Easy as PIE? Identifying Multi-Word Expressions with LLMs

Kai Golan Hashiloni and **Ofri Hefetz** and **Kfir Bar**
Efi Arazi School of Computer Science, Reichman University, Herzilya, Israel
{kai.golanhashiloni,ofri.hefetz}@post.runi.ac.il, kfir.bar@runi.ac.il

Abstract

We investigate the identification of idiomatic expressions—a semantically non-compositional subclass of multiword expressions (MWEs)—in running text using large language models (LLMs) without any fine-tuning. Instead, we adopt a prompt-based approach and evaluate a range of prompting strategies, including zero-shot, few-shot, and chain-of-thought variants, across multiple languages, datasets, and model types. Our experiments show that, with well-crafted prompts, LLMs can perform competitively with supervised models trained on annotated data. These findings highlight the potential of prompt-based LLMs as a flexible and effective alternative for idiomatic expression identification.

1 Introduction

Multiword expressions (MWEs), combinations of words that exhibit idiosyncratic syntactic or semantic behavior, are pervasive in natural language and pose a long-standing challenge for computational models. Examples such as "red tape" (referring to excessive bureaucracy) illustrate the central linguistic phenomenon of non-compositionality, where the meaning of the whole cannot be deduced from its parts. MWEs encompass a wide range of constructions, including idioms, collocations, light verb constructions, and compound nouns, and they vary significantly across languages in both form and frequency.

From a natural-language-processing (NLP) perspective, reliably identifying MWEs is essential for tasks like machine translation (Baziotis et al., 2023) and semantic representation (Cohen et al., 2022). Misinterpreting idioms as literal phrases can harm performance, especially in multilingual or low-resource settings. MWEs also often reflect cultural or domain-specific meaning, making them important for higher-level language analysis.

One particularly compelling application of MWE detection lies in the domain of intertextuality, where researchers seek to trace intellectual and textual connections across corpora, such as identifying characteristic expressions that recur across ancient manuscripts. In such settings, we believe that MWEs can act as linguistic fingerprints, enabling the identification of subtle relationships between texts that might otherwise remain hidden.

As linguistic demands grow, large language models (LLMs) have advanced state-of-the-art performance across many NLP tasks. However, their ability to detect MWEs—especially in `zero-shot` settings without fine-tuning—remains underexplored. This raises questions about how well generative LLMs capture non-compositional meaning and phrase-level semantics. A recent survey by Miletić and Walde (2024) highlights encoder-based, fine-tuned approaches to MWE detection. In contrast, we examine whether generative LLMs can identify MWEs directly in running text without additional training, focusing on idiomatic expressions—a subclass of MWEs whose meaning is not inferable from their individual words (e.g., "spill the beans").

While idioms are often figurative, not all figurative expressions are idiomatic; idioms are also fixed and conventionalized. They fall under the broader category of *formulaic language*, which includes frequent word combinations like collocations (e.g., "make a decision") and fixed phrases (e.g., "by the way"). Idioms are unique in being both formulaic and semantically non-compositional. A key challenge is that idioms can be identical in form to literal expressions, depending on context (e.g., "I spilled the beans in the kitchen"). Thus, "spill the beans" is a potential idiomatic expression (PIE) that may appear with either meaning. We define the task as follows: given a document, the model must identify all PIEs used idiomatically in context.

We design several prompting strategies, including techniques that encourage the model to "think

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 23771–23790
November 4-9, 2025 ©2025 Association for Computational Linguistics

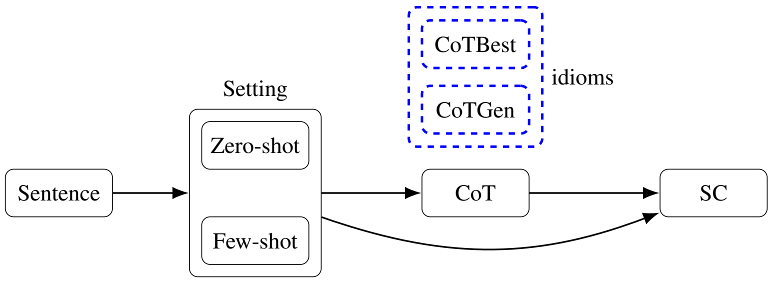

Figure 1: The promoting strategies and settings we use in this study.

creatively" before extracting MWEs. These are contrasted with a recently proposed reasoning-augmented LLMs, such as DeepSeek-R1 or OpenAI's GPT-o3, which are designed to simulate general cognitive steps before responding. We find that generative LLMs, when prompted appropriately, can match or even exceed the performance of several supervised baselines trained explicitly for idiomatic expression identification.

Our main contributions are as follows: **(1)** We demonstrate that generative LLMs are capable of detecting non-compositional instances of idiomatic expressions in running text at a quality competitive with fine-tuned identification models, even in `zero-shot` settings. **(2)** We conduct a comprehensive multilingual evaluation of MWE detection, covering a typologically diverse set of languages and highlighting key cross-linguistic differences and challenges. All code, prompts, and reproducibility details are available in our project repository.[1]

2 Related Work

2.1 Multiword Expression Identification

MWE identification has long been studied in NLP, traditionally framed as a sequence tagging task (Constant et al., 2017). Early methods relied on linguistic rules; for example, Pasquer et al. (2020) proposed Seen2Seen, which learns verbal MWE candidates and filters them using morphosyntactic constraints.

With pretrained language models, contextual embeddings became central. Taslimipoor et al. (2020) introduced MTLB-STRUCT, a semi-supervised method fine-tuning BERT to jointly predict VMWEs and syntactic dependencies, evaluated on the PARSEME 1.3 shared task (Savary et al., 2023). MWEasWSD (MaW) (Tanner and Hoffman, 2023) further combined rule-based filtering with a bi-encoder model, achieving strong results on DiMSUM (Schneider et al., 2016).

More recently, Ide et al. (2024) introduced the CoAM dataset and showed that fine-tuning large models like Qwen-2.5-72B (Team, 2024) outperforms prior methods, signaling a shift toward LLM-based solutions for MWE identification.

2.2 Idiomatic Expression Identification

Idioms are a subclass of MWEs characterized by their figurative meaning and lack of compositionality (Timothy Baldwin, 2010). Several datasets have been introduced for the task of idiom identification, including VNC-Tokens (Cook et al., 2008), Gigaword (Sporleder and Li, 2009), SemEval-2013 Task 5 (Korkontzelos et al., 2013), and IDIX (Sporleder et al., 2010). However, these resources have faced criticism for being limited in size or in the variety of idiom types they cover (Haagsma et al., 2019; Mi et al., 2024). Tedeschi et al. (2022) further note that most of these datasets focus primarily on English, and describe the more recent SemEval-2022 dataset (Madabushi et al., 2021) as small and restricted in terms of language coverage. The OpenMWE Corpus (Hashimoto and Kawahara, 2009) is a large Japanese dataset for idiom classification comprising 102,334 instances across 146 idioms.

In addition, all these datasets are largely designed for the idiom *classification* task, where the system is provided with both a context and a PIE. In contrast, we address the identification task: identifying idiomatic expressions directly from the sentence, without being given the candidate PIE in advance. An example of this tasks can be found in Dodiom (Eryiğit et al., 2022), a multilingual resource for idiom identification created via gamified crowdsourcing in Turkish and Italian. With the advent of word embeddings, both static and contextual, new approaches emerged that modeled idiomaticity as a divergence between the vector rep-

[1] `https://github.com/Intellexus-DSI/easy-as-pie/`

resentations of an expression and those of its parts (Gharbieh et al., 2016; Ehren, 2017; Senaldi et al., 2019; Hashempour and Villavicencio, 2020; Garcia et al., 2021; Nedumpozhimana and Kelleher, 2021).

Recent work has applied transformer-based models to idiom identification, including multi-stage architectures with attention (Zeng and Bhat, 2021), BiLSTM-CRF models over subword embeddings (Tedeschi et al., 2022), and lexicon-augmented transformers (Hadj Mohamed et al., 2024). LLMs have also been explored for idiom classification (De Luca Fornaciari et al., 2024; Mi et al., 2024; Phelps et al., 2024), but, to our knowledge, none have tackled MWE identification without fine-tuning. In this work, we address this gap by using generative LLMs to identify idiomatic expressions in context, relying solely on prompt-based guidance.

3 Method

We prompt a generative LLM with task instructions and a sentence, asking it to identify idiomatic expressions. We evaluate various prompting strategies across multilingual datasets.

3.1 Problem Formulation

While our primary focus is on idiom identification, we also include experiments on one dataset that covers a broader range of MWE types; in this case, we refer to the task as MWE identification (MWEI). Given that LLMs are naturally suited for processing raw, unstructured text, we prompt the model with plain sentences and ask it to return a list of idiomatic expressions in a structured JSON format. In the simplest setting (e.g., with a `zero-shot` prompt), the expected JSON structure includes an "mwes" field, which is a list of the *identified* expressions, each represented by its surface form—that is, as it appears in the input text. For more advanced prompts incorporating chain-of-thought reasoning, the output includes additional fields like "sentence" and "explanation" to justify why specific expressions are considered idiomatic. The prompt templates and output schemas are described in Section 3.3, with illustrative examples provided in Appendix K.

3.2 Datasets

In our experiments, we use two datasets with distinct characteristics, enabling more general insights into LLMs' ability to perform idiom identification in `zero-shot` settings. Additionally, we include a dataset for MWE identification (MWEI) to explore the model's capabilities beyond idioms. We use the datasets according to the original intention of the providers. Table 1 provides an overview of the dataset statistics. License information for all datasets is provided in Appendix J.

ID10M (Tedeschi et al., 2022). A multilingual dataset created for the task of idiom identification, featuring automatically generated training data in ten different languages. The authors released a manually annotated evaluation benchmark consisting of 200 samples in English, Italian, and German—and reportedly in Spanish, although only 199 samples are present in the published data. Idioms were sourced from Wiktionary,[2] with any idioms not found there excluded from annotation. The dataset includes only *continuous* idiom spans, meaning all words within each span are part of the idiom, with no interruptions or excluded tokens (also known as *gaps*). It supports multiple idioms per instance and uses the standard BIO (Begin-Inside-Outside) tagging scheme for annotating idiomatic expressions.

MAGPIE (Haagsma et al., 2020). This dataset contains 56,622 English instances drawn from diverse genres, including news and science. The dataset was originally designed for idiom *classification*, where the task is to classify a PIE in a sentence as idiomatic or literal. Since our goal is *identification*, we adapt the dataset for this task, as detailed in Appendix A, where we also describe some filtering we apply. This process yields 4,391 test samples, from which we randomly select 400 for evaluation. The filtered version, marked as *ours* in Table 1, will be released with this paper to support reproducibility and future improvements.

CoAM (Ide et al., 2024). As noted above, we also aim to extend the evaluation beyond idiomatic expressions to include additional types of MWEs. Detecting MWEs is more challenging, as they exhibit diverse linguistic properties and may include more gaps—for example, in "turn China into", the MWE is typically defined as "turn into".

For this set of experiments, we choose to use CoAM, a dataset designed for comprehensive MWE identification, extending beyond the commonly studied verbal MWEs (VMWEs) to encom-

[2]https://www.wiktionary.org/

Dataset	Task	Language	Train		Test	
			# Sentences	# MWEs/Idioms	# Sentences	# MWEs/Idioms
Id10M	Idioms	EN	37,919	4,568	200	142
		DE	26,963	819	200	111
		IT	29,523	452	200	139
		ES	28,647	1,229	199	78
MAGPIE	Idioms	EN	35,542	27,296	4,451	3,401
MAGPIE (ours)		EN	35,153	26,907	400	298
CoAM	MWEI	EN	780	489	521	385

Table 1: Dataset statistics used in our experiments, including language, sentence count, and annotated expressions.

pass a broader range of MWE types. It comprises 1,301 English sentences sourced from a variety of domains—including news articles, opinion pieces, TED talks, and web content—and features both written and transcribed spoken texts. Annotation followed a multi-phase process with dual annotators, expert adjudication, and automated validation. The dataset includes five MWE categories: NOUN, VERB, MODIFIER/CONNECTIVE, CLAUSE, and OTHER. It uses the PARSEME-TSV format (Savary et al., 2017), which supports discontinuous, overlapping, and multiple MWEs per instance. See Appendix K.2.1 for details.

To expand the multilingual coverage of our evaluation, we include Japanese and Turkish idiom identification in our analysis. For this purpose, we use the Dodiom dataset (Eryiğit et al., 2022), which was explicitly designed for idiom identification, and adapt the MWE Corpus (Hashimoto and Kawahara, 2009) into an idiom identification task following the same procedure used for MAGPIE (see Appendix A).

Using our best-performing prompt, GPT-4o-mini achieves F1 scores of approximately 34 on Japanese (290 instances) and 59 on Turkish (200 instances), with consistently low standard deviations across five random seeds.

Given these poor results, we did not pursue further experiments with these datasets.

3.3 Prompting the LLM

We approach the task of idiom identification by prompting an LLM, drawing on the growing trend of applying LLMs to non-generative extraction tasks (Liu et al., 2023; Sun et al., 2023; Smădu et al., 2024). Prompting is implemented via the LangChain framework, using OpenAI models for proprietary systems and Together-AI for open-source alternatives (referenced in Table 7).

Each model is initialized with a system prompt that defines the task and specifies the required output format; examples of these prompts are provided in Appendix K. Prompts are refined through iterative trial-and-error, aiming to maximize overall performance. Given the high cost and manual effort involved in prompt engineering, we use the smaller GPT-4o-mini model during development and transfer the optimized prompt to all other evaluated models. For the ID10M dataset, we use a single English-language prompt across all languages.

LangChain offers a way to get structured outputs from models that provide native APIs for structuring outputs, like tool/function calling or JSON mode. We use a Pydantic or TypedDict schema to structure the output of the model, for the convenience of parsing the answer to a usable dictionary. We provide the exact schema for each prompting method described below, in Appendix K. If a model fails to produce a response in the expected format, we treat the output as a hallucination and replace it with an empty list, indicating that no idiom was identified. This phenomenon is almost not observed with the stronger, larger models, like GPT-4o-mini. With smaller models like Qwen, this happens infrequently, at most once in every 100 calls. Additionally, we normalize the responses by removing any leading or trailing spaces and quotation marks.

3.3.1 Prompt Strategies

We assign a distinct name to each prompt variant used in our experiments and report results accordingly in the following section. The exact prompts are provided in Appendix K, and Figure 1 illustrates the evolution process of the different prompt strategies.

We start with the `zero-shot` prompt, where the model is simply instructed to list idioms in a given sentence. In these settings, no explanations are requested and no examples are given; it is only a direct instruction, followed by the input context and definitions. This is extended in the `few-shot` prompt by adding example pairs—half with idioms, half without—randomly selected from training data. Based on exploratory experiments (Appendix C), we use 10 examples for ID10M and MAGPIE, and 6 for CoAM. All prompts, including the `few-shot` and other more elaborate variants, build upon the basic `zero-shot` prompt, sharing the same core instruction and extending it with their respective specialized methods.

For both prompt types, we evaluate variants with and without self-consistency (`SC`) (Wang et al., 2023). In the `SC` setup, the model is sampled $n = 5$ times using the same prompt, and we retain only those idioms that appear in at least 50% of the outputs. In other words, we use majority vote over five runs to ensure stability.

We set the temperature to 0.3 when not using `SC` for stability, and to 0.8 with `SC` to encourage output diversity.

We further explore the chain-of-thought (`CoT`) prompting strategy, which has been shown to enhance LLM performance (Wei et al., 2023). Here, the model is instructed to reproduce the original sentence, followed by an explanation identifying potential idioms and justifying their inclusion or exclusion. This reasoning step is embedded in a dedicated JSON field preceding the final list of expressions, ensuring that the explanation informs the model's decision before generating the final output.

For the two idiom-identification datasets, we introduce two additional `CoT` variations:

`CoTGen`: The model is first instructed to list all potential PIEs found in the input text. For each, it then generates example sentences illustrating both idiomatic and literal usages. This is followed by a brief explanation, after which the model identifies which PIEs are actually used idiomatically in the original text, providing its final answer. The rationale behind this prompt is to guide the model through practicing both literal and idiomatic usages of the PIEs, which may help it identify similarities between the generated examples and the input sentence, thereby improving its ability to distinguish idiomatic usage in context.

`CoTBest`: Similar to `CoTGen`, the model begins by listing potential PIEs and providing an explanation for each. However, it is then instructed to select at most one idiom—the one it deems most confidently used idiomatically in the given context. The rationale is to reduce false positives by encouraging more conservative and precise identification.

As noted above, we extend our non-fine-tuning approach to the broader MWEI task, using the same overall methodology. To reflect the task's broader scope, we updated the instructions provided to the model accordingly. The prompts we use for this set of experiments are presented in Appendix K.2, where we also elaborate on the prompt optimization process of this step.

3.4 Evaluation

During evaluation, we lowercase both the input sentences and the predicted idioms or MWEs, and ignore dash ('-') characters—which commonly appear in MWEs—to avoid penalizing technically correct predictions due to formatting discrepancies. For the ID10M and MAGPIE datasets, model performance is measured using macro-F1 at the token level. To compute this, we convert the model's output (a list of idioms) into a BIO-tagged sequence by locating the idioms within the original, word-tokenized sentence. The code for this conversion, along with all other components necessary to reproduce our results, is available in our project repository.[3]

For CoAM, we use the MWE-based and token-based F1 metrics, following the definition provided by (Savary et al., 2017, 2023; Ide et al., 2024) (see Appendix D). The MWE-based F1 metric evaluates success based on exact surface match, meaning that even semantically correct predictions may be marked incorrect if they differ slightly in form. For instance, a gold annotation of "break the ice" would not match a prediction of "to break the ice", despite their clear equivalence.

To address this, token-based F1 allows partial overlaps, providing a more lenient and informative evaluation.

3.5 Models

To assess LLM capabilities in idiom identification, we compare a diverse set of models. Details on checkpoints and parameter sizes are in Appendix F, with licenses listed in Appendix J.

We include Open AI's GPT-4o as a strong closed-source model, alongside its smaller, cost-effective variant GPT-4o-mini, which we use for most experiments, including prompt and hyperparameter tuning. To compare with open-source models, we

[3] https://github.com/Intellexus-DSI/easy-as-pie

evaluate LLaMA-4-Scout, a 17B active parameter mixture-of-experts model, and Qwen2.5-72B (Qwen et al., 2025), used specifically for MAGPIE and CoAM.

We also test reasoning-focused models: GPT-o3-mini and DeepSeek-R1 (DeepSeek-AI et al., 2025). These are evaluated in a pure `zero-shot` setting, without chain-of-thought or self-consistency, to examine their built-in reasoning capabilities.

4 Results

We estimate about 700,000 API calls across all experiments, costing around $130. To balance robustness and cost, most runs used efficient models—GPT-4o-mini, LLaMA-4-Scout, and Qwen2.5-72B—each tested with three seeds and reported with mean and standard deviation. The more expensive GPT-4o was used only once with the best performing model configuration, and the reasoning models (GPT-o3-mini, DeepSeek-R1) were run once using `zero-shot` prompts.

4.1 Idioms Identification

Table 2 shows the results of all experiments on the ID10M dataset. We include the encoder-BiLSTM-CRF system as a baseline and its results, as originally reported by Tedeschi et al. (2022). This baseline model was trained on the task-specific training set, whereas our models operate in a `zero-shot` setting without any fine-tuning. The baseline models were trained on an automatically annotated training set, which may theoretically limit their competitiveness, despite their relatively strong performance.

Our results reveal several key trends. First, at least one of our model configurations outperforms the baseline in every language. Notably, this was often achieved using smaller, cost-efficient LLMs—except for German, where the best results required the larger GPT-4o model. `Few-shot` prompting consistently led to performance gains, suggesting that providing in-context examples of idiomatic usage aids model understanding. Across all languages, incorporating CoT reasoning further improved performance, though no single CoT variant consistently outperformed the others. Interestingly, self-consistency offered little benefit, indicating that LLM predictions in this task are relatively stable. Finally, reasoning-oriented models such as DeepSeek-R1 performed competitively even without explicit prompting strategies, with DeepSeek-R1 surpassing the baseline. We look into that in the following section, when we provide a deeper analysis. Figure 2 summarizes MAGPIE performance trends across model configurations, with full numerical results and standard deviations in Table 5 (Appendix E). This is the first use of MAGPIE for idiom identification rather than classification, so no direct baselines exist. We use this dataset to test whether patterns from ID10M hold. The results include precision and recall to provide a detailed view of model behavior. As with ID10M, `CoT` reasoning generally improves performance, though more complex variants like `CoTGen` and `CoTBest` sometimes reduce it, indicating that elaborate reasoning is not always beneficial. Reasoning-focused models achieve the best results, often outperforming CoT-prompted non-reasoning models. Self-consistency again shows little benefit, suggesting model outputs are stable across runs.

4.2 MWEs Identification

Table 3 shows the CoAM results, based on the best-performing configurations from the idiom identification task. To minimize cost, GPT-4o was run only once using the top `zero-shot + SC` setup (five runs total). The table also includes MaW and fine-tuned baselines from the original CoAM paper, averaged over three runs. For consistency, we re-calculated both metrics using the predictions from Ide et al. (2024), as token-based F1 was not originally reported. As shown in the table, LLMs perform below the fine-tuned Qwen-72B state-of-the-art, but our `zero-shot` configuration remains competitive, with GPT-4o outperforming some of the baseline supervised models (40.33 vs. 38.3). Token-based F1 scores are relatively higher, likely because LLMs tend to generate continuous spans that include gap words, which are penalized more harshly by MWE-based F1. Interestingly, `few-shot` prompts consistently decrease performance, while self-consistency was more helpful than in idiom identification. These trends suggest that LLMs behave less stably on MWEI, likely due to the flexible and inconsistent nature of MWE definitions. Without fine-tuning, our models struggle to align with annotation biases, contributing to the performance gap. Further analysis is provided in Section 4.3.

4.3 Qualitative Analysis

An example GPT-4o-mini response using `CoTBest` on the English ID10M dataset is shown in Ap-

Setting	Model	EN	DE	IT	ES
`Baseline` (Tedeschi et al., 2022)	ID10M	77.10	85.40	77.60	64.40
`Zero-shot`	GPT-4o-mini	83.92±0.71	70.68±0.76	68.65±0.78	70.74±0.18
	Llama-4-Scout	85.44±0.22	75.61±0.43	68.1±0.28	70.23±0.90
`Zero-shot+SC`	GPT-4o-mini	84.04±0.79	70.97±0.50	68.84±0.60	70.65±0.46
	Llama-4-Scout	85.4±0.11	75.49±0.67	68.99±0.56	70.21±0.32
`Zero-shot+SC+CoT`	GPT-4o-mini	86.59±0.37	81.6±0.60	75.66±0.28	71.09±0.44
	Llama-4-Scout	87.66±0.25	81.09±0.55	76.64±0.82	71.74±0.68
`Zero-shot+SC+CoTGen`	GPT-4o-mini	86.16±0.58	79.09±0.72	72.55±0.44	68.87±0.42
	Llama-4-Scout	86.94±0.11	80.87±0.13	77.16±0.00	71.26±0.80
`Zero-shot+SC+CoTBest`	GPT-4o-mini	87.88±0.24	79.28±0.58	72.80±0.24	71.60±0.90
	Llama-4-Scout	87.47±0.28	80.55±0.24	74.72±0.71	71.93±0.11
`Few-shot (10)`	GPT-4o-mini	88.75±0.60	78.02±3.56	76.46±1.20	73.32±0.44
	Llama-4-Scout	87.35±1.25	78.56±0.86	73.67±1.34	70.0±0.71
`Few-shot+SC`	GPT-4o-mini	88.73±0.34	78.5±3.84	76.58±1.16	73.19±0.73
	Llama-4-Scout	87.52±1.65	78.29±1.29	72.56±1.74	69.18±0.93
`Few-shot+SC+CoT`	GPT-4o-mini	89.91±0.43	84.15±0.74	78.37±0.88	73.11±1.17
	Llama-4-Scout	88.92±1.20	82.01±1.76	78.4±1.47	71.84±0.54
`Few-shot+SC+CoTGen`	GPT-4o-mini	90.54±0.64	80.5±1.30	76.97±1.36	73.03±0.92
	Llama-4-Scout	88.44±0.68	82.91±1.33	76.55±0.4	71.57±0.13
`Few-shot+SC+CoTBest`	GPT-4o-mini	**91.2±0.45**	81.17±1.71	78.3±2.24	74.44±0.29
	Llama-4-Scout	88.67±1.26	81.13±0.87	77.67±2.46	71.22±1.39
	GPT-4o	90.69	**87.80**	**82.20**	**82.36**
`Reasoning LLMs`	DeepSeek-R1	86.35	85.70	79.61	75.33
	GPT-o3-mini	86.88	81.47	73.70	71.31

Table 2: Results on the ID10M dataset (macro-F1). SC = Self-Consistency, CoT = Chain-of-Thought. The best result in each language is highlighted in bold.

pendix H. We conduct a manual analysis of the LLMs' responses to better understand the types of errors they make across the two tasks. Notably, we observe inconsistencies in the gold annotations across all datasets, particularly in the treatment of functional words. For example, in the sentence "The old computer just doesn't hold a candle to the latest models" the annotated idiom may appear either as "hold a candle to" or simply "hold a candle". Such variation introduces ambiguity in the evaluation and can penalize the model even when it produces a semantically accurate prediction.

4.3.1 Idiom Identification

We analyze a representative run of GPT-4o-mini using the `few-shot` configuration with `CoTBest` and `SC`. Table 4 presents examples from the ID10M dataset, illustrating the different error categories we identified.

We observe several recurring error types in model predictions. A common false positive is *literal as idiomatic*, where the model incorrectly labels a literal expression as idiomatic—indicating contextual misunderstanding. False negatives, or *missed predictions*, occur when idiomatic expressions are present but not detected; these are less frequent but still notable.

Boundary errors are particularly common and occur when the predicted span slightly misaligns with the gold annotation by including or omitting nearby words. Although these often reflect correct understanding, the strict BIO tagging penalizes even minor deviations—fully under the MWE-based metric and partially under the token-based

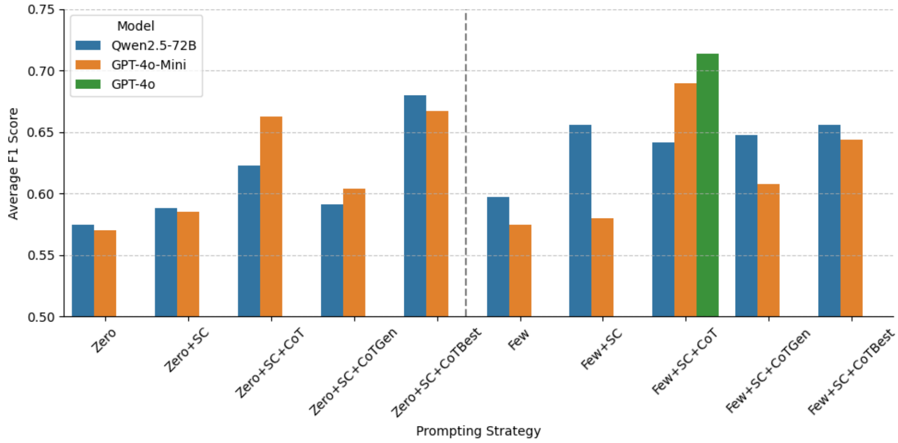

Figure 2: Average F1 score for different models and prompting strategies on MAGPIE.

one.

We also observe *modification errors*, where the model alters the idiom's surface form—typical of generative models aiming for well-formed output—despite prompts instructing it to preserve the original phrasing. Thus, we categorize these cases as failures to follow the instructions provided in the prompt. We calculate the percentage of such errors across all models per language, each running with our best-performing prompt. Overall, we see that the mean values are relatively low, with higher values in the more morphologically rich languages: German (2.6), Spanish (3.8), and Italian (3.8). In English, the score is 0.8, which can be interpreted as the model making fewer than one such error on average. Therefore, we believe that the models do follow our instructions given in the prompt.

Finally, we identify *false annotation* cases in the gold data, where idioms are mislabeled or missed entirely. Overall, we manually review the 200 English instances from the ID10M dataset, from which we identify 43 errors. Only two cases were classified with *literal as idiomatic* and four as *missed predictions*, reflecting low rates of major semantic errors. In contrast, we found 7 *false annotations* and 24 *boundary errors*, mostly involving minor function words like "to" or "a".

These findings suggest that some of the model's errors stem from technicalities in the evaluation method and are not rooted in its ability to fulfill the tasks. This implies that the model's actual performance is likely higher than the reported one. Overall, our analysis indicates that non-fine-tuned LLMs generally behave as rational and context-aware models in identifying idiomatic PIEs.

Language Differences. As shown in Table 2, model performance is generally strongest in English, which aligns with expectations given the predominance of English data in LLM pretraining. The *false annotation* error type appears more frequently in German. A native German speaker reviewed 40 instances of mismatch between model predictions and the gold labels, identifying 8 clear cases of *false annotation*. A similar review was conducted for Italian, where 33 errors were examined. This analysis revealed 5 cases of *false annotation* and 10 instances deemed ambiguous, reflecting some subjectivity in the annotation process. In Spanish, most models—except GPT-4o—scored around 70–75 F1. Further inspection showed that approximately 10% of Spanish samples involved *modification* errors, where the model altered the surface form of the idiom. These errors likely contributed to the relatively lower performance in that language.

4.3.2 MWEs vs Idioms

As discussed in Section 4, LLM performance declines when moving from idiom identification to the broader task of MWE identification. For example, under the `zero-shot+SC+CoT` setting, GPT-4o labeled expressions like "key to success", "comes to an end", and "sound and fury" as MWEs—none of which were annotated in the gold data, though one could argue they fit the definition. These cases highlight the ambiguity and definition-sensitivity of the task. Error types are equivalent to those observed in ID10M, with examples shown in Table 4. We further analyze the behavior of a reasoning model, DeepSeek-R1, which notably avoids many of those false positives. Inspecting its raw outputs,

		MWE-based-surface			Token-based-span		
Setting	**Model**	**P**	**R**	**F1**	**P**	**R**	**F1**
Baselines-FT	Llama-8B	**92.0**±2.00	14.4±2.40	24.9±3.70	**96.0**±1.65	14.2±1.86	24.75±2.87
	Llama-70B	69.0±1.70	26.8±5.10	38.3±5.20	78.67±3.21	30.0±4.24	43.2±4.16
	Qwen-7B	60.9±0.80	39.7±1.00	48.1±1.00	67.9±0.56	45.4±0.79	54.41±0.70
	Qwen-72B	63.8±2.20	**52.8**±2.10	**57.8**±1.80	70.2±1.48	**61.61**±0.76	**65.63**±0.59
Baselines-MaW	Rule	27.90	38.60	32.40	37.40	46.80	41.50
	Rule+BiEnc	47.9±0.80	36.5±0.30	41.4±0.30	56.4±0.44	39.1±0.30	46.2±0.10
	Rule+DCA	49.0±0.50	36.7±0.50	41.9±0.20	57.56±0.42	39.2±0.50	46.7±0.22
Zero-shot+SC+CoT	GPT-4o-mini	24.40±0.87	36.05±1.18	29.10±1.00	28.75±0.69	52.23±0.96	37.09±0.81
	Llama-4-Scout	24.50±0.45	45.84±0.69	31.93±0.53	27.26±0.62	59.07±0.73	37.30±0.72
	Qwen2.5-72B	26.08±0.84	36.22±1.64	30.32±1.14	29.66±0.76	53.89±2.08	38.26±1.14
	GPT-4o	37.16	44.09	40.33	40.52	54.80	46.59
Few-shot+SC+CoT	GPT-4o-mini	31.77±6.28	26.25±8.80	27.31±3.88	37.99±6.79	38.67±11.87	36.51±3.62
	Llama-4-Scout	24.14±1.24	42.26±6.49	30.48±1.25	27.32±1.17	54.80±7.24	36.23±0.81
	Qwen2.5-72B	28.42±2.00	21.61±0.99	24.52±0.98	34.53±1.36	36.42±2.42	35.41±1.16
Reasoning LLMs	DeepSeek-R1	51.73	46.98	49.24	53.72	54.80	54.26
	GPT-o3-mini	63.11	43.57	51.55	65.14	52.31	58.02

Table 3: Results on the CoAM dataset in precision (P), Recall (R), and macro-F1 for the two metrics: MWE-based and token-based. MaW =MWEasWSD (baseline), FT= fine-tuning, SC = Self-Consistency, CoT = Chain-of-Thought. We highlight the best result in each metric in bold.

Error Type	**Sentence**	**LLM's prediction**
	ID10M	
Literal as Idiomatic	This is *not my cup of tea*, it's yours, in fact I hate black tea.	not my cup of tea
Missed Prediction	That road comes to a *dead end* at the lake.	[]
Boundary Error	The old computer just doesn't *hold a candle* to the latest models	hold a candle to
Modification	Tu *serás un bombón* en un par de años.	ser un bombón
False Annotation	After passing the exam, I was on cloud nine. Und wollt *ihr wissen*, was davon das Ende ist? *Perché mai* dovrei pagare per un servizio che non utilizzo?	on cloud nine [] []
	CoAM	
False Detection	To prepare the world to...in a way that supports sustainable and equitable growth...	in a way
Missed Prediction	...from which *Prime minister* John Major's government never recovered.	[]
Boundary Error	However serious those crises *turn out* to be, historians a century...	turn out to be

Table 4: Examples of different error types. Ground truth expressions are shown in *italics* and highlighted in blue. Empty brackets in the LLM prediction column indicate that no MWEs or idioms were returned.

we find that its intermediate *thinking tokens* reflect a more structured reasoning process, allowing it to follow instructions and MWE definitions more precisely. The model appears to assess each candidate expression linguistically and semantically, resulting in more accurate predictions. An example is provided in Appendix I. We plan to further explore the use of reasoning models for MWE identification in future work.

5 Conclusions

In this paper, we introduced the first non-fine-tuning approach to the idiom identification task, an important challenge for many NLP applications. By designing prompts that guide LLMs to reason about idiomaticity—such as through chain-of-thought techniques—we achieved competitive results, including new state-of-the-art performance on one multilingual benchmark compared to supervised models. However, when extending to the broader task of MWE identification, our approach underperformed relative to fine-tuned baselines, highlighting the challenges posed by the flexible and inconsistent nature of MWEs and the need to account for annotation biases.

Limitations

While our study demonstrates the potential of LLMs to identify MWEs and idiomatic expressions in context, it also has several limitations. First, the ID10M dataset includes only 200 examples, all containing continuous idioms, which limits the variety of idiomatic constructions evaluated. Due to budget constraints, we restricted our experiments to a representative subset of configurations: we used three random seeds for the more cost-efficient models and only a single run for the more expensive ones. We also selected a limited set of models to represent broader model families, rather than exhaustively evaluating all available options. Since some of the models are proprietary, reproducibility depends on the availability and stability ensured by their providers. A potential concern with using LLMs on existing, publicly available tasks is that the models may have been exposed to the data during training. To partially address this, we include several different LLMs in our evaluation. Additionally, we did not include the PARSEME 1.3 corpus (Savary et al., 2023) in our evaluation due to its size and complexity. For the MAGPIE corpus (Haagsma et al., 2020), we evaluated on approximately 10% of the test set (400 samples), chosen to be representative but possibly insufficient to fully capture the dataset's linguistic diversity and ambiguity. Additionally, we could not find any baseline or state-of-the-art model to report results on this dataset in the capacity of the idiom identification task. Another limitation is that we manually optimized our prompts using smaller models and then reused the same prompt configurations across all models. While this approach allowed for consistent comparisons, it may not yield optimal performance for each individual model. Exploring automatic or model-specific prompt tuning could further improve results.

These limitations point to several promising directions for future work, including broader evaluations across more datasets, idiom types, and languages, as well as investigating prompt optimization strategies and model-specific tuning to enhance performance and generalization.

Ethics Statement

We used publicly available datasets and models under their licenses, details are listed in Appendix J. No personally identifiable information was processed, and no new data was collected. Our work is intended for research purposes only. We see no potential risk in our work.

Acknowledgments

This study is supported in part by the European Research Council (Intellexus, Project No. 101118558). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authorities can be held responsible for them.

We would also like to thank Dr. Ayako Nakamura-Walbeck and Dr. Nicola Bajetta for their assistance in the qualitative analysis process.

References

Christos Baziotis, Prashant Mathur, and Eva Hasler. 2023. Automatic evaluation and analysis of idioms in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3682–3700, Dubrovnik, Croatia. Association for Computational Linguistics.

Amir Cohen, Hila Gonen, Ori Shapira, Ran Levy, and Yoav Goldberg. 2022. Mcphrasy: Multi-context phrase similarity and clustering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3538–3550.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Survey: Multiword expression processing: A Survey. *Computational Linguistics*, 43(4):837–892.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2008. The vnc-tokens dataset. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions*, pages 19–22.

Francesca De Luca Fornaciari, Begoña Altuna, Itziar Gonzalez-Dios, and Maite Melero. 2024. A hard nut to crack: Idiom detection with conversational large language models. In *Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024)*, pages 35–44, Mexico City, Mexico (Hybrid). Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Rafael Ehren. 2017. Literal or idiomatic? identifying the reading of single occurrences of German multiword expressions using word embeddings. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–112, Valencia, Spain. Association for Computational Linguistics.

GülŞen Eryiğit, Ali Şentaş, and Johanna Monti. 2022. Gamified crowdsourcing for idiom corpora construction. *Natural Language Engineering*, 29(4):909–941.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564, Online. Association for Computational Linguistics.

Waseem Gharbieh, Virendra Bhavsar, and Paul Cook. 2016. A word embedding approach to identifying verb-noun idiomatic combinations. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 112–118, Berlin, Germany. Association for Computational Linguistics.

Hessel Haagsma, Johan Bos, and Malvina Nissim. 2020. MAGPIE: A large corpus of potentially idiomatic expressions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 279–287, Marseille, France. European Language Resources Association.

Hessel Haagsma, Malvina Nissim, and Johan Bos. 2019. Casting a wide net: Robust extraction of potentially idiomatic expressions. *Preprint*, arXiv:1911.08829.

Najet Hadj Mohamed, Agata Savary, Cherifa Ben Khelil, Jean-Yves Antoine, Iskandar Keskes, and Lamia Hadrich-Belguith. 2024. Lexicons gain the upper hand in Arabic MWE identification. In *Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD) @ LREC-COLING 2024*, pages 88–97, Torino, Italia. ELRA and ICCL.

Reyhaneh Hashempour and Aline Villavicencio. 2020. Leveraging contextual embeddings and idiom principle for detecting idiomaticity in potentially idiomatic expressions. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 72–80, Online. Association for Computational Linguistics.

Chikara Hashimoto and Daisuke Kawahara. 2009. Compilation of an idiom example database for supervised idiom identification. *Language Resources and Evaluation*, 43(4):355–384.

Yusuke Ide, Joshua Tanner, Adam Nohejl, Jacob Hoffman, Justin Vasselli, Hidetaka Kamigaito, and Taro Watanabe. 2024. CoAM: Corpus of all-type multiword expressions. *Preprint*, arXiv:2412.18151.

Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. 2013. SemEval-2013 task 5: Evaluating phrasal semantics. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 39–47, Atlanta, Georgia, USA. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Harish Tayyar Madabushi, Edward Gow-Smith, Carolina Scarton, and Aline Villavicencio. 2021. Astitchinlanguagemodels: Dataset and methods for the exploration of idiomaticity in pre-trained language models. *Preprint*, arXiv:2109.04413.

Maggie Mi, Aline Villavicencio, and Nafise Sadat Moosavi. 2024. Rolling the dice on idiomaticity: How llms fail to grasp context. *Preprint*, arXiv:2410.16069.

Filip Miletić and Sabine Schulte im Walde. 2024. Semantics of multiword expressions in transformer-based models: A survey. *Transactions of the Association for Computational Linguistics*, 12:593–612.

Vasudevan Nedumpozhimana and John Kelleher. 2021. Finding BERT's idiomatic key. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 57–62, Online. Association for Computational Linguistics.

Caroline Pasquer, Agata Savary, Carlos Ramisch, and Jean-Yves Antoine. 2020. Verbal multiword expression identification: Do we need a sledgehammer to crack a nut? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3333–3345, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dylan Phelps, Thomas Pickard, Maggie Mi, Edward Gow-Smith, and Aline Villavicencio. 2024. Sign of the times: Evaluating the use of large language models for idiomaticity detection. In *Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD) @ LREC-COLING 2024*, pages 178–187, Torino, Italia. ELRA and ICCL.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Agata Savary, Cherifa Ben Khelil, Carlos Ramisch, Voula Giouli, Verginica Barbu Mititelu, Najet Hadj Mohamed, Cvetana Krstev, Chaya Liebeskind, Hongzhi Xu, Sara Stymne, Tunga Güngör, Thomas

Pickard, Bruno Guillaume, Eduard Bejček, Archna Bhatia, Marie Candito, Polona Gantar, Uxoa Iñurrieta, Albert Gatt, and 9 others. 2023. PARSEME corpus release 1.3. In *Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)*, pages 24–35, Dubrovnik, Croatia. Association for Computational Linguistics.

Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, and Antoine Doucet. 2017. The PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 31–47, Valencia, Spain. Association for Computational Linguistics.

Nathan Schneider, Dirk Hovy, Anders Johannsen, and Marine Carpuat. 2016. SemEval-2016 task 10: Detecting minimal semantic units and their meanings (DiMSUM). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 546–559, San Diego, California. Association for Computational Linguistics.

Marco Senaldi, Yuri Bizzoni, and Alessandro Lenci. 2019. What do neural networks actually learn, when they learn to identify idioms? In *Proceedings of the Society for Computation in Linguistics*, volume 2, pages 310–313. University of Massachusetts Amherst.

Răzvan-Alexandru Smădu, David-Gabriel Ion, Dumitru-Clementin Cercel, Florin Pop, and Mihaela-Claudia Cercel. 2024. Investigating large language models for complex word identification in multilingual and multidomain setups. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16764–16800, Miami, Florida, USA. Association for Computational Linguistics.

Caroline Sporleder and Linlin Li. 2009. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 754–762, Athens, Greece. Association for Computational Linguistics.

Caroline Sporleder, Linlin Li, Philip Gorinski, and Xaver Koch. 2010. Idioms in context: The idix corpus.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Joshua Tanner and Jacob Hoffman. 2023. MWE as WSD: Solving multiword expression identification with word sense disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 181–193, Singapore. Association for Computational Linguistics.

Shiva Taslimipoor, Sara Bahaadini, and Ekaterina Kochmar. 2020. MTLB-STRUCT @parseme 2020: Capturing unseen multiword expressions using multi-task learning and pre-trained masked language models. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 142–148, online. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Simone Tedeschi, Federico Martelli, and Roberto Navigli. 2022. ID10M: Idiom identification in 10 languages. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2715–2726, Seattle, United States. Association for Computational Linguistics.

Su Nam Kim Timothy Baldwin. 2010. *Handbook of Natural Language Processing*, chapter 2:267-292.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Ziheng Zeng and Suma Bhat. 2021. Idiomatic expression identification using semantic compatibility. *Transactions of the Association for Computational Linguistics*, 9:1546–1562.

A MAGPIE Dataset Construction

The MAGPIE dataset was originally designed for idiom *classification*, where each instance consists of a five-sentence context extracted from a document. The target PIE appears in the middle sentence, and the task is to classify it as idiomatic or literal. Since our goal is *identification*, we adapt each instance by concatenating the five sentences into a single running text and annotating the PIE span in the middle sentence using BIO tags: B and I for the idiom, and O for all other tokens. AcknowledgmentsThis conversion introduces potential noise, as there may be other PIEs in the context that are not annotated. Evidently, we observe cases where the model correctly identifies idiomatic expressions that are not annotated in the gold data. As a result, the model is unfairly penalized for these correct predictions. However, we assume this noise is minimal and acceptable for our purposes. To reduce ambiguity, we exclude 60 instances (a negligible fraction of the dataset) in which the target PIE appears more than once in the text—as we do not know if they should be annotated as idiomatic or not. This filtering yields 4,391 test samples.

B Parsing Reasoning Model's Responses

In some cases, reasoning models fail to produce the desired structured output directly. We apply some regular-expression rules to extract the structured data from their raw response as a fallback. Specifically, we extract only the content that comes after "</think>" in DeepSeek-R1, and then we try to parse the rest to a JSON dictionary.

C Few-shot Ablation Test

Figure 3 shows an ablation test to find the best number of demonstrations provided to the model in `few-shot` settings. Specifically, we show an ablation on the English subset of the ID10M task, using GPT-4o-mini. We evaluate the average F1 score across 3 seeds, using $2, 4, \ldots, 12$ as candidates for the number of shots.

D MWE and Token Metrics

The MWE-based (exact-match) precision, recall, and F1 score follow the evaluation protocol introduced by Savary et al. (2017). Let G be the set of gold MWEs (in surface form), and H the set of predicted MWEs (hypotheses). The metrics are computed as:

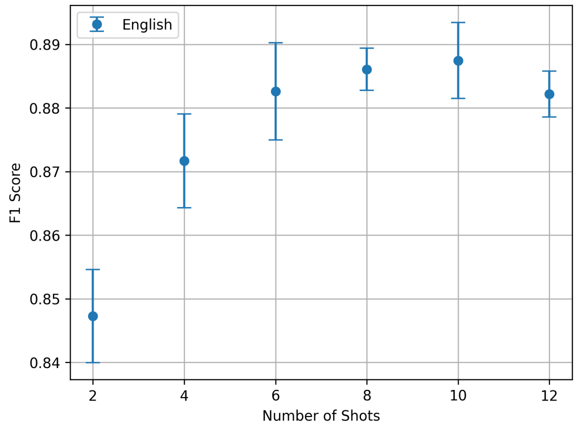

Figure 3: F1 score for English across different numbers of shots on the ID10M task. Error bars represent the standard deviation over 3 seeds.

$$\text{Recall} = \frac{|G \cap H|}{|G|} \tag{1}$$

$$\text{Precision} = \frac{|G \cap H|}{|H|} \tag{2}$$

The F1 score is the harmonic mean of precision and recall:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

For further details on this evaluation procedure for both MWE-based and Token-based, see Section 6 (Evaluation Measures) in Savary et al. (2017).

E MAGPIE Full Results

Table 5 provide the full results report on the MAGPIE 400-samples evaluation test we extracted.

F Model Checkpoints

z In Table 6, we present the checkpoints used in this work and the models' sizes.

G AI assistants

We used AI assistants (e.g., ChatGPT) to support code formatting, phrasing suggestions, and LaTeX styling during writing. All outputs were reviewed and edited by the authors. No content was directly generated or used without human verification.

H Response Example

We provide an example response of GPT-4-mini when prompted with `CoTBest` strategy on the ID10M dataset in English, see Figure 4.

Setting	Model	P	R	F1
`Zero-shot`	GPT-4o-mini	52.16±0.36	65.82±0.14	57.02±0.27
	Qwen2.5-72B	50.33±0.22	77.98±0.60	57.44±0.31
`Zero-shot+SC`	GPT-4o-mini	54.33±0.55	65.12±0.56	58.53±0.47
	Qwen2.5-72B	51.73±0.14	75.66±0.09	58.61±0.18
`Zero-shot+SC+CoT`	GPT-4o-mini	59.47±0.37	78.23±0.34	66.22±0.38
	Qwen2.5-72B	54.4±0.18	81.2±0.29	62.26±0.16
`Zero-shot+SC+CoTGen`	GPT-4o-mini	52.55±0.44	81.46±0.91	60.39±0.59
	Qwen2.5-72B	51.23±0.17	82.58±0.79	59.1±0.29
`Zero-shot+SC+CoTBest`	GPT-4o-mini	64.96±0.46	68.73±0.33	66.67±0.41
	Qwen2.5-72B	62.24±0.71	76.9±0.77	67.98±0.76
`Few-shot (10)`	GPT-4o-mini	57.4±1.09	58.14±5.61	57.44±1.97
	Qwen2.5-72B	58.99±1.13	74.94±1.32	64.93±0.64
`Few-shot+SC`	GPT-4o-mini	59.78±1.46	57.15±6.48	58±3.11
	Qwen2.5-72B	60.02±0.62	74.51±2.88	65.58±0.63
`Few-shot+SC+CoT`	GPT-4o-mini	66.73±1.11	71.59±2.08	68.91±0.41
	Qwen2.5-72B	56.11±1.17	82.51±0.74	64.16±1.14
	GPT-4o	64.57	82.58	**71.38**
`Few-shot+SC+CoTGen`	GPT-4o-mini	59.13±0.92	62.72±3.69	60.73±2.02
	Qwen2.5-72B	56.99±1.98	81.35±1.63	64.74±1.63
`Few-shot+SC+CoTBest`	GPT-4o-mini	**71.05**±0.71	59.87±2.85	64.37±1.85
	Qwen2.5-72B	57.88±1.30	81.37±1.59	65.57±0.91
`Reasoning LLMs`	DeepSeek-R1	54.46	**87.15**	63.28
	GPT-o3-mini	60.76	84.73	68.91

Table 5: Results on the MAGPIE task in macro-F1. SC = Self-Consistency, CoT = Chain-of-Thought. We mark the best results with bold numbers. Results are on our 400-samples test set.

I CoAM Response Analysis

We showcase an example where GPT-4o, a regular LLM, produces a false positive, but a reasoning model, DeepSeek-R1, was able to avoid it. We also present the full thinking process of R1 on this example in Figure 5.

J License

We list the models, datasets, and packages we use in this study, along with their respective licenses in Table 7.

K Prompts and Schema for each Prompting Method

We provide the system prompts combined with the user prompts for each task. The structured-output-schema for each prompt is also given. The examples part appears only in the `few-shot` settings and is based on the desired output structure.

For ID10M and MAGPIE, see Appendix K.1. For the final prompt of CoAM, see Appendix K.2. Notably, we experimented with more detailed prompts informed by the PARSEME annotation guidelines[4], including detailed descriptions of categories and decision criteria. However, somewhat surprisingly, this resulted in a drop in performance. For example, GPT-4o-mini achieved a token-based F1 score of 25.56 with the zero-shot prompt, but only 21.16 with the more elaborate prompt. Similarly, Llama-4 Scout scored 30.81 with the zero-shot prompt, compared to 26.99 with the elaborated version. We hypothesize that the complexity of the guidelines may overwhelm the models in a zero-

[4] https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.3/

Model	Checkpoint	# Parameters
DeepSeek-R1	deepseek-ai/DeepSeek-R1	671B
GPT-4o-mini	gpt-4o-mini-2024-07-18	>200B∗
GPT-4o	gpt-4o-2024-08-06	8B∗
GPT-o3-nini	o3-mini-2025-01-31	200B∗
Llama-4-Scout	meta-llama/Llama-4-Scout-17B-16E-Instruct	17B active 109B overall
Qwen2.5-72B	Qwen/Qwen2.5-72B-Instruct-Turbo	72B

Table 6: Checkpoints used during experiments and their number of parameters. ∗ = non-official estimation, as this information is not public.

CoT Response Example on ID10M - GPT-4o-mini

Input:
Sentence: A rock has broken the ice covering a lake.
Response:

```
{
    "sentence": "A rock has broken the ice covering a lake.",
    "potential_idioms": ["broken the ice"],
    "explanation": "In this context, 'broken the ice' is used literally to
describe a rock physically breaking the ice covering the lake. Therefore, it
is not used idiomatically or figuratively here.",
    "idioms": []
}
```

Figure 4: A response from GPT-4o-mini with CoT on ID10M.

shot setting. In light of this, we opted for a more concise prompt and focused our main efforts on idiom identification.

K.1 ID10M and MAGPIE

Figure 6 presents the system and user prompt used for the ID10M and MAGPIE datasets.

In addition, we illustrate the different schema types employed in our prompting strategies:

- **Regular Schema** (Figure 7)
- **Chain-of-Thought (CoT) Schema** (Figure 8)
- **CoTGen (Generation) Schema** (Figure 9)
- **CoTBest Schema** (Figure 10)

K.2 CoAM

The CoAM basic system and user prompt is given in Figure 11, along with both the JSON-based prompt (Figure 12) and the TSV-based one (Figure 13).

K.2.1 CoAM TSV-based Annotation and Prompt

CoAM utilizes the PARSEME-TSV format (Savary et al., 2017), which supports discontinuous, overlapping, and multiple MWEs per instance. Given a tokenized sequence (Here, tokenization refers to the process of segmenting text into linguistic units (words), performed as a preprocessing step using a method distinct from the model's internal tokenization) $x_1, \ldots, x_n$, where each x_i represents a word, the LLM is expected to output a list of MWEs, each defined by a span over the token sequence. The format uses two-column tab-separated value (TSV) format: the first column contains the tokens, and the second contains the corresponding labels. During inference, the input is provided with an empty label column, and the model is expected to assign labels to the tokens that are part of an MWE. Each distinct MWE in a sentence is labeled with a unique sequential identifier (e.g., 1, 2, ...), which is assigned to all tokens that belong to that MWE. We provide an example of how we use the TSV format

Full Response Example on CoAM - DeepSeek-R1

Sentence: The key to success will be the balance between modernity and CCP-led socialism.
Ground Truth: []
GPT-4o Preds: [key to success]
DeepSeek-R1 Preds: []
DeepSeek-R1 Raw Response: Okay, I need to identify all the multi-word expressions (MWEs) in the sentence: "The key to success will be the balance between modernity and CCP-led socialism." Let's break this down step by step.
First, I'll recall the definition of an MWE. It has to be a sequence of multiple words that are consistently used together with the same lexemes. The meaning shouldn't be deducible from the individual words (semantic idiomaticity), or there's some lexical or syntactic irregularity. Also, it shouldn't be a named entity.
Looking at the sentence:
"The key to success will be the balance between modernity and CCP-led socialism."
Starting with "key to success." The phrase "key to" is often followed by a noun, like "key to success," "key to happiness," etc. But does "key to success" qualify as an MWE? Let's check the conditions. The words are used together, but is the meaning idiomatic? "Key" here is metaphorical, meaning the essential element. However, the phrase "key to success" can be understood literally as the crucial element leading to success. The components (key, to, success) retain their individual meanings. So maybe it's not semantically idiomatic.
...
Are there any other phrases?
"Will be the balance" - "will be" is a standard verb phrase, not idiomatic. "The key to" as a phrase? Again, "key to" is a collocation but not necessarily an MWE unless it's part of an idiom.
For example, "key to the city" is a specific term, but "key to success" is more of a general phrase.
So, after analyzing each possible sequence, I don't find any MWEs that meet all three conditions.
The phrases present are either collocations, literal combinations, or part of named entities, which are excluded. Therefore, there are no MWEs in this sentence.

Figure 5: Full response of DeepSeek-R1 on CoAM.

in Figure 13.

ID10M & MAGPIE: System and user prompt

```
You are a professional linguist specializing in figurative language and your
task is to analyse sentences that may contain an idiom, also known as an
idiomatic expression. This is a definition of idiom: 'A phrase, expression, or
group of words that has a meaning different from the individual meanings of
the words themselves, and employed to convey ideas in a non-literal or
metaphorical manner'. Mark idioms only when their usage in the context is
idiomatic/figurative and let literal meanings remain unmarked. You are given
one sentence in {language}, you are an expert of this language.
If detected, write the idioms exactly as they are in the sentence, without any
changes. Only answer in JSON.
Human: Sentence: They've pissed off and left us in the lurch!
AI: idioms: [pissed off]
...
Human: Sentence: {sentence}
```

Figure 6: ID10M & MAGPIE: System and user prompt

ID10M & MAGPIE: Regular Schema

```
class Idioms(BaseModel):
    "Identified idioms in a sentence" idioms: list[str] = Field(description =
    "only the idioms in the sentence that are in figurative usage")
```

Figure 7: ID10M & MAGPIE: Regular structured output schema

Artifact	Type	License	Notes
CoAM[a]	Dataset	CC BY-NC-SA 4.0	Full test set used
ID10M[b]	Dataset	CC BY-NC-SA 4.0	Full test set used
MAGPIE[c]	Dataset	CC BY 4.0	Used a filtered subset
DeepSeek-R1[d]	Model	MIT License	Open-source
GPT-4o-mini[e]	Model	Proprietary	Accessed via OpenAI API
GPT-4o[f]	Model	Proprietary	Accessed via OpenAI API
GPT-o3 mini[g]	Model	Proprietary	Accessed via OpenAI API
LLaMA-4-Scout[h]	Model	Llama 4 License	
Qwen2.5-72B[i]	Model	MIT License	Open-source
LangChain[j]	Framework	MIT License	Used for prompting
Together AI[k]	Provider	Proprietary	Used for API access

[a] https://huggingface.co/datasets/yusuke196/CoAM
[b] https://github.com/Babelscape/ID10M/tree/master
[c] https://github.com/hslh/magpie-corpus/tree/master?tab=readme-ov-file
[d] https://api-docs.deepseek.com/news/news250120
[e] https://platform.openai.com/docs/models/gpt-4o-mini
[f] https://platform.openai.com/docs/models/gpt-4o
[g] https://platform.openai.com/docs/models/o3-mini
[h] https://ai.meta.com/blog/llama-4-multimodal-intelligence/
[i] https://huggingface.co/Qwen/Qwen2.5-72B-Instruct
[j] https://www.langchain.com/
[k] https://www.together.ai/

Table 7: License and usage summary of all datasets, models, and tools used in this study.

ID10M & MAGPIE: CoT Schema

```
class IdiomsCoT(BaseModel):
    "Identified idioms in a sentence" sentence: str = Field(description="the
    sentence you were provided with")
    explanation: str = Field(description="the explanation of the idioms in the
    sentence and if the usage is figurative or literal")
    idioms: list[str] = Field(description = "only the idioms in the sentence
    that are in figurative usage")
```

Figure 8: ID10M & MAGPIE: CoT structured output schema

ID10M & MAGPIE: CoTGen Schema

```
class IdiomsCoTGen(BaseModel):
    "Identified idioms in a sentence" sentence: str = Field(description="the
    sentence you were provided")
    potential_idioms: list[str] = Field(description="the potential idioms in
    the sentence")
    figurative_examples: list[str] = Field(description="3 generated examples
    of figurative usage of the idioms")
    literal_examples: list[str] = Field(description="3 generated examples of
    literal usage of the idioms")
    explanation: str = Field(description="the explanation of the idioms in the
    sentence and if the usage is figurative or literal")
    idioms: list[str] = Field(description="only the idioms in the sentence
    that are in figurative usage")
```

Figure 9: ID10M & MAGPIE: CoTGen structured output schema

ID10M & MAGPIE: CoTBest Schema

```
class IdiomsCoTBest(BaseModel):
    "Identified idioms in a sentence" sentence: str = Field(description="the
    sentence you were provided with")
    explanation: str = Field(description="the explanation of the idioms in the
    sentence and if the usage is figurative or literal")
    idioms: list[str] = Field(description="one idiom only. The best idiom -
    the one you are absolutely sure that appears in figurative usage")
```

Figure 10: ID10M & MAGPIE: CoTBest structured output schema

CoAM: System and user prompt

```
You are a helpful system to identify multiple-word expressions (MWEs).
Identify all the MWEs in the given sentence, and output their surface forms
exactly as they appear.
Here, an MWE is defined as a sequence that satisfies the following three
conditions.
1. It consists of multiple words that are always realized by the same lexemes.
Such words cannot be replaced without distorting the meaning of the expression
or violating language conventions.
2. It displays semantic, lexical, or syntactic idiomaticity. Semantic
idiomaticity occurs when the meaning of an expression cannot be explicitly
derived from its components. In other words, a semantically idiomatic takes on
a meaning that is unique to that combination of words. Lexical idiomaticity
occurs when one or more components of an expression are not used as
stand-alone words in standard English. Syntactic idiomaticity occurs when the
grammar of an expression cannot be derived directly from that of its
components. For example, semantically idiomatic MWEs include ”break up”, the
lexically idiomatic include ”to and from”, and the syntactically idiomatic
include ”long time no see”.
3. It is not a multi-word named entity, i.e., a specific name of a person,
facility, etc.
[FORMAT]
```

Figure 11: CoAM: System and user prompt

CoAM: FORMAT - Normal

```
Additional instructions:
- Be cautious: Only identify an expressions as MWEs if they clearly satisfies
the conditions above.
- When listing MWEs, use exactly the original surface form as it appears in
the sentence.
- Only answer in JSON.
Human: Sentence: International law does not know how to deal with him.
AI: mwes: [deal with] ...
Human: Sentence: {text}
```

Figure 12: CoAM: JSON-based prompt

CoAM: FORMAT - TSV

```
Output the sentence in TSV format, where each row contains a word and its MWE
tag. Assign the same numeric tag (starting from 1) to all words that belong to
the same MWE. Use 0 if the word is not part of any MWE. If a word belongs to
multiple MWEs, concatenate tags with semicolons (e.g., 2;5). Ensure that all
words in a valid MWE are tagged, even if they appear in separate lines.
Include the first word of the MWE, not just the idiomatic or fixed component.
Human: Sentence:
International
law
...
to
deal
with
him.
AI: mwes:
Word    MWE_tag
International    0
law    0
...
to    0
deal    1
with    1
...
Human: Sentence:
{text}
```

Figure 13: CoAM: TSV-based prompt

CoAM: Regular Schema

```
class MWEs(BaseModel):
    "Identified MWEs in a sentence."
    mwes: list[str] = Field(description="only the MWEs in the sentence that
    are following the given definition")
```

Figure 14: CoAM: Regular structured output schema

CoAM: CoT Schema

```
class MWEs(BaseModel):
    "Identified MWEs in a sentence."
    sentence: str = Field(description="the sentence you were provided with")
    explanation: str = Field(description="the explanation of the WMEs in the
    sentence and why they follow the given definition")
    mwes: list[str] = Field(description="only the MWEs in the sentence that
    are following the given definition")
```

Figure 15: CoAM: CoT structured output schema