Using GPT-4 for conventional metaphor detection in English news texts

Liang, J.; Dorst, A.G.; Prokic, J.; Raaijmakers, S.A.

Computational Linguistics in the Netherlands Journal 14 (2025) 307-341 Submitted 12/2024; Published 07/2025

Using GPT-4 for Conventional Metaphor Detection in English News Texts

Jiahui Liang[1*] J.H.L.JIAHUI@HUM.LEIDENUNIV.NL
Aletta G. Dorst[1] A.G.DORST@HUM.LEIDENUNIV.NL
Jelena Prokic[*△] J.PROKIC@HUM.LEIDENUNIV.NL
Stephan Raaijmakers[1,2*] S.A.RAAIJMAKERS@HUM.LEIDENUNIV.NL

[1] *Leiden University Centre for Linguistics (LUCL), Leiden, Netherlands*

[2] *Nederlandse Organisatie voor toegepast-natuurwetenschappelijk onderzoek (TNO), The Hague, Netherlands*

Abstract

Metaphor detection presents a significant challenge in natural language processing (NLP) due to the intrinsic complexity of metaphors. In this work, we apply a prompting approach to evaluate GPT-4's performance on the conventional metaphor identification task. We specifically investigate the effects of prompt variation, output stability, and the role of n-shot prompting. The results indicate that GPT-4's performance on the metaphor identification task is consistently low across all tested settings, significantly lagging behind the top-performing BERT model. Based on our findings and error analysis, we propose possible approaches for utilizing LLMs and AI assistants in metaphor detection and analysis.

1. Introduction

Since the cognitive turn of the 1980s, metaphors are no longer seen as mere decorative devices or instances of deviant language use. They are recognized as a fundamental cognitive tool in human understanding and communication. Metaphors allow us to think and talk about abstract, complex and unfamiliar concepts, such as time or the economy, in terms of more concrete, simple and familiar ones, such as physical space, movement or living entities. For example, we say that something happened 'in' 2024 or 'between' 2020 and 2023, that our holidays 'flew by so fast', or that prices are 'soaring' while zhe or 'flourishes'. Lakoff and Johnson's (1980, 1999) groundbreaking work showed that such metaphorical uses of words and phrases – e.g. 'in' and 'withers' – form systematic patterns in our everyday language use because they are the linguistic realizations of underlying conventional conceptual metaphors – 'in 2024' → TIME IS SPACE and 'the economy withers' → THE ECONOMY IS A PLANT / LIVING ORGANISM. Since most of the metaphors we use are conventional both in language and thought, we normally use and understand them automatically and effortlessly, without even realizing that they are metaphors.

Corpus research has since confirmed that linguistic metaphors are indeed ubiquitous in everyday discourse, occurring on average every seven to eight words (Steen et al. 2010a, Steen et al. 2010b), although statistically significant differences were observed between the registers included in the VU Amsterdam Metaphor Corpus (Steen et al. 2010c), with the number of metaphorically used words ranging from 18.6% in academic discourse and 16.4% in news to 11.9% in fiction and 7.7% in casual conversation (Dorst 2011, Herrmann 2013, Kaal 2012, Krennmayr 2011). Moreover, a great number of studies analyzing authentic language use have shown that the linguistic forms and rhetorical functions of metaphor in discourse vary considerably across domains, genres and communicative settings, from education (Cameron 2003, Low 2008), politics (Ahrens 2009, Musolff 2004), business (Koller 2004) and health communication (Semino et al. 2018) to advertising (Forceville 1996), and many more.

This combination of conventionality and ubiquity entails that identifying linguistic metaphors in authentic discourse quickly becomes both a complex and labor-intensive endeavor, especially when researchers want to employ a reliable and replicable method for metaphor identification such as MIP (Pragglejaz Group 2007) or MIPVU (Steen et al. 2010b). For projects interested in identifying all possible metaphorical language in larger datasets, rather than focusing on specific lexical items or semantic domains, manual identification is often not a viable option, especially for researchers working individually and on limited research time. And for metaphor scholars working in teams, consistency and inter-annotator reliability remains an issue and additional coding protocols often have to be established (Dorst et al. 2013). It is therefore not surprising that metaphor scholars have been especially interested in hearing whether such identification principles as those formulated by MIP and MIPVU can be turned into computational rules allowing for automated metaphor detection. As pointed out by Shutova (2015, p. 580-1), "computational work on metaphor evolved around the use of hand-coded knowledge and rules to model metaphorical associations, making the systems hard to scale", but recently, there has been a "growing interest in statistical modeling of metaphor [...] with many new techniques opening routes for improving system accuracy and robustness."

The emergence of large language models (LLMs) creates new possibilities for metaphor detection and sub-type labelling. Recent research indicates that LLMs demonstrate superior performance in contextual semantic comprehension compared to previous generations of language models (Zhou et al. 2023). Prompting (or in-context learning) approaches appear useful techniques for the application of LLMs to NLP tasks (Chung et al. 2022, Ding et al. 2023a). Building on this foundation, the current paper explores the performance of GPT-4 in detecting conventional linguistic metaphors in news texts, specifically utilizing the news subcorpus of the VU Amsterdam Metaphor Corpus (Steen et al. 2010c) and applying a prompt-based methodology. Our evaluation of the results of prompt-based metaphor identification offers new insights into the usefulness of LLM architectures for this task and reflects on the different types of metaphor that can be detected automatically and the effect of the formulation of the prompt on the metaphors identified and the readability of the output.

2. Related Work

2.1 The Evolution of Metaphor Detection Models In Recent Years

The development of metaphor detection has advanced alongside technological progress through several phases, including rule-based, statistical, and neural approaches. This progression paves the way for new exploratory trends:

Deep learning-based approach: Recent approaches of transformers-based architectures particularly emphasize a fine-tuning approach with pre-trained contextual language models such as Bidirectional Encoder Representations from Transformers (BERT), which, with its bidirectional attention mechanism, effectively captures context to distinguish between literal and metaphorical meanings (Chen et al. 2020a, Dankers et al. 2020, Liu et al. 2020, Su et al. 2021, Choi et al. 2021, Zhang and Liu 2022) and its variants, including RoBERTa (Gong et al. 2020, Babieno et al. 2022, Ge et al. 2022, Li et al. 2023, Elzohbi and Zhao 2023, Uduehi and Bunescu 2023, Wang et al. 2023), SemBERT, ALBERT (Li et al. 2020), which improve BERT by refining the learning of contextual relationships, incorporating additional semantic information, or enhancing efficiency through methods like parameter sharing and factorized embeddings, thereby improving performance on various downstream tasks, including metaphor detection. In addition, there is also research that uses Generative Pre-trained Transformer (GPT) (Wachowiak and Gromann 2023) to involve its generative capabilities to identify metaphor mappings, as well as other architectures like XLNet (Liu et al. 2020), which can capture richer bidirectional context, thus improving the effectiveness in metaphor detection.

LLMs-based approach: Compared with the traditional models above, LLMs demonstrate notable potential in language understanding, including the ability to incorporate sociocultural con-

texts and engage with multimodal data, which attracts increasing scholarly interest in exploring their application for metaphor detection and understanding. Hicke et al. (2024) demonstrate how LLMs, guided by annotation-based prompts, can effectively identify conceptual metaphors, enabling large-scale computational investigations. Jia et al. (2024) tackle challenges like data scarcity and inference costs with Curriculum-style Data Augmentation, achieving notable improvements in open-source LLM fine-tuning. Furthermore, Tong et al. (2024) introduce the Metaphor Understanding Challenge Dataset (MUNCH) to evaluate LLMs across diverse metaphorical contexts. Yang et al. (2024) enhance verb metaphor detection by integrating ChatGPT's tacit knowledge with entailment analysis, while Wang et al. (2024) propose a multi-stage prompting framework for analyzing Chinese metaphors.

2.2 Methodology of Metaphor Detection In Recent Years

Linguistic theory-based approach: Beyond traditional linguistic feature-based approaches, two main linguistic theories have been incorporated into the theoretical framework design of the models for improving metaphor detection: the Metaphor Identification Procedure (MIP) (Pragglejaz Group 2007), which compares a word's basic and contextual meanings to detect metaphors, providing a reliable and validated guideline with high inter-annotator agreement for identifying metaphorical expressions, making it widely adopted in the enhancement of metaphor detection (Choi et al. 2021, Song et al. 2021, Li et al. 2023, Wang et al. 2023); and conceptual mapping (Lakoff and Johnson 1980), which provides a cognitive framework for understanding metaphors as mappings from a source to a target domain. This theory has informed model designs by incorporating domain-specific semantic features and aligning embeddings across conceptual domains. For instance, models have used conceptual mappings to improve metaphor classification by learning transferable representations between abstract and concrete domains (Wan et al. 2020, Tian et al. 2024).

Multi-task learning-based approach: Multi-task learning optimizes multiple related tasks simultaneously, enabling shared knowledge to improve overall performance (Caruana 1997). Chen et al. (2020b) enhance metaphor detection using auxiliary tasks like idiom detection and out-of-domain metaphor annotation. Xu et al. (2024) develop a multi-modal framework with Chain-of-Thought reasoning and modality fusion for metaphor detection in memes. Zhang and Liu (2023) transfer knowledge from basic sense discrimination to address data scarcity in metaphor detection. Badathala et al. (2023) jointly model hyperbole and metaphor detection, making use of their linguistic similarity. Le et al. (2020) integrate graph convolutional networks and word sense disambiguation for metaphor detection. Mao and Li (2021) introduce a novel gating mechanism for task-specific information sharing. Dankers et al. (2019) show mutual benefits between metaphor and emotion detection through joint learning. Song et al. (2024) use syntax-aware attention and contrastive learning to enhance metaphor sensitivity, while Lai et al. (2023) unify figurative language detection across languages and figures of speech with a multilingual framework.

2.3 Advantages and Limitations of Current Metaphor Detection Approaches

Metaphor detection has witnessed significant advancements in recent years. Taking F1 score as a benchmark, which balances the model's capability to avoid falsely labeling non-metaphors as metaphors and capability to capture as many true metaphors as possible, results from the VUA Metaphor Detection Shared Tasks show that in 2018, the highest F1 score reached 65.1, whereas by 2020, over half of the competing models outperformed this benchmark, with the highest F1 score rising to 76.9 (Leong et al. 2018, Leong et al. 2020). More recently, this score has been further pushed to 79.4 (Zhang and Liu 2022).

Despite these achievements, several challenges and research gaps remain:

Lack of Integration of Sociocultural Knowledge: Sociocultural background knowledge is critical for constructing metaphorical meanings. However, existing models struggle to incorporate

such knowledge effectively into the training process, leading to an incomplete understanding of metaphor semantics, particularly in culturally specific contexts.

Dataset Limitations and Generalization Issues: Current metaphor detection models rely heavily on a limited set of metaphor corpora, restricting their generalization capabilities. The evaluation of model performance on unseen texts remains a significant challenge, and it is unclear how effectively these models perform outside of the training datasets.

Limited Focus on Conventional Metaphors: Conventional metaphors account for 99% of all metaphors, making their automatic detection essential for identifying metaphors in general. However, research specifically focused on detecting conventional metaphors remains underexplored. Compared to general metaphor detection, conventional metaphor detection requires an additional step—determining whether a metaphorical meaning has become conventionalized. This is often assessed through its frequency, stability, and, in some cases, its inclusion in dictionaries. Despite the prevalence of conventional metaphors, studies in this area are relatively sparse, with long gaps between major contributions, indicating a lack of sustained academic attention and integration with recent advances in computational modeling. Early contributions include Mason's (2004) CorMet, a corpus-based system for extracting conventional metaphors, and Wilks' (2013) algorithm for detecting metaphors embedded in word senses within lexical databases like WordNet. While these methods expanded metaphor detection to deeper semantic structures, they did not directly address the challenge of identifying whether a metaphor is conventionalized. Later, Levin et al. (2014) provided an overview of resources for detecting conventional metaphors across multiple languages, and Maudslay et al. (2022) introduced the first model for metaphorical polysemy detection (MPD), which identifies conventionalized metaphorical senses in the English WordNet. Despite these contributions, research in this area has progressed slowly, with significant time gaps between major developments

Recent Advances in LLMs: Recent work has explored the potential of LLMs in metaphor detection, achieving promising results through various approaches. For instance: prompting approaches, curriculum-style data augmentation, creating multimodal metaphor datasets based on model capabilities, implicit knowledge analysis, etc.. While these studies highlight the potential of LLMs for metaphor detection, they primarily focus on general metaphor detection tasks. The automation of conventional metaphor detection, in particular, remains underexplored. Addressing this gap, along with issues of dataset limitations, sociocultural knowledge integration, and generalization, represents an important direction for future research.

2.4 LLMs for Metaphor Detection

The emergence of LLMs has revolutionized the field of Natural Language Processing (NLP), with ongoing research focused on the evaluation of LLMs for diverse NLP tasks.

In the metaphor detection task using LLMs, the variations in prompting settings can be broadly categorized into two types. The first type is direct detection, which involves using prompts to instruct LLMs to identify metaphors within a given text. This approach utilizes the models' ability to generalize metaphorical patterns from extensive training data, allowing them to detect metaphors across diverse contexts. In this project, we focus on exploring different linguistic formulations of prompts as one-time inputs to generate model outputs, rather than incorporating interactive or iterative dialogue.

The second type is Socratic prompting, which is a reasoning-based approach used to determine whether LLMs rely on metaphorical interpretation (Qi et al. 2023). One relevant line of research focuses on evaluating the metaphor understanding abilities of LLMs. For instance, Tong et al. (2024) proposed an evaluation framework to assess whether models genuinely understand metaphors or merely rely on superficial lexical similarity. However, research on metaphor detection, such as that by Bastian et al. (2024), has employed indirect detection methods. These methods do not require

models to directly identify metaphors; instead, they use guiding questions to observe the models' reasoning processes, thereby assessing their ability to recognize and understand metaphors.

Given the challenges and limitations of existing metaphor detection models, our research will evaluate the performance, robustness, and generalization of GPT-4 in detecting conventional metaphors. Since traditional metaphor detection tasks and large-scale human-annotated metaphor corpora, such as VUAMC, adopt a per-token labeling approach, we follow this paradigm to ensure consistency with prior work and facilitate direct comparison. Our study aims to assess the effectiveness of GPT-4 in this framework while also exploring appropriate evaluation methods.

In recent years, though research on LLM-based labeling for metaphor detection has been limited, LLMs have demonstrated strong potential in data annotation tasks and have been successfully applied to various per-token labeling tasks, such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging (Tan et al. 2024). Studies suggest that while causal masking may constrain the performance of decoder-based models in sequence labeling, optimizing decoding strategies can enable LLMs to achieve near-state-of-the-art performance in information extraction tasks (Dukić and Šnajder 2024). Additionally, research on NER has shown that well-designed prompts and post-processing methods can improve per-token annotation quality, as evidenced by the zero-shot NER inference method proposed by (Xie et al. 2023) and the llmNER toolkit's optimizations for token boundary alignment and error correction (Villena et al. 2024). In POS tagging, LLMs have also exhibited reliable per-token prediction capabilities, with a multilingual model based on the Universal Dependency framework achieving up to 90% accuracy (Machado and Ruiz 2024). Furthermore, fine-tuned LLMs have demonstrated high F1 scores in NER and POS tasks for low-resource languages (Subedi et al. 2024). While challenges such as token alignment errors remain, existing studies validate the feasibility of per-token labeling with LLMs. Therefore, we adopt this approach for metaphor annotation, implementing rigorous quality control measures to enhance reliability.

3. Experiment

During the preparation of the experiment, an initial series of Q&A sessions was conducted with GPT-4, including the definition of conventional metaphors, examples of conventional metaphors and their interpretation, and metaphor classifications, to examine its understanding of conventional metaphors in its default setting. The outputs indicate that GPT-4 already possesses a basic understanding of conventional metaphors. Based on these initial findings, the experiment continued as follows, with the overall workflow illustrated in Figure 1.

3.1 Data Preparation

Data preparation for model prompting typically involves two key components: a metaphor corpus with manual annotations, which serves as ground truth for evaluating model performance, and a set of prompts that function as queries for the model.

Conventional metaphor corpus. For the current study, we used the news subcorpus (44,792 words) of the VU Amsterdam Metaphor Corpus (VUAMC; 186,688 words) (Steen et al. 2010c). As pointed out by Tong et al. (2021, p. 4676), VUAMC is "the only metaphor corpus used in studies of automated metaphor identification that is built by cognitive linguists, and the only one that deals with the metaphoricity of function words." All news texts (n=46) in VUAMC were sampled from the BNC-Baby corpus, one of the sub-corpora of the 100-million-word British National Corpus (BNC), consisting of four sets of samples, each containing one million POS-tagged words. However, the VUAMC annotations include all linguistic forms of metaphor: both metaphor proper (or 'indirect metaphor-related words') and simile (or 'direct metaphor-related words), and both non-deliberate metaphors (i.e. the type of conventional metaphor we all use routinely and automatically) and deliberate metaphors (i.e. novel, original, creative, extended, signalled, etc.). Since our current focus is on the detection of pervasive conventional metaphors in discourse, we used the additional

deliberateness annotations by Reijnierse et al. (2018) to exclude all deliberate metaphors from our dataset.

Secret held-out data. Given that GPT-4 was trained on datasets from the Internet including books, articles, websites, and social media (Baktash and Dawodi 2023, Qiu 2023), we cannot rule out that the news texts included in the VUAMC, which were sampled from BNC-Baby and may have appeared in online news archives, are part of the training data for GPT-4. To address this issue, two team members trained and used MIPVU to annotate four news articles, thereby generating a secret, held-out dataset for performance comparison. It should be noted that these texts may also be present in the training data for GPT-4, as they are publicly available news texts online; however, they had not been previously annotated for linguistic metaphor and are not part of any annotated dataset.

Dataset	**#sentences**	**#tokens**	**#M**	**%M**
Conventional_metaphor_corpus	2277	44957	6904	15.36%
Secret_held_out_data	34	870	125	14.37%

Table 1: Statistics for the annotated datasets: #sentences is the total number of sentences; #tokens is the total number of tokens; #M is the total number of metaphoric tokens; %M is the percentage of metaphoric tokens

Prompt set. Previous research has shown that prompt variation can significantly affect the output of LLMs (Gonen et al. 2022, Gu et al. 2022, Fernando et al. 2023). In particular, tiny adaptations in the wording of prompts (e.g. different punctuation, or choice of words) have been found to lead to entirely different results (Mizrahi et al. 2023).

Type	**#number**	**%percentage**
Simple instruction	14	56%
Background information	6	24%
Adaptation of MIP	5	20%

Table 2: Statistics of the prompt set. #number is the number of prompts of a prompt type; %percentage is the prompt type percentage over the prompt set

We developed a collection of 25 different prompts, which are automatically generated by GPT-4 and optimized under manual review. These prompts are divided into three distinct types:

1) Simple Instruction Prompts, which directly describe the task without any additional information or background, aiming to observe the model's output capabilities under the guidance of the most basic information.

2) Prompts with Background Information, which introduce definitional information such as traditional metaphors, to test the impact of extra contextual information on the accuracy and relevance of the content generated by the model.

3) Prompts adapted from the MIP, investigating whether the model can enhance its performance by replicating a manual metaphor identification process.

3.2 Experimental Setup

The project integrates the **gpt-4-1106-preview** model through an API. This model has a context window of 128k, allowing for the equivalent of over 300 pages of text within a single prompt. Its train data has been updated up to April 2023 (OpenAI 2023). Our implementation is available at https://github.com/Jiahui84/Conv_met_detection, ensuring transparency and reproducibility.

The default setting of the model is used for the experiment, with the temperature, or randomness, set at 0.7, which balances certainty and randomness, providing moderately diverse outputs. Additionally, to account for the randomness inherent in the generation results of the model mentioned above, the entire experiment process is repeated three times, and the results are averaged to obtain more reliable and stable results.

The prompting experimental settings include three stages:

1) **Prompt optimization:**

   The prompts were initially automatically generated by ChatGPT based on predefined criteria, including types, lengths, and quantities, to ensure diversity. Each prompt underwent preliminary validation on randomly sampled examples (from the example set) to confirm the output adhered to the specified labeling schema and formatting by a team member. After initial testing, further optimization was conducted under the review of the whole team, specifically examining the impact of keyword selection and labeling schema selection, which are described below.

   Keyword selection. For this stage, we investigate whether keywords used in prompts and labelling schemas can have impact on the model performance. We tested the same prompts using different keywords: "conventional metaphor" "lexicalized metaphor" and "metaphor" to see if the keywords affected model performance. However, after comparing and evaluating the results (F1-score, precision, recall), we found that the differences among the three were not significant. Therefore, "conventional metaphor" is chosen for its more stable and better overall performance.

   Labeling schema selection. To assess the impact of different labeling schemas on model performance, we experimented with multiple output labeling strategies in the prompts. The Inside-Beginning-Outside (IBO) schema was used to annotate whether each word in a sentence was part of a metaphorical expression: B for the beginning of a metaphor, I for inside a metaphor, and O for outside a metaphor. We also tested a binary labeling schema (1/0), where a word was annotated as 1 if it was a conventional metaphor and 0 if it was not, as well as a Yes/No labeling schema. The results showed no significant performance differences between these latter two approaches, while the IBO schema produced less reliable outputs. Given these findings, we adopted the 1/0 labeling schema for its convenience in statistical analysis. The generated outputs were manually reviewed based on the guidelines in Appendix 7.5 to ensure validity and consistent tokenization. After verification, the outputs were converted into CSV format and evaluated using Python statistical tools alongside human-labeled data.

2) **Zero-shot prompting:**

   Inspecting what the model does “out of the box” when applied to the task of conventional metaphor detection.

3) **N-shot prompting (N>0):**

   Providing N (1, 5, 10) examples with word-level labels indicating whether a word is a conventional metaphor. Each example consists of one sentence and corresponding word list with conventional metaphors annotated with 1.

The entire experiment is segmented into two stages: Training and Testing, with the prompting settings embedded within each stage. Consequently, the corpus data is split into four parts according to predefined percentage allocations: train set, develop set, test set, and example set, as shown in Table 3. The example set refers to a subset of sentences used in N-shot prompting, where N groups of sentences are paired with annotations of words in those sentences. These annotated examples are presented to the model first to guide its understanding of the task. The remaining sentences are

then provided for the model to annotate independently. Using F1 score as a benchmark, during the training phase, outputs from 25 prompts applied to the training dataset were obtained to identify the prompt with highest F1 score. Subsequently, in the developing phase, this prompt is further evaluated with different parameter combinations (e.g., temperature) in develop set to find out the best parameter combination which can achieve the highest F1 scores. These stages of the experiment are repeated on a secretly held-out dataset to validate the results and ensure the generalizability of the findings.

Dataset	**#sentences**	**#tokens**	**#M**	**%Corpus**
Conventional_train	199	4075	682	9.09%
Conventional_train_example	16	293	48	0.65%
Conventional_develop	213	4005	561	8.94%
Conventional_develop_example	16	250	34	0.56%
Conventional_test	1821	35836	5453	79.95%
Conventional_test_example	16	363	60	0.81%
Secret_test	34	870	125	100%

Table 3: Data split. #sentences is the number of sentences; #tokens is the total number of tokens; #M is the total number of metaphors; %Corpus is the percentage of the total tokens of the total corpus

In the testing phase, the best-performing combination of prompt and parameter, identified using 9.09% of the data (training set), is applied to the remaining 79.95% (testing set) to validate its effectiveness. This step ensures that the selected prompt and parameter combination achieves consistent performance across the entire corpus.

3.3 Evaluation

In the evaluation phase, the results of three repeated experiments across 0-N shot prompting have been analyzed in terms of F1-score, precision, and recall. This analysis aims to determine the best performance that GPT-4 can achieve in the conventional metaphor detection task.

Given that the output contains a large number of non-conventional metaphor labels, which are not the focus of this project, we report performance separately for metaphors (positive labels) and non-conventional metaphors (negative labels). The evaluation focuses on assessing GPT-4's performance in detecting conventional metaphors, specifically measuring F1-score, precision, and recall across 0-N shot prompting. This approach allows for a more targeted evaluation of the model's capabilities in the conventional metaphor detection task.

Additionally, an error analysis was conducted to examine the true positive (TP), false positive (FP), and false negative (FN) rates for different word categories across various shots. This analysis helps identify any patterns, such as whether GPT-4 performs better at detecting metaphors in certain word categories or shows significant improvement with N-shot prompting. And the McNemar test is used to compare the classification results of two versions of output on the same dataset, determining if there is a statistically significant difference. In our case, the McNemar test will focus on two comparisons: first, the classification results of the same prompt at different shots prompting; second, the classification results of different prompts at the same shot prompting.

4. Results

The results of our metaphor detection study are presented in three sections: 0-shot prompting, N (1, 5, 10)-shot prompting and secret held-out data as comparison. The performance report includes an evaluation of the models in terms of precision, recall, and F1-score. The scores with the best performances across all models are indicated in bold.

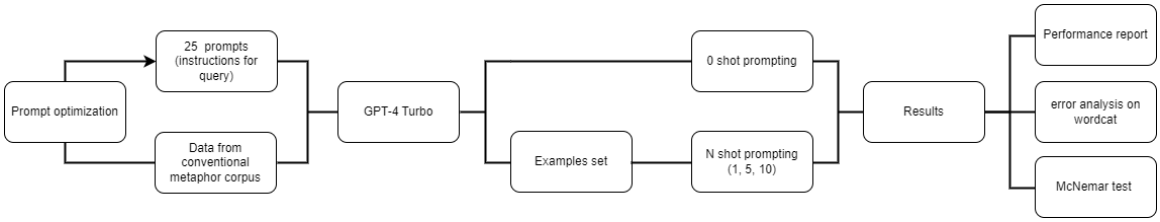

Figure 1: The workflow for metaphor detection using GPT-4 Turbo

4.1 Zero-shot Prompting

As shown in Table 12, the overall performance of the prompts for 0-shot prompting varied considerably, with substantial differences in F1 score, recall, and precision, showing ranges of 15.42%, 23.02%, and 32.36% respectively. This highlights that the choice of prompt had a substantial impact on performance during this phase. Using the F1 score as the benchmark, which balances the proportion of correctly detected conventional metaphors by the model (precision) and the proportion of annotated conventional metaphors in the corpus that were detected (recall), the best performance was observed with prompt 22 adapted from MIP, achieving an F1 score of 22.92%. Conversely, the worst-performing prompt was prompt 13, which featured a simple instruction, with an F1 score of 7.53%.

Analyzing the performance by prompt type, simple instruction prompts showed higher precision but performed poorly in F1 score and recall. This indicates that while these prompts accurately identified some conventional metaphors, they missed many others. Prompts with background information demonstrated balanced performance across all metrics, achieving higher overall effectiveness. Prompts adapted from MIP had relatively better F1 scores and recall, but their precision fluctuated considerably, suggesting a strong capability in identifying conventional metaphors but with a higher likelihood of recognizing additional, potentially irrelevant instances.

4.2 N (1, 5, 10)-shot Prompting

Figure 2 presents the distribution of F1 scores across different shot settings (0, 1, 5, and 10 shots). The boxplot reveals clear variations in performance, showing that while 1-shot prompting generally improves over 0-shot, further increasing the number of shots does not lead to consistent improvements.

In the 0-shot setup, the median F1 score is 0.161, with an interquartile range (IQR) spanning from 0.1215 to 0.1999. The scores exhibit notable dispersion, with a minimum value of 0.0567 and a maximum of 0.2574, indicating variability among different prompts. After introducing a single example in 1-shot prompting, the median increases to 0.2381, and the IQR shifts upward (0.2091 to 0.273). The maximum F1 score also reaches 0.324, exceeding that of the 0-shot setting. These results suggest that providing just one example can improve the model's performance in metaphor detection. The most pronounced improvements are observed in prompt 13 (+20.2%), prompt 18 (+18%), prompt 5 (+17%), and prompt 4 (+16.1%), while some prompts, such as prompt 7 and prompt 10, show only marginal gains. Adapted-from-MIP prompts, which performed well in the 0-shot setting, exhibit comparatively smaller improvements.

However, the trend does not continue when increasing the number of shots. In the 5-shot setting, the median F1 score decreases to 0.1964, with an IQR of 0.1766 to 0.2267. The 10-shot setting shows a similar trend, with a median of 0.2042 and an IQR of 0.1825 to 0.2315. The overall score range remains relatively stable, but the maximum F1 scores in these settings (0.2792 for 5-shot and 0.2883 for 10-shot) do not surpass those observed in 1-shot prompting. Additionally, both 5-shot and 10-shot prompting exhibit lower minimum scores (0.1099 and 0.109, respectively), suggesting that

Prompt	#F1	#Recall	#Precision	#Feature
1	0.0999	0.0544	**0.6264**	simple instruction
2	0.1209	0.0671	0.6159	simple instruction
3	0.1719	0.1143	0.4593	simple instruction
4	0.1426	0.0815	0.5699	simple instruction
5	0.0750	0.0399	0.6208	simple instruction
6	0.1063	0.0593	0.5119	simple instruction
7	0.2125	0.1353	0.4987	simple instruction
8	0.1588	0.0937	0.5227	background information
9	0.1344	0.0776	0.5036	simple instruction
10	0.1759	0.1037	0.5793	simple instruction
11	0.1290	0.0727	0.5752	simple instruction
12	0.1365	0.0782	0.5363	simple instruction
13	0.0753	0.0405	0.5426	simple instruction
14	0.1729	0.1054	0.4873	simple instruction
15	0.1727	0.1059	0.4672	simple instruction
16	0.1352	0.0771	0.5502	background information
17	0.1915	0.1165	0.5386	background information
18	0.1105	0.0616	0.5467	background information
19	0.2012	0.1259	0.5001	background information
20	0.1941	0.1226	0.5121	background information
21	0.2111	0.1320	0.5299	adapted from MIP
22	0.1700	0.1176	0.4301	adapted from MIP
23	**0.2292**	**0.1525**	0.4690	adapted from MIP
24	0.1887	0.1181	0.4857	adapted from MIP
25	0.2234	0.1492	0.4446	adapted from MIP

Table 4: Performance Metrics by Prompt Type in 0-Shot Prompting

adding more examples does not necessarily improve performance and may, in some cases, introduce inconsistencies.

These results indicate that while 1-shot prompting provides a moderate improvement over 0-shot, additional examples beyond this do not consistently enhance performance. The plateau observed in 5-shot and 10-shot prompting suggests that the effectiveness of in-context learning may depend more on prompt structure and content than on the number of examples alone. This finding aligns with previous observations that adapted-from-MIP prompts, which initially performed well in the 0-shot setting, do not show substantial improvements in few-shot scenarios. Simple instruction prompts exhibit greater variation in performance, with some benefiting from 1-shot prompting, while others show little change.

Overall, these findings highlight the role of prompt design in metaphor detection. While a small number of examples can be beneficial, increasing the number does not necessarily lead to further improvements and may introduce inconsistencies. This suggests that the impact of in-context learning in metaphor identification depends on factors beyond just the number of examples provided.

4.3 Comparison with Majority-class Baselines

Given that conventional metaphors constitute a significant proportion of metaphor stimuli datasets (Steen et al. 2010c), we compared the results from our experiment across all prompting stages from 0-shot to n-shot. We selected the highest F1 score, which appeared in the 1-shot prompting stage under Prompt 4. Therefore, we chose this score to compare with the majority-class baseline to evaluate our

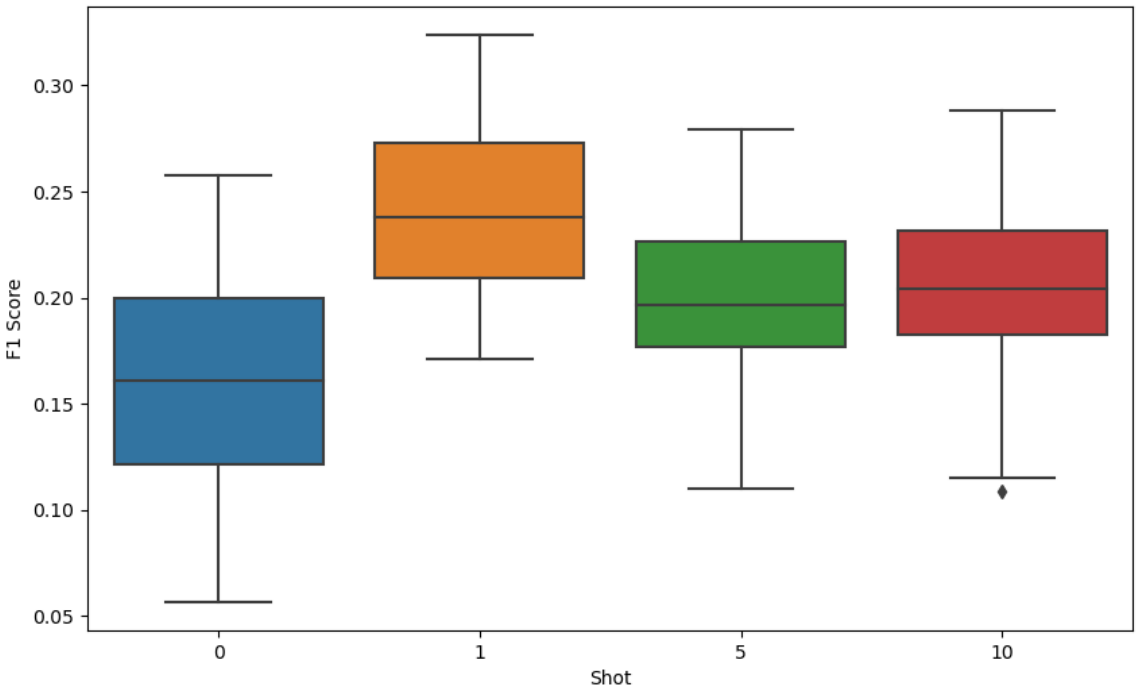

Figure 2: Comparison of F1 Score Progression from 0-shot to N (1, 5,10)-shot prompting in corpus data

experimental performance and to benchmark against state-of-the-art metaphor detection models, as shown in Table 5. All scores presented below are based on the VUA_News dataset. Since all the compared models were trained and tested on the VUAMC dataset, they all perform token-level metaphor detection, which increases the comparability of the results.

#Model	#Precision	#Recall	#F1
RoBERTa_SEQ	82.2	74.1	77.9
MSW_BASE	82.2	76.1	79.0
MisNet	**82.6**	**77.0**	**79.7**
GPT-4	55.2	21.1	30.4

Table 5: Performance of different models on the VUA_News dataset for metaphor detection

It is important to note that this study focuses on conventional metaphor detection, whereas the baseline models in the table are all general metaphor detection models. Although conventional metaphors account for a substantial proportion of metaphor data, direct comparisons may have certain limitations. Additionally, while the baseline models used the entire news corpus for training and testing data splits, our scores were derived from a randomly sampled 10% subset of the VUA_News dataset. As a result, there is a proportional difference, meaning that although all scores originate from the same corpus, the dataset content is not entirely identical.

Finally, due to the imbalance between metaphor and non-metaphor labels, our scoring method calculates the scores separately for correct labels and negative labels. However, the F1 scores presented in the table are based solely on the metaphor labels. In contrast, the baseline models did not make this distinction and calculated scores based on all labels.

Taking the above differences into account, the table shows that RoBERTa_SEQ, MSW_BASE (Babieno et al. 2022), and MisNet (Zhang and Liu 2022) all perform well, with F1 scores ranging

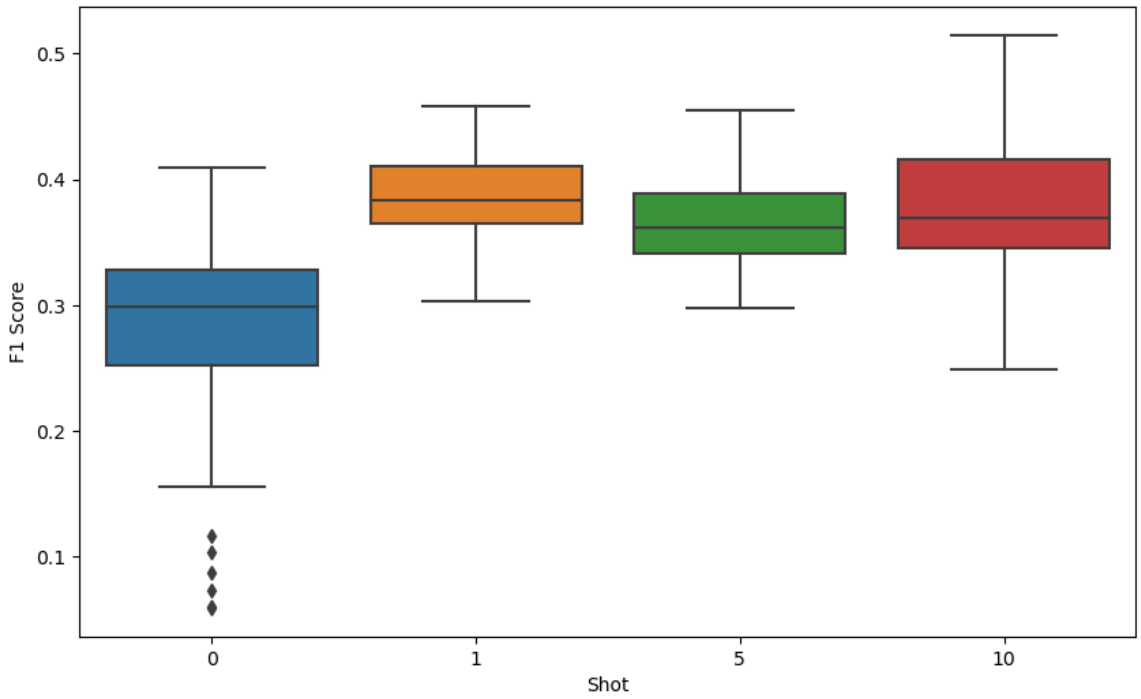

Figure 3: Comparison of F1 Score Progression from 0-shot to N (1, 5,10)-shot prompting in secret held-out data

from 77.9 to 79.7. Among them, MisNet achieves the highest F1 score of 79.7, indicating its strong generalization ability in metaphor detection tasks. In contrast, GPT-4 performs significantly worse in this experiment, with an F1 score of only 30.4, far below the other models.

5. Evaluation

5.1 Secret Held-out Data as Comparison

To assess whether GPT-4's performance on unseen data aligns with the trends observed in the main corpus, we conducted experiments on the secret held-out dataset. The results, visualized in Figure 3, show that 0-shot prompting yields the lowest F1 scores, while N-shot prompting generally leads to improved performance. However, the differences between 1-shot, 5-shot, and 10-shot prompting remain relatively small.

In the 0-shot setting, the median F1 score is 0.2983, with an interquartile range (IQR) spanning from 0.2524 to 0.3283. While this still represents the lowest performance among the four settings, it is notably higher than the 0-shot results from the main corpus. The score distribution also shows reduced variability, with a minimum of 0.0593 and a maximum of 0.4091.

The 1-shot setup exhibits an increase in median F1 score to 0.383, with an IQR from 0.365 to 0.4103. The minimum and maximum values (0.303 and 0.4577, respectively) indicate a narrower score range than in the 0-shot setting, suggesting that introducing a single example helps stabilize performance. Compared to the main corpus, the improvements in 1-shot prompting are more pronounced, with higher overall F1 scores.

For 5-shot prompting, the median F1 score is 0.3616, with an IQR from 0.3405 to 0.3889. While this represents a slight decrease compared to 1-shot, the overall range remains compact (0.2975 to

0.4545). Similarly, in the 10-shot setting, the median F1 score is 0.3687, with an IQR of 0.3456 to 0.4162. The maximum score reaches 0.5149, the highest across all settings, yet the minimum score (0.2485) is lower than in the 1-shot and 5-shot cases. This suggests that while 10-shot prompting can occasionally yield the best individual results, it does not consistently outperform 1-shot prompting across all prompts.

Overall, GPT-4's performance on the secret held-out data surpasses that on the main corpus across all shot settings, with median F1 scores generally higher by 10% to 15%. Several factors may contribute to this discrepancy.

First, the secret held-out data includes articles from news agencies beyond the corpus data sources, which may have different language styles and structures, potentially enhancing model performance on this dataset. News articles typically have a more structured and information-dense composition, which could facilitate more accurate metaphor recognition by the model.

Second, we analyzed the sentence lengths in both datasets. The average sentence length in the secret held-out data is 25.59, while in the corpus data, it is 20.67. Longer sentences might provide more contextual information, enabling the model to utilize a richer context for metaphor recognition, thereby improving F1 scores.

These findings suggest that dataset characteristics, such as source diversity and sentence length, influence GPT-4's ability to detect metaphors. While few-shot prompting improves performance compared to 0-shot, the relatively small differences among 1-shot, 5-shot, and 10-shot settings indicate that factors beyond shot number—such as prompt design and dataset structure—may have a greater impact on performance.

5.2 Error Analysis Per Word Category

We additionally performed a detailed error analysis of different word categories, highlighting the categories where models showed significant improvements, particularly within the prompting approach. Before presenting the results, the composition of the word categories of conventional metaphors in the corpus data is first examined. As shown in Figure 4, prepositions and verbs are the most frequently occurring categories, with counts of 193 and 183, respectively. Nouns follow with 125 occurrences, while ordinals and conjunctions are the least represented.

By examining the true positive (TP), false positive (FP), and false negative (FN) rates for different word categories across various shots, we observed several trends. Most word categories did not show significant differences in performance, nor did they benefit from an increase in provided examples. Additionally, the range of word categories labeled as metaphors by the model became more diverse. Notably, only the "Preposition" category exhibited improved performance with an increase in shots, as illustrated in Figure 5. Prepositional metaphors are often challenging to detect. For example, consider the phrases "**under** strain for a popular star" and "**in** a democratic outrage". In the 0 to 5-shot scenarios, the identification of prepositional conventional metaphors remained limited, with many prompts failing to recognize any instances. The results indicate that in the 0-shot setting, metaphor detection was minimal, with an interquartile range (IQR) of 0 to 1 and a maximum of 5 occurrences. A slight improvement was observed in the 1-shot setting, where the IQR ranged from 1 to 2, suggesting some potential gains in detection capacity. Similarly, the 5-shot setting exhibited a modest increase, with an IQR extending from 1 to 3 and a maximum of 17, indicating a possible trend toward improved metaphor identification. At the 10-shot level, the median number of identified metaphors increased to 8, accompanied by a wider IQR (5 to 17.5) and a maximum of 57. This suggests a potential improvement in detection performance with higher-shot prompting.

Additionally, we used the McNemar test to assess output stability across different shot conditions and to evaluate the statistical significance of variations among different prompt designs (see Appendix .2.11. The results indicate that in the same-shot prompting condition, most prompts did not exhibit significant differences. Furthermore, in the same-prompt N-shot prompting stage (n =

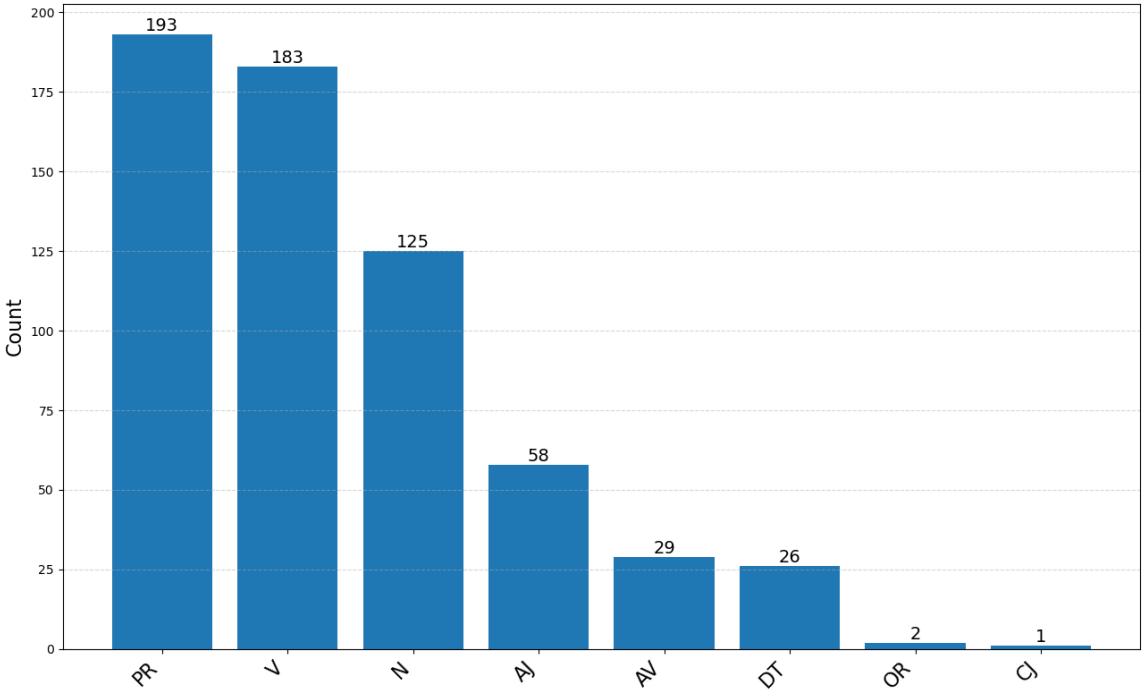

Figure 4: Word Category of Conventional Metaphors in Corpus Data. PR is preposition; V is Verb; N is Noun, AJ is Adjective; AV is Adverb; DT is Determiner; OR is Ordinal; CJ is Conjunction.

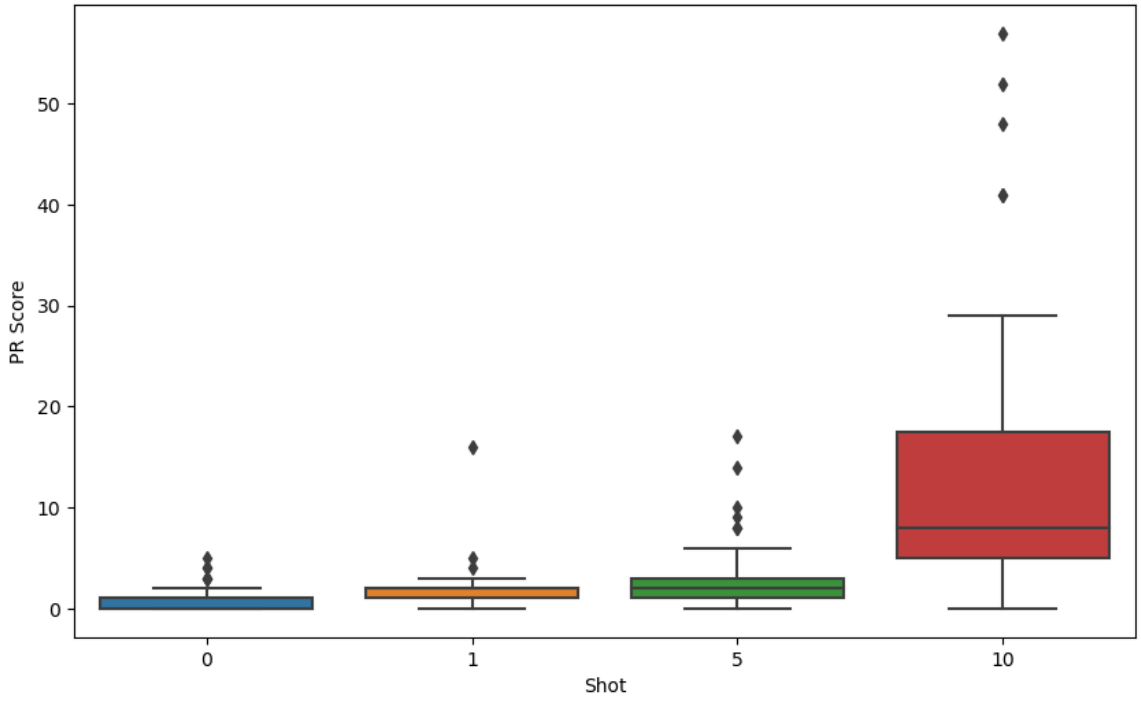

Figure 5: Boxplot of true positive of preposition from 0-shot to N-shot prompting

1, 5, 10), the instability of model outputs was notably reduced, with only a few prompts showing statistically significant differences across different runs.

6. Conclusions

In this work, we employed a prompting approach to evaluate the performance of GPT-4 in the conventional metaphor identification task. The experimental results show that the model's best F1 performance occurred in the 1-shot scenario, reaching 30.42%. Furthermore, N-shot prompting models generally outperformed 0-shot prompting, suggesting that providing examples helps enhance model performance. However, after the 1-shot scenario, increasing the number of examples did not lead to significant improvements in performance, which contrasts with the initial hypothesis that more examples would continue to improve the accuracy of the model.

To explain this phenomenon, several possible reasons are explored:

Quality and consistency of data annotation. It is noticed that while the same experiments were conducted on both corpus data and secret held-out data, the latter performed notably better. This disparity might suggest potential differences between datasets or indicate that the model is more sensitive to certain types of data.

Linguistic features of the prompt data. The style and complexity of the prompts may also have impacted the model's ability to identify metaphors consistently. Variations in these features could lead to fluctuations in performance across different prompting strategies.

Representativeness of examples. Considering the wide range of sentence lengths in the corpus data with the shortest sentence being just one word, the randomly assigned example set also contains a few short sentences. This may lack typicality and consequently affect the understanding and identification of metaphors.

Model limit. Determining the exact limits of GPT-4 for conventional metaphor detection is challenging. While GPT-4 is a general-purpose language model, it is not specifically fine-tuned for metaphor detection, which is inherently more complex due to the need to identify conventionalized metaphorical meanings. Therefore, the observed performance may reflect the model's current limitations for this particular task. However, given the vast number of possible prompt configurations and model parameters, it is reasonable to assume that more effective approaches may exist but were not explored in this study.

In conclusion, the results suggest that the limitations observed may be attributed more to the current method and prompt design rather than to the inherent abilities of GPT-4. Further exploration of optimized prompts and more targeted fine-tuning may lead to improved performance in conventional metaphor detection tasks.

7. Future Directions

The performance of GPT-4 in this project was suboptimal, and the model itself faces challenges such as lower transparency and parameter tuning limitations. We propose below a number of possible directions for utilizing LLMs and AI assistants for metaphor analysis.

7.1 Communicative Aspects

While the current project focuses on direct metaphor detection through one-time input and output without iterative conversational interaction, the communicative potential of LLMs presents a promising direction for future exploration. Dialogue-based prompting, as demonstrated in dialogue modeling approaches, involves iterative exchanges with the model and offers unique advantages. Compared to traditional metaphor detection methods, this technique may enhance metaphor detection by refining outputs through conversational feedback, generating analogies, and providing nuanced explanations.

7.2 Exploring Prompt Variation

As is mentioned above, tiny differences in prompts may result in different outputs, which can influence model performance, and the linguistic formula of prompts is unlimited, so there is still much to explore in terms of prompt variation. This includes experimenting with linguistic variations such as style and complexity, as well as different task organizations like narrative versus imperative formats. By tailoring prompts to align more closely with the intricacies of metaphorical language, the detection capabilities of the model can be potentially enhanced. Additionally, using existing training data to optimize prompt design through techniques like AutoPrompt (Levi et al. 2024) may provide new possibilities. This approach involves identifying effective manual seed prompts that have been successful and allowing the LLM to iteratively improve upon them, thereby refining its own performance. Finally, deploying Socratic prompting techniques for triggering self-reflection in LLMs is an avenue worthy of further investigation.

7.3 Exploring Output Format Variation

Since the manually annotated metaphor corpus used in this study adopts a token-level labeling scheme, we experimented with three token-level output formats: 1/0, Yes/No and IBO schema. However, alternative output formats may also influence the performance of the model. For instance, the model might perform better at the phrase level or sentence level. Additionally, the performance improvement observed after 1-shot prompting could be attributed to the model's increased familiarity with the output format. These hypotheses require further validation.

7.4 Finetuning and Retrieval-Augmented Generation (RAG)

Beyond prompt-based approaches to metaphor detection, fine-tuning and Retrieval-Augmented Generation (RAG) offer potential avenues for improving metaphor detection with LLMs. Fine-tuning open-source models such as Llama 3 enables the adaptation of LLMs to specific tasks like conventional metaphor detection by training them on targeted datasets. Retrieval-Augmented Generation approaches to metaphor detection, on the other hand, offer a potential benefit of integrating retrieval mechanisms that provide relevant context and examples from large corpora. Such retrieval-based approaches allow language models to access a broader range of metaphorical expressions and contextual information, and may improve its ability to detect and understand conventional metaphors accurately.

7.5 Logits-based Approach

While the current study relies on the model's final predictions for token-level labeling, a more direct use of logits—the raw, pre-softmax scores—may offer an alternative means of improving metaphor detection. Unlike tasks such as Named Entity Recognition (NER) or Part-of-Speech (POS) tagging, where token boundaries are typically well-defined, metaphorical expressions often exhibit fluid and ambiguous boundaries. Accessing logits may help capture the model's internal confidence and better handle such ambiguities. This approach may reduce reliance on repeated prompting or temperature tuning to control model behavior. Since OpenAI API models (including GPT-4) support direct access to logits, this method offers a practical and potentially more stable alternative to sampling-based strategies. While temperature tuning (e.g., lowering to 0 for deterministic outputs or increasing for diversity) can still be informative, its role becomes secondary when logits are used directly. Although this project does not implement a logits-based method, future work could explore its potential to refine token-level predictions, particularly for cases where model confidence is unevenly distributed across a metaphorical expression.

Acknowledgments

This research was supported by the Chinese Scholarship Council.

References

Ahrens, K., editor (2009), *Politics, Gender, and Conceptual Metaphors*, Palgrave Macmillan, Basingstoke.

Babieno, Mateusz, Masashi Takeshita, Dusan Radisavljevic, Rafal Rzepka, and Kenji Araki (2022), Miss roberta wilde: Metaphor identification using masked language model with wiktionary lexical definitions, *Applied Sciences* **12** (4), pp. 2081.

Badathala, Naveen, Abisek Rajakumar Kalarani, Tejpalsingh Siledar, and Pushpak Bhattacharyya (2023), A match made in heaven: A multi-task framework for hyperbole and metaphor detection, *in* Rogers, Anna, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, Association for Computational Linguistics, pp. 388–401. https://aclanthology.org/2023.findings-acl.26.

Baktash, J.A. and M. Dawodi (2023), Gpt-4: A review on advancements and opportunities in natural language processing, *arXiv preprint*. Accessed: 28 March 2024. https://arxiv.org/pdf/2305.03195.

Bastian, Eres Ferro, Stephan Raaijmakers, and Lettie Dorst (2024), *Exploring pain metaphor comprehension abilities in large language models*, Master's thesis, Leiden University Centre for Linguistics, Netherlands. Unpublished.

Cameron, L. (2003), *Metaphor in Educational Discourse*, Continuum, London and New York.

Caruana, Rich (1997), Multitask learning, *Machine learning* **28**, pp. 41–75, Springer.

Chen, X., C. W. Leong, M. Flor, and B. B. Klebanov (2020a), Go figure! multi-task transformer-based architecture for metaphor detection using idioms: Ets team in 2020 metaphor shared task.

Chen, X., C. W. Leong, M. Flor, and B. B. Klebanov (2023), A comprehensive survey on pretrained foundation models: A history from bert to chatgpt, *arXiv preprint arXiv:2302.09419*. Accessed: 15 June 2023.

Chen, Xianyang, Chee Wee (Ben) Leong, Michael Flor, and Beata Beigman Klebanov (2020b), Go figure! multi-task transformer-based architecture for metaphor detection using idioms: ETS team in 2020 metaphor shared task, *in* Klebanov, Beata Beigman, Ekaterina Shutova, Patricia Lichtenstein, Smaranda Muresan, Chee Wee, Anna Feldman, and Debanjan Ghosh, editors, *Proceedings of the Second Workshop on Figurative Language Processing*, Association for Computational Linguistics. https://aclanthology.org/2020.figlang-1.32.

Chen, Xin et al. (2021), Metaphor identification: A contextual inconsistency based neural sequence labeling approach, *Neurocomputing* **428**, pp. 268–279.

Choi, M., S. Lee, E. Choi, H. Park, J. Lee, D. Lee, and J. Lee (2021), Melbert: Metaphor detection via contextualized late interaction using metaphorical identification theories.

Chung, H. W., L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, and J. Wei (2022), Scaling instruction-finetuned language models, *arXiv preprint arXiv:2210.11416*. Accessed: 15 June 2023.

Dankers, Vera, Karan Malhotra, Gaurav Kudva, Vadim Medentsiy, and Ekaterina Shutova (2020), Being neighbourly: Neural metaphor identification in discourse, *Proceedings of the Second Workshop on Figurative Language Processing*, pp. 227–234.

Dankers, Verna, Marek Rei, Martha Lewis, and Ekaterina Shutova (2019), Modelling the interplay of metaphor and emotion through multitask learning, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2218–2229.

Deignan, A., J. Littlemore, and E. Semino (2013), *Figurative language, genre and register*, Cambridge University Press, Cambridge.

Deignan, Alice (2015), Metaphor in use: context, culture and communication, *Journal of Multilingual and Multicultural Development* **36** (5), pp. 548–550.

Ding, N., Y. Chen, B. Xu, Y. Qin, Z. Zheng, S. Hu, and B. Zhou (2023a), Enhancing chat language models by scaling high-quality instructional conversations, *arXiv preprint arXiv:2305.14233*.

Ding, Q., D. Ding, Y. Wang, C. Guan, and B. Ding (2023b), Unraveling the landscape of large language models: A systematic review and future perspectives, *Journal of Electronic Business & Digital Economics* **3** (1), pp. 3–19.

Dorst, A.G. (2011), *Metaphor in Fiction: Language, Thought and Communication*, BoxPress, Oisterwijk.

Dorst, A.G., W.G. Reijnierse, and G. Venhuizen (2013), One small step for mip towards automated metaphor identification? formulating general rules to determine basic meanings in large-scale approaches to metaphor, *Metaphor and the Social World* **3** (1), pp. 77–99.

Dukić, D. and J. Šnajder (2024), Looking right is sometimes right: Investigating the capabilities of decoder-only llms for sequence labeling, *arXiv preprint*.

Efrat, A., O. Honovich, and O. Levy (2022), Lmentry: A language model benchmark of elementary language tasks. Accessed: 28 March 2024.

Elzohbi, M. and R. Zhao (2023), Contrastwsd: Enhancing metaphor detection with word sense disambiguation following the metaphor identification procedure.

Fernando, C. et al. (2023), Promptbreeder: Self-referential self-improvement via prompt evolution. Accessed: 28 March 2024.

Forceville, C. (1996), *Pictorial Metaphor in Advertising*, Routledge, Abingdon.

Ge, M., R. Mao, and E. Cambria (2023), A survey on computational metaphor processing techniques: From identification, interpretation, generation to application - artificial intelligence review, *Artificial Intelligence Review*, Springer. https://link.springer.com/article/10.1007/s10462-023-10564-7.

Ge, Ming, Rui Mao, and Erik Cambria (2022), Explainable metaphor identification inspired by conceptual metaphor theory, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36, pp. 10681–10689.

Gibbs, Jr., R.W. (1994), *The poetics of mind: Figurative thought, language, and understanding*, Cambridge University Press, Cambridge, UK.

Gonen, H., S. Iyer, T. Blevins, N. A. Smith, and L. Zettlemoyer (2022), Demystifying prompts in language models via perplexity estimation.

Gong, H., K. Gupta, A. Jain, and S. Bhat (2020), Illinimet: Illinois system for metaphor detection with contextual and linguistic information.

Gu, J., H. Zhao, H. Xu, L. Nie, H. Mei, and W. Yin (2022), Robustness of learning from task instructions.

Herrmann, J. B. (2013), *Metaphor in Academic Discourse: Linguistic Forms, Conceptual Structures, Communicative Functions and Cognitive Representations*, Vol. 333 of *LOT Dissertation Series*, LOT, Utrecht.

Hicke, Rebecca MM and Ross Deans Kristensen-McLachlan (2024), Science is exploration: Computational frontiers for conceptual metaphor theory, *arXiv preprint arXiv:2410.08991*.

Jia, Kaidi and Rongsheng Li (2024), Metaphor detection with context enhancement and curriculum learning, *in* Duh, Kevin, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, pp. 2726–2737. https://aclanthology.org/2024.naacl-long.149.

Jia, Kaidi, Yanxia Wu, and Rongsheng Li (2024), Curriculum-style data augmentation for llm-based metaphor detection, *arXiv preprint arXiv:2412.02956*.

Kaal, A.A. (2012), *Metaphor in Conversation*, BoxPress, Oisterwijk.

Kim, Jeongyeon, Sangho Suh, Lydia B Chilton, and Haijun Xia (2023), Metaphorian: leveraging large language models to support extended metaphor creation for science writing, *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, pp. 115–135.

Koller, V. (2004), *Metaphor and Gender in Business Media Discourse: A Critical Cognitive Study*, Palgrave Macmillan, Basingstoke.

Krennmayr, T. (2011), *Metaphor in Newspapers*, Vol. 276 of *LOT Dissertation Series*, LOT, Utrecht.

Kövecses, Z. (2005), *Metaphor in culture: Universality and variation*, Cambridge University Press, Cambridge, UK.

Lai, Huiyuan, Antonio Toral, and Malvina Nissim (2023), Multilingual multi-figurative language detection, *in* Rogers, Anna, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, Association for Computational Linguistics, pp. 9254–9267. https://aclanthology.org/2023.findings-acl.589.

Lakoff, G. and M. Johnson (1980), *Metaphors We Live By*, University of Chicago Press, Chicago.

Lakoff, G. and M. Johnson (1999), *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*, Basic Books, New York.

Le, Duong, My Thai, and Thien Nguyen (2020), Multi-task learning for metaphor detection with graph convolutional neural networks and word sense disambiguation, *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34, pp. 8139–8146.

Leong, Chee Wee, Beata Beigman Klebanov, and Ekaterina Shutova (2018), A report on the 2018 vua metaphor detection shared task, *Proceedings of the workshop on figurative language processing*, pp. 56–66.

Leong, Chee Wee, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen (2020), A report on the 2020 vua and toefl metaphor detection shared task, *Proceedings of the second workshop on figurative language processing*, pp. 18–29.

Levin, L. S., T. Mitamura, B. MacWhinney, D. Fromm, J. G. Carbonell, W. Feely, and C. Ramirez (2014), Resources for the detection of conventionalized metaphors in four languages, *LREC*, pp. 498–501.

Li, S., J. Zeng, J. Zhang, T. Peng, L. Yang, and H. Lin (2020), Albert-bilstm for sequential metaphor detection.

Li, Y., S. Wang, C. Lin, and F. Guerin (2023), Metaphor detection via explicit basic meanings modelling.

Liu, J., N. O'Hara, A. Rubin, R. Draelos, and C. Rudin (2020), Metaphor detection using contextual word embeddings from transformers.

Low, G. (2008), Metaphor and education, *in* Gibbs, R. W. Jr., editor, *The Cambridge Handbook of Metaphor and Thought*, Cambridge University Press, Cambridge, pp. 212–231.

Low, G. (2010), *Researching and applying metaphor in the real world*, John Benjamins Publishing Company, Amsterdam, Netherlands.

Machado, M. T. and E. E. S. Ruiz (2024), Evaluating large language models for the tasks of pos tagging within the universal dependency framework, *Proceedings*.

Mao, Rui and Xiao Li (2021), Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification, *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35, pp. 13534–13542.

Mason, Z. J. (2004), Cormet: A computational, corpus-based conventional metaphor extraction system, *Computational Linguistics* **30** (1), pp. 23–44.

Maudslay, R. H. and S. Teufel (2022), Metaphorical polysemy detection: Conventional metaphor meets word sense disambiguation. Accessed: 28 March 2024.

Mizrahi, M., G. Kaplan, D. Malkin, R. Dror, D. Shahaf, and G. Stanovsky (2023), State of what art? a call for multi-prompt llm evaluation. Accessed: 28 March 2024.

Musolff, A. (2004), *Metaphor and Political Discourse: Analogical Reasoning in Debates about Europe*, Palgrave Macmillan, Basingstoke.

OpenAI (2023), New models and developer products announced at devday. Accessed: 28 March 2024.

Pragglejaz Group (2007), Mip: A method for identifying metaphorically used words in discourse, *Metaphor and Symbol* **22** (1), pp. 1–39.

Qi, Shuai, Zhihao Cao, Jian Rao, Lijun Wang, Jing Xiao, and Xiang Wang (2023), What is the limitation of multimodal llms? a deeper look into multimodal llms through prompt probing, *Information Processing*
Management **60** (6), pp. 103510.

Qiu, R. (2023), Gpt revolutionizing ai applications: Empowering future digital transformation, *Digital Transformation and Society* **2** (2), pp. 101–103.

Reijnierse, W. G., C. Burgers, T. Krennmayr, and G. J. Steen (2018), Dmip: A method for identifying potentially deliberate metaphor in language use, *Corpus Pragmatics* **2** (2), pp. 129–147.

Reijnierse, W. Gudrun et al. (2017), *The Value of Deliberate Metaphor*, LOT, Utrecht.

Semino, E., Z. Demjen, A. Hardie, S. A. Payne, and P. E. Rayson (2018), *Metaphor, Cancer and the End of Life: A Corpus-based Study*, Routledge, London.

Shutova, E. (2015, p. 580-1), Design and evaluation of metaphor processing systems, *Computational Linguistics* **41** (4), pp. 579–623.

Song, W. et al. (2021), Verb metaphor detection via contextual relation learning. Accessed: 28 March 2024.

Song, Ziqi, Shengwei Tian, Long Yu, Xiaoyu Zhang, and Jing Liu (2024), Multi-task metaphor detection based on linguistic theory, *Multimedia Tools and Applications* pp. 1–14, Springer.

Srivastava, A., A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, and G. Wang (2022), Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Steen, G. (2015), Developing, testing and interpreting deliberate metaphor theory, *Journal of Pragmatics* **90** (1), pp. 67–72.

Steen, G.J., A.G. Dorst, J.B. Herrmann, A.A. Kaal, and T. Krennmayr (2010a), Metaphor in usage, *Cognitive Linguistics* **21** (4), pp. 765–796.

Steen, G.J., A.G. Dorst, J.B. Herrmann, A.A. Kaal, T. Krennmayr, and T. Pasma (2010b), *A Method for Linguistic Metaphor Identification: From MIP to MIPVU*, John Benjamins, Amsterdam.

Steen, G.J., A.G. Dorst, J.B. Herrmann, A.A. Kaal, T. Krennmayr, and T. Pasma (2010c), Vu amsterdam metaphor corpus. http://hdl.handle.net/20.500.12024/2541.

Su, C., K. Wu, and Y. Chen (2021), Enhanced metaphor detection via incorporation of external knowledge based on linguistic theories.

Subedi, B., S. Regmi, B. K. Bal, and P. Acharya (2024), Exploring the potential of large language models (llms) for low-resource languages: A study on named-entity recognition (ner) and part-of-speech (pos) tagging for nepali language, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 6974–6979.

Tan, Z., D. Li, S. Wang, A. Beigi, B. Jiang, A. Bhattacharjee, and H. Liu (2024), Large language models for data annotation and synthesis: A survey, *arXiv preprint*.

Tian, Yuan, Ruike Zhang, Nan Xu, and Wenji Mao (2024), Bridging word-pair and token-level metaphor detection with explainable domain mining, *in* Ku, Lun-Wei, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, pp. 13311–13325. https://aclanthology.org/2024.acl-long.719.

Tong, X., R. Choenni, M. Lewis, and E. Shutova (2024), Metaphor understanding challenge dataset for llms. arXiv.org.

Tong, Xiaoyu, Ekaterina Shutova, and Martha Lewis (2021, p. 4676), Recent advances in neural metaphor processing: A linguistic, cognitive and social perspective, *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 4673–4686.

Uduehi, O. O. and R. C. Bunescu (2023), An expectation-realization model for metaphor detection. Accessed: 28 March 2024.

Villena, F., L. Miranda, and C. Aracena (2024), llmner: (zero|few)-shot named entity recognition, exploiting the power of large language models, *arXiv preprint*.

Wachowiak, L. and D. Gromann (2023), Does gpt-3 grasp metaphors? identifying metaphor mappings with generative language models. Accessed: 28 March 2024.

Wan, M. et al. (2020), Using conceptual norms for metaphor detection. Accessed: 28 March 2024.

Wang, Jie, Jin Wang, and Xuejie Zhang (2024), Chinese metaphor recognition using a multi-stage prompting large language model, *CCF International Conference on Natural Language Processing and Chinese Computing*, Springer, pp. 234–246.

Wang, S., Y. Li, C. Lin, L. Barrault, and F. Guerin (2023), Metaphor detection with effective context denoising.

Wilks, Y., A. Dalton, J. Allen, and L. Galescu (2013), Automatic metaphor detection using large-scale lexical resources and conventional metaphor extraction, *Proceedings of the First Workshop on Metaphor in NLP*, pp. 36–44.

Wilks, Yorick (1975), A preferential, pattern-seeking, semantics for natural language inference, *Artificial Intelligence* **6** (1), pp. 53–74.

Wilks, Yorick (1978), Making preferences more active, *Artificial Intelligence* **11** (3), pp. 197–223.

Xie, T., Q. Li, J. Zhang, Y. Zhang, Z. Liu, and H. Wang (2023), Empirical study of zero-shot ner with chatgpt, *arXiv preprint*.

Xu, Yanzhi, Yueying Hua, Shichen Li, and Zhongqing Wang (2024), Exploring chain-of-thought for multi-modal metaphor detection, *in* Ku, Lun-Wei, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, pp. 91–101. https://aclanthology.org/2024.acl-long.6.

Yang, Cheng, Puli Chen, and Qingbao Huang (2024), Can chatgpt's performance be improved on verb metaphor detection tasks? bootstrapping and combining tacit knowledge, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1016–1027.

Yang, Qimeng, Long Yu, Shengwei Tian, and Jinmiao Song (2021), Collaborative semantic representation network for metaphor detection, *Appl. Soft Comput.*, Elsevier Science Publishers B. V., NLD. https://doi.org/10.1016/j.asoc.2021.107911.

Zhang, Shenglong and Ying Liu (2022), Metaphor detection via linguistics enhanced siamese network.

Zhang, Shenglong and Ying Liu (2023), Adversarial multi-task learning for end-to-end metaphor detection, *in* Rogers, Anna, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, Association for Computational Linguistics, pp. 1483–1497. https://aclanthology.org/2023.findings-acl.96.

Zhou, Chao, Qiaoyan Li, Cheng Li, Jiawei Yu, Yao Liu, Guangyi Wang, Ke Zhang, Chunyu Ji, Qi Yan, Lei He, Haoyang Peng, Jia Li, Jiali Wu, Zhiyuan Liu, Peng Xie, Chao Xiong, Jian Pei, Philip S. Yu, and Lijun Sun (2023), A comprehensive survey on pretrained foundation models: A history from bert to chatgpt, *arXiv*. Accessed: 19 August 2024. https://arxiv.org/abs/2302.09419.

A Appendix: guideline for model output sorting

The sorting of outputs is divided into two main parts: The first part involves organizing the results of the model output, which are in txt format, containing word list with conventional metaphors labelled with 1, and the second part involves generating a csv file from the organized results of the first part, using the words and labels of model outputs to compare with ground truth labels from corpus for score calculation.

The objective here is to ensure the text is in the correct format (list words and labels) for csv file generation. The encountered issues and their solutions include:

.1 Sorting of Raw Text Outputs

The objective here is to ensure the text is in the correct format (list words and labels) for csv file generation. The encountered issues and their solutions include:

.1.1 Word Formatting Issues

- Problem: Words are not on separate lines but are placed on the same line, separated by spaces or commas.
- Solution: Format them into one word per line.

.1.2 Self-Created Word Groups and Labels

- Problem: The model creates its own word groups and assigns labels to the entire group. For example,

```
Standard: Letter
          Policy
Output:   Letter Policy
```

- Solution: Require the model to output again.

.1.3 Marks Before Words

- Problem: There are marks (e.g. "') before words.
- Solution: Remove the marks.

.1.4 Retaining Original Sentences and Word Lists

- Problem: Both the original sentences and word lists are retained. For example,

```
Standard: Time
          flies:1
Output:   Time flies
          Time
          flies:1
```

- Solution: Delete the original sentences.

.1.5 Separate Word Lists and Labels

- Problem: Word lists and labels are separate. For example,

```
Output: Time
        flies

        Here are the conventional metaphorical words:
        flies:1
```

- Solution: Merge the labels with the word lists.

.1.6 Including Additional Content

- Problem: Includes explanations of words, opening or closing sentences, or symbols.
- Solution: Delete non-list content, retaining only the word list.

.1.7 Partial Word Lists

- Problem: Only a part of the word list is enumerated.
- Solution: Require the model to output the complete list.

.1.8 All Words Marked as 1

- Problem: Every word in a sentence is marked as 1.
- Solution: Retain this judgment as the model's output may vary.

.1.9 Words and labels are not on the same row

- Problem: Words and labels are on different rows.
- Solution: Put the labels back to the same rows with the words.

.2 Sorting of Raw CSV (Raw csv) Outputs

After sorting the word lists in the txt files to right format (word:label, one word per row), the sorted outputs are then transformed into two columns in a csv file. These are compared with the manual word and label columns. The process involves handling misalignments between the machine output word list and the manual word list, mainly in two categories:

.2.1 Mismatch in Word Tokenization

- Problem: Words with different tokenization from the manual version.
  - Words are split differently than expected. For example,

```
Manual: Runner+beans
Output: Runner
        Beans:1
```

  - Words are combined differently than expected. For example,

```
Manual: Letter:1
        Policy
Output: Letter Policy:1
```

  - Words are partially split differently than expected. For example,

```
Manual: set+up:1
Output: up:1
```

- Solution: Since there is no clear pattern for when and how the word tokenization problems occur, manually check and adjust are used in this step:
  - For problem one and three, if a part of the word pair is labelled as 1, then the whole word pair will be labelled as 1.
  - For problem two, if the word pair is labelled as 1, then every separate words will be labelled as 1.

.2.2 Incomplete word:label Lists

- Problem: The output word list is incomplete.
- Solution: Align using Python, with missing labels defaulted to 0.

.2.3 Word Formatting Issues

- Problem: Words are not on separate lines but are placed on the same line, separated by spaces or commas.
- Solution: Format them into one word per line.

.2.4 Self-Created Word Groups and Labels

- Problem: The model creates its own word groups and assigns labels to the entire group. For example,

```
Standard: We
        must
        kill:1
        the
        program
Output: We must kill the program:1
```

- Solution: Require the model to output again.

.2.5 Marks Before Words

- Problem: There are marks (e.g. "') before words.
- Solution: Remove the marks.

.2.6 Retaining Original Sentences and Word Lists

- Problem: Both the original sentences and word lists are retained. For example,

```
Standard: Time
        flies:1
Output: Time flies
        Time
        flies:1
```

- Solution: Delete the original sentences.

.2.7 Separate Word Lists and Labels

- Problem: Word lists and labels are separate. For example,

```
Standard: Time
        flies:1
Output: Time
        flies
        Here are the conventional metaphorical words:
        flies:1
```

- Solution: Merge the labels with the word lists.

.2.8 Including Additional Content

- Problem: Includes explanations of words, opening or closing sentences, or symbols. For example,

```
Output:
Step 1: Analyze every word in each given sentence to identify
conventional metaphors. Conventional metaphors identified in the
sentences:

1. (No conventional metaphor identified)}

2. "set+up" (could be interpreted as a metaphor for establishing),
"ended" (metaphor for cessation),
"futile" (metaphor for lack of success),
"infighting" (metaphor for conflict)}
......

Step 2: List every word in the given sentence sequentially
```

- Solution: Delete non-list content, retaining only the word list.

.2.9 Partial Word Lists

- Problem: Only a part of the word list is enumerated. For example,

```
Standard: word list 1
        word list 2
        ......
        word list 10
Output: Since there are 21 sentences and the task is quite lengthy,
I will demonstrate the process using the first three sentences:
        word list 1
```

- Solution: Require the model to output the complete list.

.2.10 All Words Marked as 1

- Problem: Every word in a sentence is marked as 1. For example,

```
Standard: We
        must
        kill:1
        the
        program
Output: We:1
        must:1
        kill:1
        the:1
        program:1
```

- Solution: Require the model to output again

.2.11 Words and labels are not on the same row

- Problem: Words and labels are on different rows. For example,

```
Standard: Time
        flies:1
Output: Time
        flies
        :1
```

- Solution: Put the labels back to the same rows with the words.

B Appendix: performance report of one-shot prompting of corpus data

Prompt	#F1	#Recall	#Precision	#Feature
1	0.1987	0.1242	0.4976	simple instruction
2	0.2071	0.1248	0.6090	simple instruction
3	0.2228	0.1409	0.5331	simple instruction
4	**0.3042**	**0.2108**	0.5522	simple instruction
5	0.2446	0.1553	0.5762	simple instruction
6	0.1971	0.1192	0.5673	simple instruction
7	0.2189	0.1647	0.4046	simple instruction
8	0.2709	0.1803	0.5453	background information
9	0.1990	0.1198	0.5929	simple instruction
10	0.1794	0.1043	**0.6421**	simple instruction
11	0.2385	0.1509	0.5703	simple instruction
12	0.2161	0.1364	0.5211	simple instruction
13	0.2774	0.1886	0.5251	simple instruction
14	0.2241	0.1425	0.5267	simple instruction
15	0.2160	0.1359	0.5271	simple instruction
16	0.2029	0.1242	0.5564	background information
17	0.2832	0.1930	0.5318	background information
18	0.2896	0.2019	0.5131	background information
19	0.2627	0.1797	0.4880	background information
20	0.2857	0.1925	0.5554	background information
21	0.2977	0.2063	0.5359	adapted from MIP
22	0.2045	0.1281	0.5116	adapted from MIP
23	0.2672	0.1808	0.5133	adapted from MIP
24	0.2468	0.1642	0.5011	adapted from MIP
25	0.2785	0.1886	0.5342	adapted from MIP

Table 6: Performance Metrics by Prompt Type in One-Shot Prompting

C Appendix: performance report of five-shot prompting of corpus data

Prompt	**#F1**	**#Recall**	**#Precision**	**#Feature**
1	0.1871	0.1159	0.4852	simple instruction
2	0.1865	0.1159	0.4803	simple instruction
3	0.1840	0.1159	0.4617	simple instruction
4	0.2120	0.1375	0.4729	simple instruction
5	0.1592	0.0998	0.4458	simple instruction
6	0.1808	0.1137	0.4556	simple instruction
7	0.1987	0.1498	0.3510	simple instruction
8	0.2177	0.1431	0.4707	background information
9	0.1649	0.1004	0.4708	simple instruction
10	0.1807	0.1104	**0.4991**	simple instruction
11	0.1854	0.1170	0.4519	simple instruction
12	0.1661	0.1015	0.4574	simple instruction
13	0.1735	0.1137	0.4342	simple instruction
14	0.1971	0.1242	0.4771	simple instruction
15	0.1963	0.1253	0.4590	simple instruction
16	0.1814	0.1120	0.4776	background information
17	0.2151	0.1387	0.4945	background information
18	0.2092	0.1359	0.4621	background information
19	0.2362	0.1803	0.4142	background information
20	0.2375	0.1575	0.4877	background information
21	**0.2635**	**0.1830**	0.4760	adapted from MIP
22	0.1626	0.1004	0.4289	adapted from MIP
23	0.2427	0.1658	0.4604	adapted from MIP
24	0.2273	0.1503	0.4664	adapted from MIP
25	0.2439	0.1675	0.4548	adapted from MIP

Table 7: Performance Metrics by Prompt Type in 5-Shot Prompting

D Appendix: performance report of ten-shot prompting of corpus data

Prompt	**#F1**	**#Recall**	**#Precision**	**#Feature**
1	0.1720	0.1048	0.4844	simple instruction
2	0.1980	0.1209	0.5513	simple instruction
3	0.2398	0.1536	0.5466	simple instruction
4	0.2196	0.1420	0.4877	simple instruction
5	0.1721	0.1026	0.5348	simple instruction
6	0.1488	0.0876	0.4935	simple instruction
7	0.1876	0.1181	0.4569	simple instruction
8	0.2336	0.1492	0.5405	background information
9	0.1988	0.1237	0.5090	simple instruction
10	**0.2643**	**0.1708**	0.5841	simple instruction
11	0.1939	0.1204	0.5017	simple instruction
12	0.1759	0.1076	0.4815	simple instruction
13	0.2225	0.1370	**0.5990**	simple instruction
14	0.2038	0.1276	0.5076	simple instruction
15	0.1700	0.1043	0.4615	simple instruction
16	0.1920	0.1204	0.4757	background information
17	0.2198	0.1359	0.5749	background information
18	0.1895	0.1187	0.4703	background information
19	0.2368	0.1553	0.4989	background information
20	0.2380	0.1553	0.5100	background information
21	0.2558	0.1681	0.5353	adapted from MIP
22	0.1160	0.0666	0.4540	adapted from MIP
23	0.2137	0.1392	0.4598	adapted from MIP
24	0.2224	0.1436	0.4933	adapted from MIP
25	0.2191	0.1420	0.4825	adapted from MIP

Table 8: Performance Metrics by Prompt Type in 10-Shot Prompting

E Appendix: performance report of zero-shot prompting of secret held-out data

Prompt	#F1	#Recall	#Precision	#Feature
1	0.1022	0.0560	0.5839	simple instruction
2	0.3033	0.1973	0.6623	simple instruction
3	0.3034	0.2320	0.4391	simple instruction
4	0.3014	0.1893	0.7387	simple instruction
5	0.0642	0.0347	0.4330	simple instruction
6	0.2364	0.1467	0.6117	simple instruction
7	0.3023	0.1920	0.7229	simple instruction
8	0.3483	0.2373	0.6552	background information
9	0.2883	0.2027	0.5010	simple instruction
10	0.2914	0.1813	0.7554	simple instruction
11	0.1926	0.1120	0.7228	simple instruction
12	0.2780	0.1760	0.6633	simple instruction
13	0.1873	0.1093	0.6639	simple instruction
14	0.3099	0.2000	0.6919	simple instruction
15	0.3444	0.2453	0.5794	simple instruction
16	0.2822	0.1813	0.6375	background information
17	0.3648	0.2720	0.5565	background information
18	0.2572	0.1573	0.7136	background information
19	**0.3916**	**0.2747**	0.6820	background information
20	0.3261	0.2160	0.6659	background information
21	0.2843	0.1733	**0.8247**	adapted from MIP
22	0.2812	0.1813	0.6757	adapted from MIP
23	0.2588	0.1653	0.5980	adapted from MIP
24	0.3170	0.2080	0.6688	adapted from MIP
25	0.3546	0.2667	0.5317	adapted from MIP

Table 9: Performance Metrics by Prompt Type in One-Shot Prompting

F Appendix: performance report of one-shot prompting of secret held-out data

Prompt	#F1	#Recall	#Precision	#Feature
1	0.3365	0.2560	0.4944	simple instruction
2	0.3895	0.2773	0.6540	simple instruction
3	0.3859	0.3013	0.5366	simple instruction
4	0.3998	0.3200	0.5330	simple instruction
5	0.3717	0.2800	0.5535	simple instruction
6	0.3766	0.2640	**0.6580**	simple instruction
7	0.3664	0.2747	0.5508	simple instruction
8	0.4114	0.3307	0.5488	background information
9	0.3907	0.2880	0.6077	simple instruction
10	0.3385	0.2293	0.6517	simple instruction
11	0.4309	0.3333	0.6104	simple instruction
12	0.3358	0.2373	0.5741	simple instruction
13	0.3788	0.3120	0.4848	simple instruction
14	0.3763	0.2693	0.6251	simple instruction
15	0.3902	0.2907	0.5980	simple instruction
16	0.3771	0.2693	0.6332	background information
17	0.3768	0.2960	0.5187	background information
18	0.3892	0.3200	0.4973	background information
19	0.4176	0.3173	0.6176	background information
20	0.3647	0.2747	0.5428	background information
21	0.3620	0.2853	0.4956	adapted from MIP
22	0.3772	0.2960	0.5204	adapted from MIP
23	0.4252	0.3413	0.5638	adapted from MIP
24	0.4159	0.3120	0.6265	adapted from MIP
25	**0.4514**	**0.3653**	0.5908	adapted from MIP

Table 10: Performance Metrics by Prompt Type in 1-Shot Prompting

G Appendix: performance report of five-shot prompting of secret held-out data

Prompt	#F1	#Recall	#Precision	#Feature
1	0.3135	0.2187	0.5541	simple instruction
2	0.3675	0.2613	0.6199	simple instruction
3	0.3508	0.2507	0.5854	simple instruction
4	0.3754	0.2693	0.6248	simple instruction
5	0.3785	0.2907	0.5426	simple instruction
6	0.3349	0.2507	0.5049	simple instruction
7	0.3528	0.2507	0.5967	simple instruction
8	0.3404	0.2720	0.4550	background information
9	0.3478	0.2507	0.5687	simple instruction
10	0.3704	0.2907	0.5114	simple instruction
11	0.4123	0.3013	**0.6529**	simple instruction
12	0.3380	0.2400	0.5729	simple instruction
13	0.3962	**0.3440**	0.4683	simple instruction
14	0.3475	0.2427	0.6215	simple instruction
15	0.3546	0.2693	0.5202	simple instruction
16	0.3591	0.2587	0.5884	background information
17	0.3761	0.2853	0.5516	background information
18	0.3796	0.3093	0.4913	background information
19	0.3615	0.2587	0.6033	background information
20	0.3874	0.2987	0.5521	background information
21	0.3287	0.3413	0.3400	adapted from MIP
22	0.3482	0.2480	0.5851	adapted from MIP
23	0.4155	0.3440	0.5251	adapted from MIP
24	0.3698	0.2773	0.5554	adapted from MIP
25	**0.4373**	0.3440	0.6006	adapted from MIP

Table 11: Performance Metrics by Prompt Type in 5-Shot Prompting

H Appendix: performance report of ten-shot prompting of secret held-out data

Prompt	#F1	#Recall	#Precision	#Feature
1	0.2661	0.1813	0.5000	simple instruction
2	0.3355	0.2347	0.5902	simple instruction
3	0.3319	0.2267	0.6206	simple instruction
4	0.3859	0.2827	0.6141	simple instruction
5	0.3914	0.2960	0.5780	simple instruction
6	0.3574	0.2533	0.6122	simple instruction
7	0.3628	0.2480	0.6783	simple instruction
8	0.4351	0.3493	0.5776	background information
9	0.3081	0.2080	0.6039	simple instruction
10	0.3256	0.2213	0.6263	simple instruction
11	0.4442	0.3280	**0.6897**	simple instruction
12	0.3778	0.2880	0.5572	simple instruction
13	0.4085	0.3307	0.5366	simple instruction
14	0.3373	0.2560	0.4981	simple instruction
15	0.4211	0.3147	0.6424	simple instruction
16	0.3361	0.2320	0.6246	background information
17	0.3903	0.2800	0.6442	background information
18	0.3836	0.2960	0.5524	background information
19	0.3509	0.2560	0.5604	background information
20	0.4105	0.3120	0.6000	background information
21	0.3777	0.2853	0.5598	adapted from MIP
22	0.3838	0.2827	0.5988	adapted from MIP
23	0.4454	**0.3680**	0.5722	adapted from MIP
24	0.3814	0.2853	0.5778	adapted from MIP
25	**0.4652**	0.3600	0.6608	adapted from MIP

Table 12: Performance Metrics by Prompt Type in 10-Shot Prompting

I Appendix: McNemar test for prompt instability

The McNemar test is a statistical test used to determine if there are significant differences for a dichotomous dependent variable between two related groups. In this project,

We use the McNemar test to determine whether the differences between the various 0-shot and N-shot results, as well as the different prompt designs, are statistically significant.

For the former, the comparisons were made among all possible pairs between 1 to 10 shot within three runs of experiments. This was done to account for the model's output instability, as the same input prompts could yield different outputs across runs. By repeating each experiment three times and averaging the results, the scores became more reliable and reflective of the model's overall performance.

The results indicated that most prompts did not show significant differences in performance across these shot conditions. However, when examining prompts that did exhibit significant differences, instability is observed. Specifically, prompts with significant differences in the same comparison varied across the three experimental iterations. For example, in the comparison between 0-shot and 1-shot, prompts 4, 14, 22, and 25 showed significant differences in the first experiment, whereas in the second experiment, the prompts with significant differences were prompts 7, 15, and 24.

For the latter, results show that in 0-shot prompting phase, the frequency of statistically significant differences among different types of prompts is relatively high. However, this phenomenon weakens in N-shot prompting phase.