Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Can Computer Understand Chinese Internet Slang?

Semester Thesis

Zhenjie Jiang

zhejiang@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Ye Wang, Meng Zhao
Prof. Dr. Roger Wattenhofer

February 4, 2022

Acknowledgements

I express my special thanks and gratitude to my main supervisor, Ye Wang, who has assisted me during my individual, providing many helpful advice and recommendations, which helped me greatly during this project.

I would like to also thank Zhao Meng, for providing help insights and Prof Dr. Roger Wattenhofer for giving the opportunity to do this project.

Abstract

Internet Slang has becoming a very popular phenomenon on the Chinese Internet. One of the most widely used Chinese Internet Slang is abbreviations created by taking the first letter of the Pinyin representation of each Chinese character in a word. These Internet Slang are often very hard to understand for people who don't interact with them everyday. In this paper, we demonstrate it is possible to translate these types of abbreviations to proper Chinese with modern machine learning models. We tested a sequence-to-sequence model, a sequence-to-word model and a multiple choice model, which we achieved 49.04%, 70.89% and 95.75% accuracy respectively. We further tested the multiple choice model on posts from popular Chinese social media Weibo, and demonstrated promising results, and shortcomings.

Contents

Chapter 1

Introduction

Internet Slang is a common occurrence across the digital word, they are abbreviated words used for simplifying communication through a keyboard. Despite the surge in the usage of Internet Slang as the internet became more widespread, they are still often very confusing for people who are not accustomed to them. This is especially true for the Chinese Language due the variety of the slang (see 2.1). In this papers, we focus on a particular type of Chinese Internet Slang that is formed by taking the initial characters of the pinyin of the Chinese characters in a word/phrase (see 2.1.1) [1]. We refer as "head abbreviation" through out this paper. Head abbreviations are often very confusing for readers because the large amount of the Chinese words and phrases which share the same pinyin initials. This makes browsing the Chinese internet more difficult and confusing even for people who do use internet often [2].

In this paper, we aim to develop machine learning models for translating head abbreviations to help facilitate browsing online. To the best of our knowledge, there is very little amount of work done on translating internet slang. The prior attempts have been mostly based on a dictionary approach, or treating internet slang as new words [3], however such approach is very limiting due to the constant flow of new slang. With a machine learning approach, we hope to create a more adaptable model for translating head abbreviations. Modern machine learning model, especially the transformers models have shown to be capable of capturing a lot of knowledge regarding a particular language [4]. If the model is given a text, we theorize it is possible for the model to translate the head abbreviations, given the rest of the text.

In this paper, we propose three approaches based on the transformers model [5] for this task: (1) A sequence-to-sequence model, where we attempts to recreate the entire input text with the abbreviation translated. (2) A sequence-to-word model, where we only outputs the translation of the head abbreviation without the rest of the text. (3) A multiple choice model, where the model has to pick the correct translation out of four choices given the input text as context. We analyze and discuss the advantage and disadvantage of each approach.

Chapter 2

Background

2.1 Chinese Internet Slangs

2.1.1 Chinese to Latin abbreviation

Chinese to Latin abbreviation is a type of abbreviation that consists of replacing Chinese characters with alphabetical letters. This type of abbreviation is often used online as they are easier to type with a QWERTY keyboard compared to the original Chinese characters [2]. In addition, the abbreviation of vulgar and sensitive words are frequently used to avoid censorship or filters, for example, 你妈死了(pinyin: ni ma si le, English: your mum is dead) → NMSL [1].

The most common type of Chinese to Latin abbreviation uses the initial letter of the pinyin representation of each Chinese character to represent the word [1] (this will be referred as head abbreviation), i.e. 中国(pinyin: zhong guo, English: China) → ZG. Although this is very similar to acronyms in English, but since a lot of Chinese characters share the same pinyin (see 2.1), and the pinyin representation of a character is quite different from the Chinese character itself, it is not easy to understand what they mean without prior knowledge. Chen [2] surveyed 157 native Mandarin Chinese speakers and found a significant amount of the Chinese people did not recognize some of the abbreviations, such as the ones for politically sensitive phrases where 40.8% did not understand any of them. The usage of these abbreviations also often differs between the communities and social media platforms, which makes them even harder to recognize. This project mainly tackles this type of internet slang to help people read Chinese online contents.

Other types of Latin abbreviations include mixing English and Chinese, for example: “笑cry”, which directly translates to “laugh cry”, represents the emoji “Face with Tears of Joy”; “A片”, where “A” comes from the English word “adult”, and “片” is “video" in Chinese, means “adult video”. This type of slang is typically less common, and the use case tends to be more specific. There are also cases where Chinese character are replaced with similar sounding English letters, i.e. 屁民(pinyin: pi min, English: ordinary people) → p民, where p sounds exactly

Weibo Post	Meaning of "zs"
好家伙 打开微博都跟着ZS ZS的我以为全世界都玩魔兽了这版本战士加强了还是怎么滴	战士/warrior
恋爱就是慢性zs	自杀/suicide
小憩zs粉	郑爽/Zheng Shuang (name)

Table 2.1: The table shows some of the texts containing the abbreviation "zs" , where each "zs" refers to a different word.

the same as pi in pinyin. These types of abbreviation are easy to understand for Chinese speakers because the pronunciation is similar. However, they make it harder to identify head abbreviation because words in Chinese are not separated by spaces, for example, in the case "A片", the reader can technically view "A" as a head abbreviation by itself, instead of perceiving "A片" as a single word/phrase. Therefore, knowing whether a word contains only letters or contains both Chinese character and letters can be difficult.

2.1.2 Chinese Numeronyms

Chinese Numeronyms take the form of replacing words with numbers [6]. The most common case for Chinese Numeronyms is replacing words/Chinese Characters with a similar sounding numbers in Chinese, such as 一身一世(pinyin: yi sheng yi shi, English: one life, one world) → 1314 (pinyin: yi san yi si). This is typically done with one number per Chinese character. In some cases, the Numeronyms became viral and gained cultural significance, as it is often really easy to associated a Numeronym with a date, for example, 20th May has now become another Valentine's day in China, due to the Chinese Numeronym 520 (Chinese: "我爱你", English:"I love you") [7]. In addition, it can also extend to some English words/phrases, such as bye bye → 88 (pinyin: bai bai), where English words are replaced with similar sounding numbers in Chinese.

There are also special Chinese Numeronyms that are derived from a story or cultural phenomena, for example "250" (English: idiot), which did not come from a similar-sounding Chinese word, but originated from a story, although nowadays most users are unaware of its origin [6].

2.1.3 Chinese to Chinese abbreviation

Chinese to Chinese abbreviation consists of keeping only certain characters of the original word. This type of abbreviation follows certain grammatical rules as demonstrated by Yang et al. [8] who attempted to automatically generate these abbreviations. The most common use of Chinese to Chinese abbreviation is to shorten long names of facilities. Some of the examples are: 中国中央

电视台(English: China central television) → 央视, 清华大学(English: Tsinghua University) → 清华. While other types of abbreviation are mainly popular on the Internet, Chinese to Chinese abbreviation is also frequently used in the speaking language [8].

2.1.4 Neologisms

Neologisms are newly created words to describe new phenomena on the Internet, such as "五毛党" which refers to people who are paid to leave comments or reviews. Although this type of slang is interesting, we are not concerned with them in this project, as they carry new meanings and unlike the abbreviation, they do not have an original representation.

2.1.5 Dialects

There are also internet slang borrowed from other languages or certain local dialect, such as "Otaku" from Japanese which means nerd in English. However, since these type of Slang requires knowledge of multiple languages, it is out of the scope of the project.

Chapter 3

Related Work

3.1 JianPin

JianPin (简拼) [9] is simple spelling method of Chinese input method, which allows the users to type parts of the pinyin, such as only typing initials of pinyin. The method matches the input to words / phrases using a phonetic dictionary, and return a list of matched words / phrases arranged according to word frequency [9]. This method is used in most modern Chinese input methods, such as Gboard from Google.

Although this method can match head abbreviations to Chinese characters, it still requires the user to pick the word / phrase with the desired meaning [10]. Therefore it does not actually translates the Slang nor helps the user to understand the meaning.

3.2 Fill-in-the-blanks

Fill-in-the-blank task involves texts with missing words, which the model to predict [11]. Our task of translating head abbreviations in sentences shares similarity with the fill-in-the-blank task because abbreviations can be viewed as blanks in the sentence.

In previous work, Jiang et al. [12] created a neural recommendation model that selects a Chengyu (four word Chinese Idiom) out of four choices which best fits the blank of the sentence. They used a Bi-LSTM network to encode the words, and compared the representation of the Chengyu definitons and the query (the sentence), and finally determine the probability score of each candidate with a Linear function. According to their experiment, their model were able to outperform Chinese university students.

However, unlike Jiang et al. [12]'s task which allows the model to choose the answer out of four choices, our task requires the model to determine translation of abbreviation by itself without the meaning of the abbreviations explicitly given,

which makes the task more challenging.

3.3 Transformers Model

Transformers is a deep learning architecture which utilizes the attention mechanism [5], and it is currently the state-of-the-art architecture for NLP tasks.

BERT [13] is one of the most well known transformers model which has demonstrated great performances in many NLP tasks. In this paper, we utilizes BERT for the multiple choices model.

For the sequence-to-sequence model and the sequence-to-word model, we uses BART. BART [14] is a pretrained sequence to sequence transformers model from Facebook AI. BART is a denoising autoencoder which can recreate the original document from a corrupted documents, which is similar to mapping head abbreviations to its original form. Furthermore BART has also demonstrated great performance in translation tasks, which makes it a fitting choices to translate head abbreviations in a sequence-to-sequence manner.

Chapter 4

Dataset

4.1 Dataset

We use the public dataset webtext2019zh[1], an online question and answering dataset. It is good for training a model for dealing with online texts because the formality of each entry varies, some of the entries contains both English and Chinese which is representative to the Chinese online contents. The original dataset contains 4.12 million question and answering pairs for training, and 68k pairs each for validation and testing. We separated the pairs, and only kept the answers because questions are not necessary for our purpose of training a model to translate head abbreviations.

4.2 Data Process

In order to format the dataset to be used for training the models, we first segmented the entries to words using the jieba[2] library, and randomly converted a Chinese word from each entry to head abbreviations by first translating the word into pinyin and keeping the initial letter. Due to resource limitation, we only kept entries that are less or equal to 100 characters long.

4.2.1 Multiple Choice

For multiple choice model, we need to generate distractors for each entry. In order to create reasonable distractors, we first mapped each word from the dataset to their corresponding head abbreviations. Then for each entry in the dataset, we randomly selected three words with the same head abbreviation as the one in the entry which was created using the method above. Last, We shuffle and combine the three words and the correct translation of the chosen head abbreviation of the entry to form all the choices.

[1] https://github.com/brightmart/nlp_chinese_corpus
[2] https://github.com/fxsjy/jieba

Chapter 5

Methods

We propose three approaches for the translation of head abbreviations: sequence-to-sequence, sequence-to-word and multiple-choice.

5.1 Sequence-to-Sequence Model (seq2seq)

In the seq2seq model, we fine-tune a version of BART[14] that is pretrained for the Chinese language[1]. The model is an autoencoder, which consists of a Bidirectional Encoder and an Autoregressive Decoder. The encoder encodes the input text to create an embedding, and the decoder takes in the embedding and generates the output text.

In this approach we treat the task as an machine translation task, where the inputs are the texts containing the head abbreviation, and the translations are the input texts recreated with the head abbreviation translated into Chinese characters (Fig 5.1 shows an example of the input and output).

In order to achieve this goal, we fine-tune the BART model through supervised learning, where the training data consist of input and target (the expected output) pairs. The model takes in the input and produces an output, which we compare with the target and update the model with the cross-entropy loss.

5.2 Sequence-to-Word Model (seq2word)

The seq2word model is a variation of the seq2seq model. The overall architecture of the model is the same as the seq2seq model, however, we simplify the problem

[1] https://huggingface.co/fnlp/bart-base-chinese

Figure 5.1: Sequence-to-Sequence example

Figure 5.2: Sequence-to-Word example

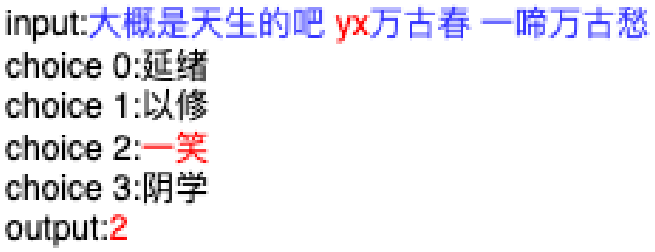

Figure 5.3: Multiple Choice example

by changing the target output to only the translation of the head abbreviation. We theorize this modification will conserve more model capacity for the actual translation of the head abbreviation since it is no longer necessary for the embedding created by the encoder to capture the entire text, as the decoder does not need to recreate the complete input.

In the seq2seq model, the model needs to identify where the head abbreviation is in the text, which is a very difficult task in itself. The texts can often contain other alphabetical letters which are not apart of the target head abbreviation, which can confuse the model, i.e. "我bzd我该怎么办，help" (I don't know what to do, help), in this sentence "bzd" is the target head abbreviation which translates to "不知道" (don't know), while also containing the English word "help". In order to simplify this problem, we further change the input text through appending the head abbreviation at the end of the text. This idea is inspired by models used in the Question and Answering tasks such as SQUAD [15], where people typically construct the input as the concatenation of the context which provides the information for answering the question and the question it self. For our task, the input text can be viewed as the context, and the head abbreviation is the question. Under this format, the location of the head abbreviation is now fixed, which means the model no longer needs to identify where the head abbreviation is, which reduces the complexity of the problem, and should help improve model performance. Fig 5.2 shows an example of input and output. The fine-tuning method remain the same as the seq2seq model.

5.3 Multiple Choice Model (choice)

In the multiple choice model, we take a different approach for the task. We treat the problem as a multiple choice task, where the model has to choose the correct translation of the head abbreviation out of 4 choices. This approach shares

similarity to Jiang et al. [12], which demonstrated that neural network models can perform better than human on certain multiple choice tasks. Because this is not longer a seq2seq task, instead of BART, we use a version of BERT[13] that is pretrained for the Chinese language [16] with a linear layer on top of the pooled output of BERT and a softmax function.

In this model, we construct each entry as 4 text and choice pairs, i.e. [[text, choice0], [text, choice1], [text, choice2], [text, choice3]], where the text is the input text with the head abbreviation we want to translate. The model outputs a score for each choice through the softmax function, and we pick the choice with the highest score as the answer/translation. Fig 5.3 demonstrates an example.

The multiple choice task should be an easier task than before, as by restricting the answer to 4 choices, it greatly reduces the search space of the answers.

5.3.1 Automation

The base Choice model makes the assumption that user already have some ideas for the translation of the head abbreviations are, as it requires to user to input the choices. In order to alleviate this problem, we further developed a version that adds some automation to the inference process of this model.

We choose to automate the model by using a dictionary, which maps head abbreviations to the possible Chinese words. With the dictionary, we can recursively apply the model, by using the corresponding words in the dictionary as choices until we get a single final answer.

Unfortunately, a complete dictionary that contrains all the mappings does not yet exist. For the purpose of this project, we uses the API 'nbnhhsh' [2] which contains some possible translations for some of the popular abbreviations, which we utilized to create the dictionary. However, the 'nbnhhsh' dictionary is limited in the results it produces, the possible words for the head abbreviations generated by 'nbnhhsh' is far from complete, and sometimes lacks in quality.

This approach still requires the user to identify the head abbreviation in the text and has the drawback of needing to recursively apply the model, which can cause performance issues if there are too many possible Chinese words for the head abbreviation.

[2] https://github.com/itorr/nbnhhsh

Chapter 6

Experimentation & Evaluation

For the experiment, we used 300k data for training, and 15k data for testing. The dataset is described in Section 4, and how each model uses the data is explained in Section 5. All the models were trained for 3 epochs with Adam optimizer and a linear scheduler which reduces the learning rate as the number of training steps increase.

In the experiment, We found that increasing the maximum length of the embedding had noticeable impact on the models' performance, despite we capped the length of the input texts to 100. However, increasing the length of the embedding also increases the amount of VRAM consumed. In the end, we set the maximum length of the embedding of all three models to 512, which seems to provide a good trade off between performance and resource consumption

The performance of each model is evaluated on the test data which was not seen during training. We measure the performance using accuracy. Given the test data $D_{test} = \{(x_i, y_i))\}_{i=1}^{N}$, and model f, where x is the input, y is the target and N is the number of test data, the accuracy is formulate as

$$\text{Accuracy}(\mathcal{D}_{\text{test}}) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x,y)\in\mathcal{D}_{\text{test}}} \mathbf{1}_{[y=f(x)]}.$$

The results can be found in Table 6.1.

Model	Test Accuracy
Seq2Seq	49.04%
Seq2Words	70.89%
Choice	95.75%

Table 6.1: Comparison of model accuracy on test data

6.1 seq2seq performance

In the sequence-to-sequence approach, the model has to identify where the head abbreviation is, translate it and recreate the sentence. This task is much more complex compare to the other two approaches, which likely lead to the the worst performance out of the three models with an accuracy of 49.04%.

The accuracy suffered from the nature of the task, where in some instances, despite the model managed to translate the head abbreviation, it failed to recreate the rest of the input text exactly. For example, given input text "你是什么样d人，终究会遇到同样的人。", and target "你是什么样的人，终究会遇到同样的人", with head abbreviat "d", which translates to "的", the model ouputed "你是什么样的人，终究会遇到同的人". The model's output is exactly the same as target, except it output the traditional Chinese version of "样", which is "樣". In practice, in this case both the target and the output of the model would serve the same purpose for the user.

Despite the poor performance, the seq2seq approach is also the most flexible approach, as it can be easily extended to translate multiple head abbreviations within the same text without much modification to the model. The model required the least amount of prepossessing on the input text.

6.2 seq2word performance

Under the seq2word approach, the performance is much improved over the seq2seq model and achieved 70.89% accuracy. Some of the accuracy improvement comes from that the model do not need to output the entire text, which contributed to a significant amount of error from the seq2seq model. Additionally, the simplification of the overall task means, the model is only doing the translation of the head abbreviation and not other unnecessary tasks.

Although this is technically a downgrade in terms of functionality, as the model needs the user to identify the head abbreviations, but the task of locating the head abbreviation is trivial for a human user. And the model demonstrates that it is possible for model to learn to translate head abbreviations with a reasonable accuracy.

6.3 choice performance

For this experiment, we used the base version of the model without the automation, as the 'nbnhhsh' dictionary does not contain all the head abbreviations in the test dataset, instead we used the choices generated during data processing, which is described in section 4.2.1 .

The multiple choice model achieved the best performance out of the three approaches with 95.75%. This is expected as the task is greatly simplified, since the correct answer is always contained in one of the choices. The model was able to pick the correct choice majority of the time, despite the supplied choices are shared the same head abbreviations, which shows that the model is not blindly matching pinyin to chinese character. If the user already have some guesses for translation, the choice model is really effective.

Although The model demonstrates outstanding performance, it does make the assumption that user already have some ideas for the translation of the head abbreviations are, as it requires to user to input the choices, which can make it less useful for certain user. Further, we also attempted to use the automation (Section 5.3.1) strategy with the dictionary created from mapping all words in the training to their head abbreviations, however, due to the need to recursively apply the model, the run time becomes extremely long.

6.4 Further Evaluation

In order to evaluate the model's performance in a more realistic environment, we applied the automated version of the multiple choice model on Weibo posts (Weibo is a popular Chinese social media, similar to Twitter). We randomly sampled 82 Weibo posts that contained head abbreviations and compared the predictions of the model with the translations from ourselves. For simplicity, we limited the length of the head abbreviation to two.

Out of the 82 posts, the model were able to correctly predict 31 of them. Interestingly, we found that out of the 51 posts that the model got wrong, at least 20 of them, the translation of head abbreviations were names of celebrities, where many of them were not in the choices given by the "nbnhhsh" dictionary. We were able to determined these head abbreviations represented names, since they were usually accompany by other celebrities' name, and are often used as noun in a way clearly representing a human. However, in some of the case, it was only decipherable because we had prior knowledge on what people typically post. In fact, there are also 12 posts where we weren't able to decider what the head abbreviation meant.

Also curiously, out of the head abbreviations that the model managed to translate, a number of the translations were never seen during training, which shows model were able to learn more than just simply memorizing the correct mapping.

Weibo is a very challenging task for the model, compared to our training data, the texts are much less formal compared to the training data, and often contain very little information for even experienced human to determine the meaning. For example one of the posts is "11.30fh", here "fh" translates to "反黑", which refers

the event where fans make a lot of comments to hide the bad comments often regarding a celebrity they follow. These types of posts contains very minimal amount of information for the model to use, as it is only intended for people already familiar with the contexts. Many Weibo posts also contain image data which the model cannot access.

One potential way to improve the model for Weibo data might be to pre-train a transformer model specifically for Weibo, similarly to the "Bertweet" [17] model which pre-trained the transformer model on Twitter data, which significantly improved performance for Twitter based tasks. However, due to resource limitation, we couldn't test this hypothesis out.

Chapter 7

Conclusion

In conclusion, we developed three type of machine learning model to translate head abbreviations, namely, sequence-to-sequence model, sequence-to-word model and multiple-choice model. We were able to achieve 49.04%, 70.89% and 95.75& respectively. We discovered that sequence-to-sequence model sometimes fail to recreate the full input, which caused the performance is significantly increased when we no longer need the model to recreate the entire input in the sequence-to-word model. We were able to achieve the best performance with the multiple choice approach, where we limited the search space of the model by limiting the choices, however provide the right choices to the model can be a difficult problem. We further evaluated the multiple choice model using real world Weibo posts, and discovered that the irregularity of the texts can be very challenging to solve for both the model and native Chinese speakers.

For future work, we suggest to test the three approaches with larger transformers models, as larger model can likely capture more knowledge and patterns regarding head abbreviations. We also suggest to pretrain the transformers model on Weibo data to help with Weibo posts. And for further extension, we recommend look into incorporate other types of data such as images and videos as apart of the input, as on social media, texts are often accompanied by other types of data, which can be important for understanding the meaning of the texts.

Bibliography

[1] T. M. Fang, "'your mom is dead': The origins of the chinese internet slang nmsl," 2020. [Online]. Available: https://supchina.com/2020/04/23/nmsl-the-origins-of-the-chinese-internet-slang/#:~:text=If%20you%20followed%20the%20recent,%E2%80%9CYour%20mom%20is%20dead.%E2%80%9D

[2] S. Y. Chen, "From omg to tmd – internet and pinyin acronyms in mandarin chinese," *Language@Internet*, vol. 11, no. 3, 2014. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:0009-7-39804

[3] S. Wilson, W. Magdy, B. McGillivray, K. Garimella, and G. Tyson, "Urban dictionary embeddings for slang NLP applications," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4764–4773. [Online]. Available: https://aclanthology.org/2020.lrec-1.586

[4] J. Wallat, J. Singh, and A. Anand, "Bertnesia: Investigating the capture and forgetting of knowledge in bert," 2021.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[6] N. Thomas, "The fun guide to chinese number slang online," 2021. [Online]. Available: https://www.fluentu.com/blog/chinese/chinese-number-slang/

[7] C. Team, "What is may 20, 520 day holiday in china? meet chinese internet valentine's day," 2021. [Online]. Available: https://www.chinainternetwatch.com/7517/internet-valentines-day/#:~:text=520%20is%20a%20short%20form,Day%20every%20year%20to%20celebrate.

[8] D. Yang, Y.-C. Pan, and S. Furui, "Automatic chinese abbreviation generation using conditional random field." 01 2009, pp. 273–276.

[9] H. Zhang, "Chinese character input simple 'pinyin' implementation method and system," 2007. [Online]. Available: https://patents.google.com/patent/CN101079060A/en/

[10] Z. Wang, H. Liu, Y. Deng, and J. Xu, "Improvement of chinese input method based on standard keyboard," in *2010 Second International Workshop on Education Technology and Computer Science*, vol. 3, 2010, pp. 189–192.

[11] C. Donahue, M. Lee, and P. Liang, "Enabling language models to fill in the blanks," *CoRR*, vol. abs/2005.05339, 2020. [Online]. Available: https://arxiv.org/abs/2005.05339

[12] 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*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 154–158. [Online]. Available: https://aclanthology.org/W18-0516

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[14] 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*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703

[15] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," 2016.

[16] Y. Shao, Z. Geng, Y. Liu, J. Dai, F. Yang, L. Zhe, H. Bao, and X. Qiu, "Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation," 2021.

[17] D. Q. Nguyen, T. Vu, and A. Tuan Nguyen, "BERTweet: A pre-trained language model for English tweets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 9–14. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.2