Recommendation (SIR): This task is the same as Chinese Idiom Recommendation, i.e., given a context with a blank, we are required to select the best idiom from a candidate set. The difference is that we may use sentiment dictionaries like CALO or the dataset constructed from ISC to help us better choose a suitable Chengyu for a given context.

9.2 Explaining Chinese Chengyu Recommendation Model

In Chapter 6, we adopted the attribution method *Integrated Gradients* as an explanation approach for the model. The method shows impressive results over how each character contributes to the final decision.

However, the attributions at character level are not always explainable for human. This requires us to design a better model to ground those attributions to syntactic rules or linguistic cues. For the more challenging near-synonyms issue, identifying the difference and explaining the fitness in a context are still unexploited in the deep learning era.

Bibliography

[1] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *international conference on learning representations*, 2017.

[2] L. J. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

[3] G. Bacon and T. Regier. Probing sentence embeddings for structure-dependent tense. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 334–336, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5440. URL https://aclanthology.org/W18-5440.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[5] A. Bakarov. A survey of word embeddings evaluation methods. *CoRR*, abs/1801.09536, 2018. URL http://arxiv.org/abs/1801.09536.

[6] A. Balahur, R. Steinberger, M. Kabadjov, V. Zavarella, E. van der Goot, M. Halkia, B. Pouliquen, and J. Belyaeva. Sentiment analysis in the news. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/909_Paper.pdf.

[7] T. Baldwin and S. N. Kim. *Handbook of Natural Language Processing, Second Edition*, chapter Multiword expressions, pages 267–292. CRC Press, 01 2010.

[8] T. Baldwin, C. Bannard, T. Tanaka, and D. Widdows. An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 89–96, Sapporo, Japan, July 2003. Association for Computational Linguistics. doi: 10.3115/1119282.1119294. URL https://aclanthology.org/W03-1812.

[9] M. Bansal, K. Gimpel, and K. Livescu. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 809–815, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2131. URL https://aclanthology.org/P14-2131.

[10] M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1023. URL https://aclanthology.org/P14-1023.

[11] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, Mar. 2003. ISSN 1532-4435.

[12] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. doi: 10.1162/tacl_a_00051. URL https://aclanthology.org/Q17-1010.

[13] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

[14] C. Cacciari. The place of idioms in a literal and metaphorical world. *Idioms: Processing, structure, and interpretation*, pages 27–55, 1993.

[15] N. Calzolari, C. J. Fillmore, R. Grishman, N. Ide, A. Lenci, C. MacLeod, and A. Zampolli. Towards best practice for multiword expressions in computational lexicons. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain, May 2002. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2002/pdf/259.pdf.

[16] M. Carpuat and M. Diab. Task-based evaluation of multiword expressions: a pilot study in statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 242–245, Los Angeles, California, June 2010. Association for Computational Linguistics. URL https://aclanthology.org/N10-1029.

[17] D. Chen, J. Bolton, and C. D. Manning. 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)*, pages 2358–2367, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1223. URL https://aclanthology.org/P16-1223.

[18] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318, Santa Cruz, California, USA, June 1996. Association for Computational Linguistics. doi: 10.3115/981863.981904. URL https://aclanthology.org/P96-1041.

[19] X. Chen, L. Xu, Z. Liu, M. Sun, and H. Luan. Joint learning of character and word embeddings. In *Proceedings of the 24th International Conference on Artificial Intelligence*, page 1236–1242. AAAI Press, 2015. ISBN 9781577357384.

[20] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012. URL https://aclanthology.org/W14-4012.

[21] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does bert look at? an analysis of bert's attention. In *BlackBoxNLP@ACL*, 2019.

[22] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, page 160–167, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390177. URL https://doi.org/10.1145/1390156.1390177.

[23] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198.

[24] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

[25] M. Constant, G. Eryiğit, J. Monti, L. van der Plas, C. Ramisch, M. Rosner, and A. Todirascu. Survey: Multiword expression processing: A Survey. *Computational Linguistics*, 43(4):837–892, Dec. 2017. doi: 10.1162/COLI_a_00302. URL https://aclanthology.org/J17-4005.

[26] A. Copestake, F. Lambeau, A. Villavicencio, F. Bond, T. Baldwin, I. A. Sag, and D. Flickinger. Multiword expressions: linguistic precision and reusability. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain, May 2002. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2002/pdf/145.pdf.

[27] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, S. Wang, and G. Hu. Pre-training with whole word masking for chinese bert. *CoRR*, abs/1906.08101, 2019.

[28] Y. Cui, T. Liu, W. Che, L. Xiao, Z. Chen, W. Ma, S. Wang, and G. Hu. 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)*, pages 5883–5889, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1600. URL https://aclanthology.org/D19-1600.

[29] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.58. URL https://aclanthology.org/2020.findings-emnlp.58.

[30] X. Dai, Y. Liu, X. Wang, and B. Liu. WINGS:writing with intelligent guidance and suggestions. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 25–30, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-5005. URL https://aclanthology.org/P14-5005.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[32] B. Dhingra, H. Liu, Z. Yang, W. Cohen, and R. Salakhutdinov. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1168. URL https://aclanthology.org/P17-1168.

[33] X. Duan, B. Wang, Z. Wang, W. Ma, Y. Cui, D. Wu, S. Wang, T. Liu, T. Huo, Z. Hu, and et al. Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension. *Chinese Computational Linguistics*, page 439–451, 2019. ISSN 1611-3349. doi: 10.1007/978-3-030-32381-3_36. URL http://dx.doi.org/10.1007/978-3-030-32381-3_36.

[34] P. Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[35] N. C. ELLIS, R. SIMPSON-VLACH, and C. MAYNARD. Formulaic language in native and second language speakers: Psycholinguistics, corpus linguistics, and tesol. *TESOL Quarterly*, 42(3):375–396, 2008. doi: 10.1002/j.1545-7249.2008.tb00137.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1545-7249.2008.tb00137.x.

[36] A. Ettinger, A. Elgohary, and P. Resnik. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2524. URL https://aclanthology.org/W16-2524.

[37] A. Feldman and J. Peng. Automatic detection of idiomatic clauses. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 435–446, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37247-6.

[38] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In *Proceedings of the*

10th International Conference on World Wide Web, page 406–414, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133480. doi: 10.1145/371920.372094. URL https://doi.org/10.1145/371920.372094.

[39] J. Ganitkevitch, C. Callison-Burch, C. Napoles, and B. Van Durme. Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1168–1179, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1108.

[40] M. Garcia, T. Kramer Vieira, C. Scarton, M. Idiart, and A. Villavicencio. Assessing the representations of idiomaticity in vector models with a noun compound dataset labeled at type and token levels. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2730–2741, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.212. URL https://aclanthology.org/2021.acl-long.212.

[41] M. Garcia, T. Kramer Vieira, C. Scarton, M. Idiart, and A. Villavicencio. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564, Online, Apr. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.310. URL https://aclanthology.org/2021.eacl-main.310.

[42] M. Giulianelli, J. Harding, F. Mohnert, D. Hupkes, and W. Zuidema. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5426. URL https://aclanthology.org/W18-5426.

[43] J. Goodman. A bit of progress in language modeling, 2001.

[44] J. Guo, W. Che, H. Wang, and T. Liu. Learning sense-specific word embeddings by exploiting bilingual resources. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 497–507, Dublin, Ireland, Aug. 2014. Dublin City University and Association for Computational Linguistics. URL https://aclanthology.org/C14-1048.

[45] Z. Guo, Y. Zhao, Y. Zheng, X. Si, Z. Liu, and M. Sun. Thuctc: An efficient chinese text classifier., 2016. URL https://github.com/thunlp/THUCTC.

[46] H. Haagsma, M. Nissim, and J. Bos. The other side of the coin: Unsupervised disambiguation of potentially idiomatic expressions by contrasting senses. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 178–184, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics. URL https://aclanthology.org/W18-4919.

[47] H. Haagsma, J. Bos, and M. Nissim. MAGPIE: A large corpus of potentially idiomatic expressions. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 279–287, Marseille, France, May 2020. European Language Resources

Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.35.

[48] Y. Hao, Y. Zhang, K. Liu, S. He, Z. Liu, H. Wu, and J. Zhao. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1021. URL https://aclanthology.org/P17-1021.

[49] G. Heidorn. Intelligent writing assistance. *Handbook of natural language processing*, pages 181–207, 2000.

[50] I. Hendrickx, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Szpakowicz, and T. Veale. SemEval-2013 task 4: Free paraphrases of noun compounds. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 138–143, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/S13-2025.

[51] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.pdf.

[52] J. Hewitt and P. Liang. Designing and interpreting probes with control tasks. 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)*, pages 2733–2743, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275.

[53] J. Hewitt and C. D. Manning. A structural probe for finding syntax in word representations. 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)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419.

[54] F. Hill, A. Bordes, S. Chopra, and J. Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

[55] W. Y. Ho, C. Kng, S. Wang, and F. Bond. Identifying idioms in Chinese translations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 716–721, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/462_Paper.pdf.

[56] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[57] A. Horbach, A. Hensler, S. Krome, J. Prange, W. Scholze-Stubenrecht, D. Steffen, S. Thater, C. Wellner, and M. Pinkal. A corpus of literal and idiomatic uses of German infinitive-verb compounds. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 836–841, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1135.

[58] J. Huang, F. Qi, C. Yang, Z. Liu, and M. Sun. COS960: A chinese word similarity dataset of 960 word pairs. *CoRR*, abs/1906.00247, 2019. URL http://arxiv.org/abs/1906.00247.

[59] P. Isabelle, C. Cherry, and G. Foster. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1263. URL https://aclanthology.org/D17-1263.

[60] G. Jawahar, B. Sagot, and D. Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://aclanthology.org/P19-1356.

[61] Z. Jiang, B. Zhang, L. Huang, and H. Ji. Chengyu cloze test. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 154–158, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-0516. URL https://aclanthology.org/W18-0516.

[62] P. Jin and Y. Wu. SemEval-2012 task 4: Evaluating Chinese word similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 374–377, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1049.

[63] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. doi: 10.1162/tacl_a_00300. URL https://aclanthology.org/2020.tacl-1.5.

[64] D. Kauchak and R. Barzilay. Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 455–462, New York City, USA, June 2006. Association for Computational Linguistics. URL https://aclanthology.org/N06-1058.

[65] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL https://aclanthology.org/D14-1181.

[66] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[67] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. *1995 International Conference on Acoustics, Speech, and Signal Processing*, 1:181–184 vol.1, 1995.

[68] E. Kochmar, S. Gooding, and M. Shardlow. Detecting multiword expression type helps lexical complexity assessment. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4426–4435, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.545.

[69] M. Kurfalı and R. Östling. Disambiguation of potentially idiomatic expressions with contextual embeddings. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 85–94, online, Dec. 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.mwe-1.11.

[70] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082.

[71] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

[72] O. Levy and Y. Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2050. URL https://aclanthology.org/P14-2050.

[73] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

[74] H. Li and B. Yuan. Chinese word segmentation. In *Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation*, pages 212–217, Singapore, Feb. 1998. Chinese and Oriental Languages Information Processing Society. doi: http://hdl.handle.net/2065/12081. URL https://aclanthology.org/Y98-1020.

[75] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du. Analogical reasoning on Chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting*

of the Association for Computational Linguistics (Volume 2: Short Papers), pages 138–143, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2023. URL https://aclanthology.org/P18-2023.

[76] P. Lison and A. Kutuzov. Redefining context windows for word embedding models: An experimental study. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 284–288, Gothenburg, Sweden, May 2017. Association for Computational Linguistics. URL https://aclanthology.org/W17-0239.

[77] C. Liu and R. Hwa. Phrasal substitution of idiomatic expressions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 363–373, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1040. URL https://aclanthology.org/N16-1040.

[78] Y. Liu, B. Liu, L. Shan, and X. Wang. Modelling context with neural networks for recommending idioms in essay writing. *Neurocomputing*, 275:2287–2293, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.11.005. URL https://www.sciencedirect.com/science/article/pii/S0925231217317198.

[79] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[80] Y. Liu, B. Pang, and B. Liu. Neural-based Chinese idiom recommendation for enhancing elegance in essay writing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5522–5526, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1552. URL https://aclanthology.org/P19-1552.

[81] S. Long, R. Wang, K. Tao, J. Zeng, and X. Dai. Synonym knowledge enhanced reader for Chinese idiom reading comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3684–3695, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.329. URL https://aclanthology.org/2020.coling-main.329.

[82] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://aclanthology.org/D15-1166.

[83] N. Madnani and B. J. Dorr. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387, Sept. 2010. doi: 10.1162/coli_a_00002. URL https://aclanthology.org/J10-3003.

[84] J. Mallinson, R. Sennrich, and M. Lapata. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, Valencia, Spain, Apr. 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-1083.

[85] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

[86] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[87] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[88] A. Mnih and G. E. Hinton. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1081–1088. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3583-a-scalable-hierarchical-distributed-language-model.pdf.

[89] D. Newman, N. Koilada, J. H. Lau, and T. Baldwin. Bayesian text segmentation for index term identification and keyphrase extraction. In *Proceedings of COLING 2012*, pages 2077–2092, Mumbai, India, Dec. 2012. The COLING 2012 Organizing Committee. URL https://aclanthology.org/C12-1127.

[90] G. Nunberg, I. A. Sag, and T. Wasow. Idioms. In S. Everson, editor, *Language*, volume 70, pages 491–538. Linguistic Society of America, 1994.

[91] K. Omelianchuk, V. Atrasevych, A. Chernodub, and O. Skurzhanskyi. Gector–grammatical error correction: Tag, not rewrite. *arXiv preprint arXiv:2005.12592*, 2020.

[92] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

[93] J. Peng, A. Feldman, and E. Vylomova. Classifying idiomatic and literal expressions using topic models and intensity of emotions. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2019–2027, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1216. URL https://aclanthology.org/D14-1216.

[94] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

[95] M. Pershina, Y. He, and R. Grishman. Idiom paraphrases: Seventh heaven vs cloud nine. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 76–82, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-2709. URL https://aclanthology.org/W15-2709.

[96] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 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)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

[97] M. E. Peters, M. Neumann, L. Zettlemoyer, and W.-t. Yih. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1179. URL https://aclanthology.org/D18-1179.

[98] S. Prabhumoye, Y. Tsvetkov, R. Salakhutdinov, and A. W. Black. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1080. URL https://aclanthology.org/P18-1080.

[99] Y. Qiu, H. Li, S. Li, Y. Jiang, R. Hu, and L. Yang. Revisiting correlations between intrinsic and extrinsic evaluations of word embeddings. In M. Sun, T. Liu, X. Wang, Z. Liu, and Y. Liu, editors, *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 209–221, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01716-3.

[100] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training, 2018. URL https://openai.com/blog/language-unsupervised/.

[101] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[102] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[103] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

[104] C. Ramisch. Multiword expressions acquisition. 2015.

[105] S. Reddy, D. McCarthy, and S. Manandhar. An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218, Chiang Mai, Thailand, Nov. 2011. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I11-1024.

[106] I. A. Sag, T. Baldwin, F. Bond, A. Copestake, and D. Flickinger. Multiword expressions: A pain in the neck for nlp. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 1–15, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45715-2.

[107] I. Samenko, A. Tikhonov, and I. P. Yamshchikov. Synonyms and antonyms: Embedded conflict. *CoRR*, abs/2004.12835, 2020. URL https://arxiv.org/abs/2004.12835.

[108] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1036. URL https://aclanthology.org/D15-1036.

[109] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016. URL http://arxiv.org/abs/1611.01603.

[110] C. C. Shao, T. Liu, Y. Lai, Y. Tseng, and S. Tsai. Drcd: a chinese machine reading comprehension dataset, 2018.

[111] Y. Shao, R. Sennrich, B. Webber, and F. Fancellu. 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)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1005.

[112] X. Shi, I. Padhi, and K. Knight. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1159. URL https://aclanthology.org/D16-1159.

[113] V. Shwartz and I. Dagan. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419, 2019. doi: 10.1162/tacl_a_00277. URL https://aclanthology.org/Q19-1027.

[114] C. Sporleder and L. Li. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 754–762, Athens, Greece, Mar. 2009. Association for Computational Linguistics. URL https://aclanthology.org/E09-1086.

[115] T.-R. Su and H.-Y. Lee. Learning Chinese word representations from glyphs of characters. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 264–273, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1025. URL https://aclanthology.org/D17-1025.

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

[117] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang. Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975, Apr. 2020. doi: 10.1609/aaai.v34i05.6428. URL https://ojs.aaai.org/index.php/AAAI/article/view/6428.

[118] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 3319–3328. JMLR.org, 2017.

[119] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks, 2014.

[120] K. S. Tai, R. Socher, and C. D. Manning. 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)*, pages 1556–1566, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1150. URL https://aclanthology.org/P15-1150.

[121] M. Tan and J. Jiang. A BERT-based dual embedding model for Chinese idiom prediction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1312–1322, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.113. URL https://aclanthology.org/2020.coling-main.113.

[122] M. Tan and J. Jiang. Does BERT understand idioms? a probing-based empirical study of BERT encodings of idioms. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1397–1407, Held Online, Sept. 2021. INCOMA Ltd. URL https://aclanthology.org/2021.ranlp-1.156.

[123] M. Tan, J. Jiang, and B. T. Dai. A bert-based two-stage model for chinese chengyu recommendation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(6), Aug. 2021. ISSN 2375-4699. doi: 10.1145/3453185. URL https://doi.org/10.1145/3453185.

[124] S. Tan and J. Zhang. An empirical study of sentiment analysis for chinese documents. *Expert Syst. Appl.*, 34(4):2622–2629, May 2008. ISSN 0957-4174. doi: 10.1016/j.eswa.2007.05.028. URL https://doi.org/10.1016/j.eswa.2007.05.028.

[125] H. Tayyar Madabushi, E. Gow-Smith, C. Scarton, and A. Villavicencio. AStitchInLanguageModels: Dataset and methods for the exploration of idiomaticity in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3464–3477, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.294. URL https://aclanthology.org/2021.findings-emnlp.294.

[126] H. Tayyar Madabushi, E. Gow-Smith, M. Garcia, C. Scarton, M. Idiart, and A. Villavicencio. SemEval-2022 Task 2: Multilingual Idiomaticity Detection and Sentence Embedding. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics, 2022.

[127] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, T. R. McCoy, N. Kim, V. B. Durme, R. S. Bowman, D. Das, and E. Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. *ICLR*, 2019.

[128] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://aclanthology.org/W03-0419.

[129] S. Tratz. *Semantically-enriched parsing for natural language understanding*. University of Southern California, 2011.

[130] J. Turian, L.-A. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-1040.

[131] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[132] E. Wallace, Y. Wang, S. Li, S. Singh, and M. Gardner. Do NLP models know numbers? probing numeracy in embeddings. 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)*, pages 5307–5315, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1534. URL https://aclanthology.org/D19-1534.

[133] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

[134] B. Wang, A. Wang, F. Chen, Y. Wang, and C.-C. J. Kuo. Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, 8:e19, 2019. doi: 10.1017/ATSIP.2019.12.

[135] L. Wang and S. Yu. Construction of Chinese idiom knowledge-base and its applications. In *Proceedings of the 2010 Workshop on Multiword Expressions: from Theory to Applications*, pages 11–18, Beijing, China, Aug. 2010. Coling 2010 Organizing Committee. URL https://aclanthology.org/W10-3703.

[136] L. Wang, S. Yu, X. Zhu, and Y. Li. Chinese idiom knowledge base for chinese information processing. In *Proceedings of the 13th Chinese Conference on Chinese Lexical Semantics*, pages 302–310, Berlin, Heidelberg, 2013. Springer-Verlag. ISBN 978-3-642-36336-8. doi: 10.1007/978-3-642-36337-5_31. URL http://dx.doi.org/10.1007/978-3-642-36337-5_31.

[137] S. Wang. *Chinese Multiword Expressions: Theoretical and Practical Perspectives*. Springer Singapore, 2019. ISBN 9789811385100. URL https://books.google.com.sg/books?id=VtTBDwAAQBAJ.

[138] S. Wang and J. Jiang. Learning natural language inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1170. URL https://aclanthology.org/N16-1170.

[139] S. Wang and J. Jiang. 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*, 2017. URL https://openreview.net/forum?id=HJTzHtqee.

[140] S. WANG and J. JIANG. Machine comprehension using match-lstm and answer pointer. ICLR, 2017.

[141] W. Wang, B. Bi, M. Yan, C. Wu, J. Xia, Z. Bao, L. Peng, and L. Si. Structbert: Incorporating language structures into pre-training for deep language understanding. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJgQ4lSFPH.

[142] X. Wang, Y. Jia, B. Zhou, Z.-Y. Ding, and Z. Liang. Computing semantic relatedness using chinese wikipedia links and taxonomy. *Journal of Chinese Computer Systems*, 32(11):2237–2242, 2011.

[143] X. Wang, H. Zhao, T. Yang, and H. Wang. Correcting the misuse: A method for the Chinese idiom cloze test. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 1–10, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.deelio-1.1. URL https://aclanthology.org/2020.deelio-1.1.

[144] J. Wieting, J. Mallinson, and K. Gimpel. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 274–285, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1026. URL https://aclanthology.org/D17-1026.

[145] L. Williams, C. Bannister, M. Arribas-Ayllon, A. Preece, and I. Spasić. The role of idioms in sentiment analysis. *Expert Systems with Applications*, 42(21):7375 – 7385,

2015. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2015.05.039. URL http://www.sciencedirect.com/science/article/pii/S0957417415003759.

[146] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

[147] Y. Wu and W. Li. Overview of the nlpcc-iccpol 2016 shared task: Chinese word similarity measurement. In C.-Y. Lin, N. Xue, D. Zhao, X. Huang, and Y. Feng, editors, *Natural Language Understanding and Intelligent Applications*, pages 828–839, Cham, 2016. Springer International Publishing. ISBN 978-3-319-50496-4.

[148] Z. Wu, Y. Chen, B. Kao, and Q. Liu. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.383. URL https://aclanthology.org/2020.acl-main.383.

[149] Z. Xiaobing and Q. Lina. A survey of chinese idioms in mainstream print media. *Language Teaching and Linguistic Studies*, 2010.

[150] J. Xu, J. Liu, L. Zhang, Z. Li, and H. Chen. Improve Chinese word embeddings by exploiting internal structure. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1041–1050, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1119. URL https://aclanthology.org/N16-1119.

[151] L. Xu, H. Hu, X. Zhang, L. Li, C. Cao, Y. Li, Y. Xu, K. Sun, D. Yu, C. Yu, Y. Tian, Q. Dong, W. Liu, B. Shi, Y. Cui, J. Li, J. Zeng, R. Wang, W. Xie, Y. Li, Y. Patterson, Z. Tian, Y. Zhang, H. Zhou, S. Liu, Z. Zhao, Q. Zhao, C. Yue, X. Zhang, Z. Yang, K. Richardson, and Z. Lan. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.419. URL https://aclanthology.org/2020.coling-main.419.

[152] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.

[153] S. M. Yimam, H. Martínez Alonso, M. Riedl, and C. Biemann. Learning paraphrasing for multiword expressions. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 1–10, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-1801. URL https://aclanthology.org/W16-1801.

[154] J. Yu, X. Jian, H. Xin, and Y. Song. Joint embeddings of Chinese words, characters, and fine-grained subcharacter components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 286–291, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1027. URL https://aclanthology.org/D17-1027.

[155] L. Yu and A. Ettinger. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.397. URL https://aclanthology.org/2020.emnlp-main.397.

[156] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang. Refining word embeddings using intensity scores for sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):671–681, 2017.

[157] X. L. L. H. P. Yu and R. H. C. Jianmei. Constructing the affective lexicon ontology [j]. *Journal of the China Society for Scientific and Technical Information*, 2:6, 2008.

[158] T. Zagibalov and J. Carroll. Unsupervised classification of sentiment and objectivity in Chinese text. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008. URL https://aclanthology.org/I08-1040.

[159] X.-b. ZENG, Z.-p. ZHANG, R. LIU, E.-h. YANG, and P. ZHANG. Investigation and discussion on chinese idioms and idiomatic phrases in the chinese language situation report. *Journal of Chinese Information Processing*, page 06, 2008.

[160] B. Zhang, W. Sun, X. Wan, and Z. Guo. Pku paraphrase bank: A sentence-level paraphrase corpus for chinese. In J. Tang, M.-Y. Kan, D. Zhao, S. Li, and H. Zan, editors, *Natural Language Processing and Chinese Computing*, pages 814–826, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32233-5.

[161] Q. Zhang, X. Liu, and J. Fu. Neural networks incorporating dictionaries for chinese word segmentation. 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16368.

[162] X. Zhang and Y. LeCun. Which encoding is the best for text classification in chinese, english, japanese and korean?, 2017.

[163] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL https://aclanthology.org/P19-1139.

[164] Z. Zhao, T. Liu, S. Li, B. Li, and X. Du. Ngram2vec: Learning improved word representations from ngram co-occurrence statistics. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 244–253, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1023. URL https://aclanthology.org/D17-1023.

[165] C. Zheng, M. Huang, and A. Sun. ChID: A large-scale Chinese IDiom dataset for cloze test. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 778–787, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1075. URL https://aclanthology.org/P19-1075.

[166] G. Zhou, Z. Xie, T. He, J. Zhao, and X. T. Hu. Learning the multilingual translation representations for question retrieval in community question answering via non-negative matrix factorization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(7):1305–1314, 2016.

[167] X. Zhou, X. Wan, and J. Xiao. Attention-based LSTM network for cross-lingual sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 247–256, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1024. URL https://aclanthology.org/D16-1024.

[168] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1561.