# Assessing Fine-tuned LLMs for Chinese Idiom Translation

**Kexuan Zhang**
University of Toronto
kex.zhang@mail.utoronto.ca

**William Zhang**
University of Toronto
williamqd.zhang@mail.utoronto.ca

## Abstract

This paper presents a project that focused on fine-tuning the Davinci and M2M100 models to perform language translation task from Chinese idioms to English, aiming to capture the compact and underlying meanings of idioms, with a performance assessment using BLEU, METEOR, and COMET metrics. While the mathematical based matrices show a lack of performance for producing exact translations compared to references, COMET's scoring indicates the models' capability of obtaining a reasonable understanding of the idioms' semantic meanings. [1]

## 1 Introduction

With the advent of large language models like GPT-3, BERT, we have achieved remarkable performance in natural language processing. However, when it comes to the task regarding picking up the underlying meanings, most of the language models fail to maintain a good accuracy. One such example is the translation of idiom in Chinese language which has a compact representation (normally 4 characters), while contains deep-seated metaphorical meanings. Therefore, in this project, we fine-tuned two pre-trained models, Davinci and M2M100, to perform language translation that aims to pick up the underlying meanings of Chinese idioms, and then analyze the performance of each based on various metrics: BLEU, METEOR, and COMET. In this paper, we will provide a brief overview of related work, describe the methods we applied, analyze on the performance, and discuss about the findings and outline directions for future work.

## 2 Related Work

It has been decades since the advent of language translation industry, which has grown very mature and various models and related work have been well defined. In this project we kept building on these previous works, elaborated as follow:

### 2.1 Data

The data comes directly from the **PETCI** dataset [1] (*A Parallel English Translation Dataset of Chinese Idioms*). The raw dataset contains 4,310 Chinese idioms with 29,936 English translations. This means for each idioms it contains distinct translations from various sources. We are using the *filtered* version of the dataset which categorize translations into "gold", "human", and "machine", with "gold" referring to the first translation of each idioms in the Dictionary translations.

[1] Link to project code: https://github.com/KexuanZhang/Chinese-Idioms-Translation-with-Neural-Network

Preprint. Under review.

### 2.2 Pre-trained models

**Davinci** is part of the GPT-3 series of models developed by OpenAI. It is the most capable and expensive model compared to others (ada, babbage,curie) in its family [2]. It consists of a decoder-only transformer network and has an approximate of 175 billion parameters. **M2M100** is a multilingual machine translation model developed by Facebook AI. Unlike most machine translation models designed to translate between specific language pairs, M2M100 is a many-to-many model [3], meaning it can translate between any combination of the languages it was trained on, for more than 100 languages.

### 2.3 Metrics

We used three metrics to evaluate the performance of the models. **BLEU** is a commonly used metric for evaluating the quality of machine translation output. It measures the degree of overlap between machine-generated and target translations based on n-gram comparisons [4]. **METEOR** is another widely used metric for evaluating machine translation quality. METEOR computes a score based on unigram-recall, unigram-precision, and the ordering of the translation with respect to the reference [5]. METEOR shows promising performance in correlation value with Chinese-to-English dataset. **COMET** is a newer metric for machine translation. Unlike the previous two, COMET is based on neural framework, designed to predict human scoring of MT quality [6]. COMET can accurately predict how well a human expert would translate a text by comparing the meanings of different texts in detail. Thus, we expect it to demonstrate a better evaluation of our idiom translation.

## 3 Methods and Algorithms

### 3.1 Data preparation for fine-tune

To restrict our fine-tuning to only high-quality translations, for each idiom in the dataset, we extracted the "gold" quality translation and one "human" quality (the first one). That yields a total of 8108 rows of filtered data, where each is a matching of the idiom to one of its translations. Then, we split the data randomly into 64% train data, 20% test data, and 16% validation data. For the Davinci model, according to OpenAI's fine-tune documentation, we added a fixed separator to end of each prompt "->", a space at the start of each completion and a different separator "\n" at the end. The same is prepared for the validation data. Apendix A demonstrates the first three training data for Davinci fine-tune. For M2M100, we further split the training data into a source list of all Chinese idioms, and a target list of all the reference translations.Apendix B demonstrates the first three training data for M2M100.

### 3.2 Algorithm for Fine-tuning Davinci

We used OpenAI's fine-tune API to fine-tune their Davinci model. We invoked **openai.FineTune.create** with the training and validation data uploaded as *jsonl* file, and set the model='davinci'. The default batch_size is set to be 256, and num_epoch is set to 4, the other parameters are untouched. Once, the fine-tuned model is finished training, we retrieve the model_id using **openai.FineTune.retrieve**. To evalute the model using idioms from the test data, we use **openai.Completion.create** with max_tokens=15, and extract the first complete sentence returned by the request as the model's translation.

### 3.3 Algorithm for Fine-tuning M2M100

- **Model setup**: we load the pre-trained facebook/m2m100_418M conditional generation model and a corresponding tokenizer, specify the tokenizer source language to be "zh" (Chinese) and target language "en" (English)
- **Tokenizing data**: we define a **TranslationDataset** class to tokenize and store parallel corpora of the data with the tokenizer, and limit the maximum length to be 20. Then, we use **DataLoader** from PyTorch to group 16 batches and shuffle the data.
- **Set up training**: We employ an AdamW as optimizer, with pre-trained model parameters and learning rate $5e-5$. Then, we set up a scheduler from **torch.optim** to manage the learning rate. In our case, we use step_size 1 and gamma 0.9.

- **Training**: we fine tuned the model for 3 epochs. In each epoch, we iterate through the batches, generate outputs for each, calculate **CrossEntropyLoss**, and perform backward pass to update weights. After training, we save the model and tokenizer locally.
- **Testing**: during test time, we load the local fine tuned model; We encode and tokenize the input prompts; Lastly, we use the generate method of the model to generate translation, providing "en" (English) as **forced_bos_token_id** parameter to hard control output translation to be in English.

# 4 Experiments and Results

## 4.1 Metrics Evaluation

To compute BLEU and METEOR scores, both methods take in a "hypothesis" (translated text) and a list of "reference" (of minimum one string), then return a score between 0 to 1, with a higher score indicating better translation quality. To compute the COMET score, a pre-trained model **wmt22-comet-da**, is loaded first. Unlike BLEU and METEOR, the input for COMET includes the original Chinese text along with a single "hypothesis" and a single "reference"[6]. The model predicts the quality of the translation pairing and also returns a score between 0 and 1, with the higher the better, and 0 implying that the translation is not better than random chance.

## 4.2 Results and Observations

We took a full use of the remanent data to explore on the behaviors of the models when evaluating metrics. Specifically, the dataset contains one most precise translation from dictionary labeled **gold** (which the model is trained on); multiple other human translations that are less frequently used labeled **human**, and multiple machine generated translations from Google and DeepL labeled **machine** [1]. Apendix C demonstrates the first three entries of test set. Then, we separately calculated the metrics for each idiom with **only_gold**, **human_and_gold**, and **all** as references. Since COMET takes one string as reference, we therefore only use the **gold** set as references. After that, we take the average of the score of all idioms of each metrics as in the following table:

| | Davinci | | | M2M100 | | |
|---|---|---|---|---|---|---|
| | BLEU | METEOR | COMET | BLEU | METEOR | COMET |
| Gold | 0.0113212 | 0.1283618 | 0.5657916 | 0.0014125 | 0.0816444 | 0.4897104 |
| Gold_human | 0.0757820 | 0.3213828 | / | 0.0034567 | 0.1746898 | / |
| All | 0.0860249 | 0.3937279 | / | 0.0154059 | 0.3149239 | / |

Table 1: Model scoring on different metrics and dataset

From the table we can make following obervations:

- **For both models**, the score is quite low when evaluating translation with Gold reference, and the score improves as we include more human generated references, and improves even more when also include machine generated references.
- **For Davinci**, there is a huge increase in score from Gold to Gold_human, and a slightly increase from Gold_human to All for both BLEU and METEOR. **For M2M100**, there is a moderate improvement from Gold to Gold_human, while a intense improvement from Gold_human to All for BLEU. While the score for METEOR improve evenly through 3 datasets.
- **For comparison**, davinci obtains a better score compared to M2M100 for every metrics and dataset. For most of scores of BLEU metric, the score of Davinci is considerably higher (Some are even 10 times higher) than M2M100, and only the scores for METEOR are on a comparable scale.

# 5 Discussion

From the previous results, we could make several interesting observations and analyze on it.

- **Poor performance on Gold dataset**: Even though the models are trained on part of the Gold dataset, they still have a poor test time performance when getting unseen idioms. This can infer that it is indeed hard to pick up the underlying meaning of too compact and obscure language representations with insufficient data. And there is a high possibility for overfitting during test time for this task.
- **Huge boost in BLEU for Davinci**: The BLEU metric is measuring the form and positional similarity of two sentences [4], thus the more alike the two sentences, the higher the score. In addition, the Gold dataset provide translations that are more compact, abstract, and concise; the gold_human dataset contains multiple translations that use more casual and daily; lastly, the All dataset further includes the machine translations that have literal translations. Therefore, the high score achieved with Gold_human can demonstrates that Davinci indeed pick up the meaning of idioms, and its output translation has high similarity with human translations, which is the dominant terms in the dataset. However, it has a low form and positional similarity with gold translations. We could infer from this that Davinci does a good job picking up the meaning, but it is lack of ability to make an concise but expressive translation.
- **Difference in behavior of M2M100 and Davinci when including machine translations**: We can observe that Davinci's score for BLEU improves slightly, while M2M100 improves dramatically after including the machine translated data. The higher increase in score, meaning the more alike the model translation is to the machine translations. From this we can interpret that the fine tuning for Davinci is successfully differentiate its expressive ability from other translators, while M2M100 maintains a high similarity.
- **Reacquire comparability in METEOR**: In BLEU metric, we can see M2M100 loses badly to Davinci, in which the score for Davinci is about 10 times higher than M2M100. However, the METEOR scores for M2M100 return to a compatible level. In addition to positional similarity, METEOR also takes into account the different form of same words, synonymy, word order, and applying penalty for unaligned words [5]. From this we can infer that, even though M2M100 could have a poor performance in form and position similarity, it could still pick up the meaning and potentially expressing the same idea in different forms.
- **COMET score**: The original paper reports scores ranging from 0.4 to 0.6 [6]. While the raw score itself lacks interpretability due to its underlying structure of neural networks, the COMET score is useful for ranking different translations. We observe the same pattern that the Davinci model outperforms the M2M100 by a margin. Moreover, the COMET score has demonstrated a strong correlation [6] with human evaluations of translations, surpassing both BLEU and METEOR. This highlights the potential of the COMET score as a more reliable and accurate measure of translation quality that closely aligns with human judgement. The superior performance of the Davinci model suggests that it has a better understanding of the underlying meanings of idiomatic expressions.

## 6 Conclusion

In this paper, we evaluated the performance of two fine-tuned state-of-art language models on translating Chinese idioms, against several machine translation metrics. Between the two models, we observed that the performance can differ significantly depending on the dataset and metric used for evaluation. Davinci tends to perform better on the BLEU metric, while M2M100 shows more consistent improvement across all datasets on the METEOR metric, and both achieves a reasonable performance on the COMET metric.

Overall, the results indicate that fine-tuning pre-trained models for the task of picking up the underlying meanings of Chinese idioms can lead to reasonable performance, but there is still much room for improvement. Additionally, the limit in dataset size could have lead to overfitting and poor generalization to unseen idioms. Our future work aims to design strategies to pre-process the idioms and capture their metaphorical meanings before sending them to fine-tune, which could potentially improve the performance of the models.

## Contributions

- Kexuan Zhang: Fine-tuned the M2M100 model, prepared data, perform tests, and evaluate the model performance based on the 3 metrics. Contributed equally to the report.
- William Zhang: Fine-tuned the Davinci model, prepared data, perform tests, and evaluate the model performance based on the 3 metrics. Contributed equally to the report.

## References


[1] K. Tang, "Petci: A parallel english translation dataset of chinese idioms," *arXiv preprint arXiv:2202.09509*, 2022.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[3] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, *et al.*, "Beyond english-centric multilingual machine translation," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 4839–4886, 2021.

[4] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, July 2002.

[5] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, (Ann Arbor, Michigan), pp. 65–72, Association for Computational Linguistics, June 2005.

[6] R. Rei, C. Stewart, A. C. Farinha, and A. Lavie, "Comet: A neural framework for mt evaluation," *arXiv preprint arXiv:2009.09025*, 2020.

## A Appendix: Example training data for Davinci fine-tune

[{'prompt': '顾此失彼->', 'completion': ' take one into consideration to the neglect of the other\n'},
{'prompt': '大腹便便->', 'completion': ' potbellied\n'},
{'prompt': '赤地千里->', 'completion': ' a thousand li of barren land - a scene of utter desolation\n'}]

## B Appendix: Example training data for M2M100 fine-tune

source list: ['街谈巷议</s>', '摇身一变</s>', '斩钉截铁</s>']
target list: ['street gossip</s>', 'suddenly changed</s>', '"chop up nail and cut through iron"</s>']

## C Appendix: Example data for Gold, Gold_human, and All sets

[{'chinese': '一般见识',
'gold': 'lower oneself to the same level as somebody else',
'gold_human': ['lower oneself to the same level as somebody else'],
'all': ['lower oneself to the same level as somebody else', 'General knowledge', 'General Insight']},

{'chinese': '一本万利',
'gold': 'make big profits with a small capital',
'gold_human': ['make big profits with a small capital', 'a small investment brings a ten thousand-fold profit', 'highly profitable'],
'all': ['make big profits with a small capital', 'a small investment brings a ten thousand-fold profit', 'highly profitable', 'a profit', 'All in one', 'A million dollars']},

{'chinese': '一土',
'gold': 'mere dust heaps',
'gold_human': ['mere dust heaps', 'only a clod of yellow earth', 'a grave', 'something utterly insignificant'],
'all': ['mere dust heaps', 'only a clod of yellow earth', 'a grave', 'something utterly insignificant', 'a handful of loess', 'A shovelful of yellow earth', 'One shovelful of earth', 'A shovelful of earth', 'One shovelful of dirt']}]