CoAM: Corpus of All-Type Multiword Expressions

Anonymous ACL submission

Abstract

Multiword expressions (MWEs) refer to idiomatic sequences of multiple words. MWE identification, i.e., detecting MWEs in text, can play a key role in downstream tasks such as machine translation, but existing datasets for the task are inconsistently annotated, limited to a single type of MWE, or limited in size.

To enable reliable and comprehensive evaluation, we created CoAM: **Co**rpus of **A**ll-Type **M**ultiword Expressions, a dataset of 1.3K sentences constructed through a multi-step process to enhance data quality consisting of human annotation, human review, and automated consistency checking. Additionally, for the first time in a dataset of this kind, CoAM's MWEs are tagged with MWE types, such as NOUN and VERB, enabling fine-grained error analysis. Annotations for CoAM were collected using a new interface created with our interface generator, which allows easy and flexible annotation of MWEs in any form, including discontinuous ones.[1] Through experiments using CoAM, we find that a fine-tuned large language model outperforms the current state-of-the-art approach for MWE identification. Furthermore, analysis using our MWE type tagged data reveals that VERB MWEs are easier than NOUN MWEs to identify across approaches.

1 Introduction

Vocabulary plays a critical role in the comprehension of natural language. While often simplified as a collection of single words, vocabulary also includes idiomatic sequences known as multiword expressions (MWEs) (Baldwin and Kim, 2010), which form an important part of language knowledge (Jackendoff, 1995). We focus specifically on sequences whose meaning or grammatical structure cannot be derived from their constituent words, as they can impede text analysis or comprehension; that is, we exclude transparent collocations. For example, the bold sequences below constitute MWEs.

(a) *ACL **stands for** Association for Computational Linguistics.*

(b) *He has been **under the weather** lately.*

MWE identification (MWEI), the process of automatically tagging MWEs in text (Constant et al., 2017), is valuable for tasks where word-by-word text processing is insufficient, such as machine translation (Li et al., 2024) and lexical complexity assessment (Kochmar et al., 2020). Briakou et al. (2024) investigated step-by-step translation using a large language model (LLM) and demonstrated that identifying MWEs in the initial step improves the translation quality. Furthermore, MWEI enables applications, such as reading assistance systems, to perform automatic glossing, for example, by combining MWEI with definition modeling (Bevilacqua et al., 2020). For example, Huang et al. (2022) discusses how definition modeling for jargon, which largely consists of MWEs, can help laypeople comprehend specialized terminology.

Despite the importance of MWEI, existing datasets for the task are inconsistently annotated, limited to a single type of MWE, or limited in size, as described in Section 2. This hinders reliable and comprehensive evaluations of MWEI systems.

In this paper, we introduce the **Co**rpus of **A**ll-type **M**ultiword expressions (CoAM), a dataset of 1.3K sentences for MWEI. "Types" refer to MWE categories assigned based on the part of speech of the MWE, with "all-type" signifying the inclusion of all such categories of MWE. To ensure annotation quality, we assigned two annotators and a reviewer to each sentence and checked all annotations for consistency. Additionally, MWEs in the CoAM test set are tagged with their types, such as NOUN and VERB, facilitating fine-grained error analysis of MWEI systems. This enables us to address questions such as: "Are verbal MWEs—the focus of the long-running PARSEME project

[1] The CoAM dataset and source code of the interface generator will be publicly accessible upon acceptance.

(Savary et al., 2017)—more difficult to identify compared to other categories of MWEs?"

Annotations for CoAM were collected using a checkbox-based annotation interface created using our new interface generator, CAIGen: **C**heckbox-based **A**nnotation **I**nterface **Gen**erator. It allows easy and flexible annotation of MWEs in any form, including discontinuous MWEs such as *pick . . . up* in ***pick** me **up** at the station*, which are common in English and have historically been a challenge for MWEI systems (Rohanian et al., 2019).

Using CoAM, we evaluate two distinct MWEI approaches. The first approach, MWEasWSD (MaW, Tanner and Hoffman, 2023), combines a rule-based pipeline and a trainable bi-encoder model, achieving state-of-the-art performance on the DiMSUM dataset (Schneider et al., 2016). The second is LLM fine-tuning for MWEI, inspired by the effectiveness of similar approaches in tasks such as named entity recognition (NER, Zhou et al., 2024). We use CoAM to train and evaluate LLMs from the Llama (Dubey et al., 2024) and Qwen (Qwen Team, 2024) model families.

The results reveal that a fine-tuned Qwen model with 72B parameters greatly outperforms MaW, demonstrating the effectiveness of LLM fine-tuning. On the other hand, all approaches suffer from low recall (e.g., 50.7% for Qwen-72B). Further analysis shows that fine-tuned LLMs struggle more with detecting NOUN and CLAUSE MWEs than detecting VERB MWEs. MWEs not contained in WordNet (Miller, 1995), e.g., *real estate*, were found particularly difficult to identify, presumably because they are less widely recognized as MWEs.

2 Related Work

Datasets Previous studies have presented several datasets for MWEI and idiom[2] identification. The **DiMSUM** (Schneider et al., 2016) dataset consists of 5,799 sentences annotated with MWEs, including, but not limited to, verbal MWEs, noun MWEs, phatics, and multi-word (MW) proper nouns. Although DiMSUM has been used in multiple MWEI studies, e.g., Kirilin et al. (2016) and Liu et al. (2021), there exist inconsistencies in the annotation, which hinders proper evaluation. Tanner and Hoffman (2023) found that over 80% of the false positives of their system were actually caused by DiMSUM's inconsistent annotation. Next, the **PARSEME** corpus (Savary et al., 2017; Walsh et al., 2018) is a high-quality dataset for the identification of verbal MWEs. It has been continuously updated, and the latest version (1.3) contains over 455,000 sentences across 26 languages. However, their focus is limited to verbal MWEs. Lastly, **ID10M** (Tedeschi et al., 2022) is an idiom detection dataset consisting of automatically created training data in 10 languages and a manually curated evaluation benchmark of four languages. Their evaluation dataset was created with the help of professional annotators, but it contains only 200 sentences per language. They used Wiktionary as the source of idioms, and MWEs not in Wiktionary were skipped in the annotation. Additionally, they did not annotate discontinuous idioms.

Tagging Schemes ID10M uses the **BIO** scheme. DiMSUM uses the more flexible **6-tag** scheme, which allows discontinuous MWEs, but cannot handle overlapping MWEs (see Appendix E.2 for an example of overlapping MWEs). In contrast, the **parseme-tsv** format (Savary et al., 2017) accepts MWEs in any form, which leads us to adopt an equivalent data format.

Other Tasks Whereas these studies addressed MWE/idiom identification as a sequence tagging task, others worked on a related but different task, idiom usage recognition. It is a binary classification of word sequences in context as idiomatic/figurative or literal. The detection task in Muzny and Zettlemoyer (2013), SemEval-2013 Task 5b (Korkontzelos et al., 2013), SemEval-2022 Task 2 Subtask A (Tayyar Madabushi et al., 2022), and the MAGPIE corpus (Haagsma et al., 2020) are set up for this task. Our task setting is more realistic and challenging than these, requiring systems to identify all MWEs within a given sentence.

3 Task Formulation

We formulate the MWEI task as token-level sequence tagging, where the inputs are tokenized sentences and each token can belong to multiple MWEs. Given the token sequence of each sentence, $x_1, \ldots, x_n$, the task is to output a list of MWEs where each MWE is represented as a list of token indices. The i-th MWE in the sentence is represented with $[\mathrm{idx}(t_{i,1}), \mathrm{idx}(t_{i,2}), \ldots]$ where $t_{i,j}$ is the j-th token of the MWE and $\mathrm{idx}(t)$ is t's index. Our annotation interface allows annotations in this scheme, as illustrated in Figure 1.

[2] According to Tedeschi et al. (2022), idioms are a subset of MWEs.

4 Construction of CoAM

4.1 Data Selection and Preprocessing

In selecting the sources of our dataset, we prioritized sources aimed at general audiences so that they are in standard English and mostly free of grammatical errors. We also included both written and transcripted spoken texts. Consequently, we utilized the following four data sources (see Appendix A.1 for their details). **News** is news text written by professional writers sourced from EMM NewsBrief, which was introduced by Glavaš and Štajner (2013).[3] **Commentary** is commentaries on news from the WMT23 Shared Task monolingual training data (Kocmi et al., 2023). **TED** is a collection of TED talk transcriptions from two sources. (1) **NAIST** is the dataset by Neubig et al. (2014); (2) **IWSLT** is the subset of IWSLT 2017 Shared Task (Cettolo et al., 2017). **UD** is a collection of single sentences sourced from weblogs, reviews, question-answers, newsgroups, and emails in the English Web Treebank. It is part of the Universal Dependencies (Zeman et al., 2018) and the English PARSEME corpus (Walsh et al., 2018). We did not include any sentences from UD in the test set, because UD contains user-generated content with frequent grammatical errors. All the other sources were used both for training and test splits.

For all sources, we took the first 10 (or all, when there are less than 10) sentences from the article or talk and presented them in original order. Each sentence was tokenized using spaCy (we use version `3.7.1` and the `en_core_web_lg` model throughout this paper), with the exception of the UD sentences, which were already tokenized. Note that what we call *words* are tokens given in this manner.

4.2 Annotation Guidelines

The construction of a reliable dataset requires clear guidelines for annotators in order to minimize misannotation. However, we found no such guidelines for all-type MWE annotation, which led us to create new guidelines.

We define MWEs as idiomatic sequences that satisfy the following three conditions, based on the definition by the often-cited Baldwin and Kim (2010) and the PARSEME annotation guidelines.[4]

(a) **It consists of at least two words that are always realized by the same lexemes.** This condition means that *his* in *put yourself in his shoes* is not part of the MWE, as other words like *Michael's* can replace it.

(b) **It displays semantic, lexical, or syntactic idiomaticity.** Semantic idiomaticity occurs when the meaning of an expression cannot be derived from its constituent words. It is the most important type of idiomaticity because it accounts for the majority of MWEs being classified as idiomatic. See additional discussion on semantic idiomaticity in relation to non-compositionality in Appendix A.2. Note that transparent collocations such as *stuck at* are not an MWE in our definition because they are not semantically, lexically, or syntactically idiomatic. The meaning or grammatical structure of transparent collocations can be understood by their constituents, and thus they can be processed word-by-word.

(c) **It is not a proper noun, i.e., a specific name of a person, facility, and so on.** Previous MWE studies either classified MW proper nouns as MWEs (Schneider et al., 2016) or excluded them as non-idiomatic (Tayyar Madabushi et al., 2022). We opted for the latter, as proper nouns are linked to *encyclopedic* knowledge rather than *lexicographic* knowledge (Navigli and Ponzetto, 2012), to which typical MWEs belong to.

The full MWE definition is given in Appendix E. The guidelines were updated whenever an issue was found, e.g., an ambiguous description.

4.3 Annotation Interface

For flexible and efficient annotation, we developed a novel annotation interface generator, CAIGen. CAIGen builds a checkbox-based interface in Google Sheets[5], allowing annotators to perform annotations of MWEs and other kinds of spans simply by checking checkboxes as shown in Figure 1. In the interface generation process, for each sentence, CAIGen first writes the sentence ID, the sentence itself, and a bordered box to show the result of annotations. Then, it arranges each token from left to right, wrapping the line when its length hits a pre-set limit.

Its main advantages over other comparable annotation tools are summarized in Table 1. The first tool, **brat**[6] (Stenetorp et al., 2012), has been

[3] The dataset also contains data from WikiNews and Wikipedia, but we preferred EMM NewsBrief, which contains more MWEs according to our preliminary analysis based on the annotation by Kochmar et al. (2020).

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

[5] https://developers.google.com/sheets

[6] https://github.com/nlplab/brat

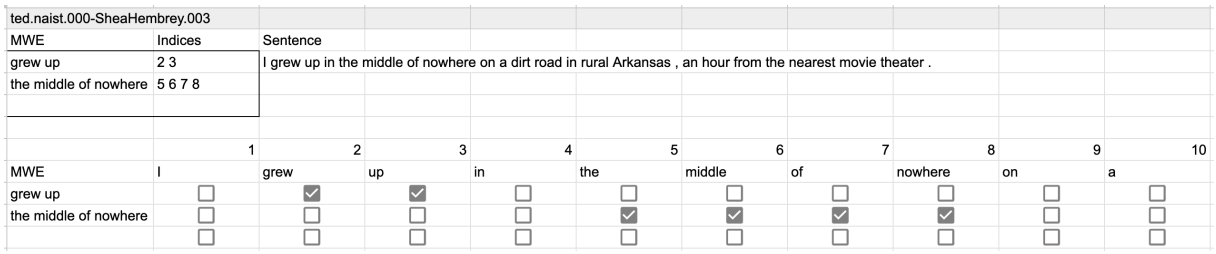

Figure 1: An example sentence presented in our interface by CAIGen. Checks in the checkboxes are instantly reflected in the bordered zone at the upper right, showing which MWEs are currently marked. The number of rows is reduced here for brevity; the actual interfaces we used have nine rows.

	Flexible annotation	Simple interface	Easy collaboration	Customizable interface
brat	✗	✓	✗	✗
FLAT	✓	✗	✗	✗
Ours	✓	✓	✓	✓

Table 1: Comparison of annotation tools.

used for annotation of a wide range of tasks such as sentiment analysis (Pontiki et al., 2016) and NER (Tabassum et al., 2020). **FLAT**[7] was used for the annotation of the PARSEME corpus (Savary et al., 2017). The advantages of CAIGen over these are as follows. First, the generated interface accepts annotations of spans of any form, including discontinuous or overlapping spans (see Appendix E.2 for examples), making it more flexible than the previous tools. Second, CAIGen provides a simple spreadsheet-based interface, which is familiar and accessible for technical and non-technical annotators alike. Third, CAIGen alleviates researchers' overhead of managing servers and annotator accounts by delegating to Google, while the other tools require either (1) distributing data slices to each annotator and having them run applications locally or (2) running and managing the application on a remote server. Lastly, our interface is highly customizable because CAIGen builds it using Google Apps Script, a programming language based on JavaScript; for example, simple customization would allow you to collect additional annotations about span types.

4.4 Annotation

The annotation was done by two annotators hired through a company and five authors. The hired annotators comprise one native and one non-native English speaker, both with at least six years of experience as translators. They received feedback from the authors after annotating the first 50 or so sentences to correct misunderstandings about the guidelines. We paid roughly 1 USD per sentence as a reward. Meanwhile, the author annotators consist of three native and two non-native speakers, all with a language-related degree and sufficient English proficiency to perform the task. For reliable annotation, we assigned each sentence to two annotators (one hired annotator and one author), ensuring that at least one was a native English speaker, as the task requires a rich English vocabulary.

All annotators were instructed to carefully read the guidelines and perform annotation using the checkbox-based interface. In the authors' annotation, we marked unclear sentences to remove them later because it is hard to determine whether a span is an MWE in such sentences.

4.5 Inter-Annotator Agreement

With two annotators being assigned to each sentence, we measured the inter-annotator agreement (IAA) by the MWE-based (exact-match) F1 score. The IAA was found to be only 37.3%, which could be attributed to annotators failing to annotate valid MWEs. We thus suggest that MWE datasets should be constructed with at least two independent annotators to reduce missed MWEs. Nonetheless, the IAA of CoAM is comparable with other MWEI datasets (see Section 8 for a detailed discussion).

4.6 Review

To solve the disagreement between the two annotators and correct any other problematic annotations, two authors—both native English speakers—reviewed all the annotations. We presented them with the annotations in a special interface for reviewing, where the tags given by only one annotator were highlighted, and we asked them to mark inappropriate tags to be deleted and newly

[7] https://github.com/proycon/flat

Type	PoS of Head	Description	Example
NOUN	NOUN, PRON, PROPN	Noun MWEs or compounds.	*the middle of nowhere, red tape*
VERB	VERB	Verbal MWEs.	*stand for, pick up, break a leg*
MOD/ CONN	ADJ, ADV, ADP, CCONJ, SCONJ	Adjectival, adverbial, or connective MWEs.	*under the weather, of course, in spite of*
CLAUSE	VERB, AUX	MWEs containing (1) a verb or auxiliary verb and (2) its nominal subject, such as phatics and proverbs.	*you're welcome, the early bird gets the worm, when it comes to*
OTHER	Any	MWEs whose head is not contained in them or whose PoS is none of the PoS above.	*and so on*

Table 2: MWE types in CoAM. The PoS are denoted with UPOS tags. Underlines denote the head of an MWE.

Sentence	Marked
Having never booked train tickets online before thought I would **give** it a **try** and was very surprised at how much I saved.	✓
Would recomend **giving** this a a **try**.	✗

Table 3: Inconsistent annotation of `give_try`. Note that the second sentence is reproduced as in DiMSUM, including apparent typos.

found MWEs to be added. When the review was complete, we updated our dataset according to the marks added by reviewers.

4.7 Consistency Check

One of the primary issues with the annotations in the DiMSUM dataset is their inconsistency—that is, a number of MWEs are annotated in one location and not another, despite equivalent constituents and semantics being present in both places. For example, see how `give_try` is labeled in only one of the sentences in Table 3.

In order to quantify this issue in both DiMSUM and CoAM—and to eliminate it from CoAM—we used a partially automated approach to find inconsistencies in both the entirety of CoAM and 1.3K sentences (the size of CoAM) randomly sampled from DiMSUM. We started by initializing an empty set M, then iterated through all given sentences and added all labeled MWEs from them to M. Next, we used a simple rule-based MWEI pipeline, repurposed from Tanner and Hoffman (2023), to find constituent groups in other sentences that could correspond to an MWE in M but were not already labeled. Finally, an author and native speaker of English reviewed each of these candidate constituent groups to see if they are semantically equivalent to already labeled instances of this MWE—that is, to see if they represent an inconsistency.

We found 118 instances of inconsistencies like this in the random sample of DiMSUM and 147 in CoAM,[8] reaffirming the difficulty involved in producing consistent MWE annotations. However, we then added the missing labels to all MWEs found in this way, eliminating these inconsistencies from the final CoAM data.

4.8 MWE Type Tagging

To enable fine-grained error analysis using CoAM, we automatically tagged all test-set MWEs, and training-set MWEs except those from UD, with MWE types. Inspired by the classification by Schneider et al. (2014b), we group MWEs into five types: NOUN, VERB, MODIFIER/CONNECTIVE (MOD/CONN), CLAUSE, and OTHER. The details are described in Table 2. We tag MWEs in each sentence through the following automatic operations after dependency parsing using spaCy. For each MWE $\boldsymbol{m} = (w_1, \ldots, w_{|\boldsymbol{m}|})$, we look for its syntactic head, namely, try to find w^* that has all the other words in $\boldsymbol{m}$ as its descendants. If $\boldsymbol{m}$ has such w^*, we determine its type based on the PoS of w^*; for example, we tag $\boldsymbol{m}$ as VERB when w^* is a verb. If $\boldsymbol{m}$ does not have such w^*, we tag $\boldsymbol{m}$ as OTHER. For the details, see the algorithm in Appendix A.3. We did not tag the sentences in UD because none of them are contained in the test set.

In order to measure the accuracy of our type tagging approach, we perform manual evaluation. The resulting accuracy was 89.3% (see Appendix A.4 for the details), which is sufficient for our purposes of performance analysis.

4.9 Statistics

Table 4 shows dataset statistics. CoAM has more than 1.3K sentences. The MWE density is 6–7%

[8] When counting inconsistencies, we always consider positive labels correct, and negative labels inconsistent.

	Sentences	Words	MWEs	MWE Density (%)	MWE Type Proportion (%)				
					NOUN	VERB	MOD/CONN	CLAUSE	OTHER
News	360	9,328	230	5.5	33.5	47.8	16.5	0.0	2.2
Commentary	357	9,310	272	7.0	29.8	30.5	29.4	1.1	9.2
TED	299	6,592	212	7.2	33.0	37.3	21.2	3.8	4.7
UD	285	5,001	160	7.2	-	-	-	-	-
Train	780	16,817	489	6.7	33.4	35.9	22.8	1.5	6.4
Test	521	13,414	385	6.5	30.6	40.0	22.9	1.6	4.9
Total	1,301	30,231	874	6.6	31.9	38.1	22.8	1.5	5.6

Table 4: Statistics of CoAM. MWE density is the percentage of words in MWEs. The test set comprises part of News, Commentary, and TED, while the training set comprises the rest of the data.

in both training and test sets. This proportion is much lower than that of DiMSUM (Schneider et al., 2016)—13% overall—most likely because DiMSUM includes proper noun phrases in MWEs. Among the five MWE types, VERB and NOUN are the most frequent ones across data sources. Our type tagging approach successfully assigned a specific (non-OTHER) MWE type to almost 95% of the MWEs. See Appendix A.5 for additional analysis regarding MWE continuity.

5 MWEI Approaches

We use CoAM to evaluate the following MWEI approaches.

5.1 MWEasWSD

MWEasWSD (MaW) is an approach that uses (1) an MWE lexicon—WordNet (Miller, 1995)—and a rule-based pipeline to identify MWE candidates and (2) a trainable model to filter MWE candidates (Tanner and Hoffman, 2023). It achieved state-of-the-art performance on the DiMSUM dataset. They published four of their filtering models, among which we use the bi-encoder (BiEnc) and DCA poly-encoder (DCA). Both models are based on BERT (Devlin et al., 2019), specifically `bert-base-uncased`. They have been trained with SemCor (Miller et al., 1993), and we further fine-tune each model with the CoAM training set. We also run MaW with the rule-based pipeline only, i.e., without a filtering model.

5.2 LLM Fine-Tuning

Recent studies have shown the effectiveness of LLM fine-tuning for a wide range of tasks, such as NER (Zhou et al., 2024) and grammatical error correction (Kaneko and Okazaki, 2023). Framing their task as sequence transduction, they achieved high performance by providing LLMs with prompts that included instructions on the task.

Inspired by their success, we perform LLM fine-tuning, where the inputs to LLMs are instructions for MWEI (a summary of the annotation guidelines) followed by formatted tokens in a sentence.

6 Experiments

6.1 Setup

LLM Fine-Tuning We use four instruction-tuned LLMs that are available on Hugging Face Hub: `Llama-3.1-8B-Instruct`, `Llama-3.1-70B-Instruct` (Dubey et al., 2024), `Qwen-2.5-7B-Instruct`, and `Qwen-2.5-72B-Instruct` (Qwen Team, 2024). We abbreviate them, e.g., to Llama-8B, omitting the versions. For efficient training and inference, we use QLoRA (Dettmers et al., 2023), performing 4-bit NormalFloat quantization and double quantization. Other hyperparameters are described in Appendix B.1. The computational budgets are in Appendix C.

To find a performant input-output format, we conduct preliminary experiments with three formats, adjusting the prompt for each format. These experiments are detailed in Appendix B.1. We find `tsv_to_tsv` (see Table 5) is the only format the models can comply with and thus adopt it.

All prompts contain a placeholder for a definition of MWEs to clarify what sequences should be marked. We report scores using what we refer to as the *long definition* in Table 6. For scores using a shorter definition, see the ablation study in Appendix B.1.

Evaluation Metrics We use MWE-based precision, recall, and F1 score (Savary et al., 2023). See Appendix B.2 for their exact definitions.

Role	Message
System	`You are a helpful system to identify multiple-word expressions (MWEs).`
User	`Identify all the MWEs in the given sentence, and output their surface forms and the indices of their components.\n` `\n` `[MWE_DEFINITION]\n` `\n` `Each sentence is given as a string of words delimited by '\n'. Respond in TSV format, where the first and second columns contain words and MWE tags, respectively. The MWE tag should be a string of MWE identifiers. When a word belongs to multiple MWEs, the tag should be a concatenation of their numbers delimited by semicolons.\n` `\n` `Sentence:\n` `ACL\n` `stands\n` ⋮

Table 5: Example prompt for fine-tuning, based on the `tsv_to_tsv` format. In our main experiments, `[MWE_DEFINITION]` will be filled with the long MWE definition described in Appendix B.1.

6.2 Results and Analysis

In Table 6a, the left three columns show the overall scores. Fine-tuned Qwen-72B achieves the best F1, surpassing the highest F1 by Rule+DCA of MaW. Moreover, fine-tuned Qwen-72B outperforms all MaW systems in precision and recall. This indicates the effectiveness of fine-tuning LLMs for MWEI—particularly LLMs with a large number of parameters. One explanation for this could be that knowledge about MWEs was acquired by LLMs through pre-training, and we can harness this knowledge for MWEI by fine-tuning.

Meanwhile, all systems except for the rule-based pipeline suffer from low recall. Even the best system, fine-tuned Qwen-72B, achieves a recall of only 50.7%, missing almost half of the gold MWEs.

Analysis of Recall Given the low recall across all systems, we conduct further analysis to determine which MWEs the models struggle to identify.

The right columns in Table 6a compare recall by MWE type, showing that CLAUSE and NOUN MWEs tend to be more difficult to identify than MOD/CONN or VERB MWEs across models.

In Table 6b, the left columns analyze how much recall changes depending on whether the MWE is seen or unseen, where an MWE in the test set is considered unseen if the multi-set of lemmas of its constituents was never annotated in the training set. We find that seen MWEs are easier to identify than unseen ones across models/systems. This result resembles that of the PARSEME shared task 1.2 (Ramisch et al., 2020), demonstrating that unseen MWEs remain challenging in MWEI.

The right columns in Table 6b reveal that MWEs not contained in WordNet are much more difficult to identify than those in WordNet. It is natural that MaW systems cannot identify the MWEs not in WordNet, as their rule-based pipeline cannot detect these candidates, but interestingly we find that fine-tuned LLMs also struggle to identify MWEs not in WordNet. For Qwen-72B, the recall gap between MWEs contained and not contained in WordNet is about 22 points, much larger than the gap between seen/unseen MWEs, about 12 points. We hypothesize that this is caused by MWEs in WordNet being more widely recognized as MWEs or idioms and that this recognition is reflected in the training data of the LLMs, enhancing their ability to identify these MWEs.

Table 7 shows examples of correctly identified MWEs and missed MWEs by fine-tuned Qwen-72B. VERB MWEs in WordNet like *fire up* are relatively easy to identify, with a recall of 79.0% achieved by the model. Meanwhile, the NOUN MWE *real estate* was not identified. The aforementioned hypothesis could explain this, as *real estate* is not in WordNet. Accurately identifying such multi-word entities or multi-word terms (Savary et al., 2019) could be a challenge for future studies.

Ablation Study on LLM Fine-Tuning Given the high performance of fine-tuned Qwen-72B, we investigate how much fine-tuning (FT) contributes to the performance by comparing FT to zero-shot learning (ZSL) and few-shot learning (FSL). In ZSL, we provide the models with the same prompt as FT. In FSL, we sample 5 pairs of input (sentence) and output (gold MWE set) from the training set, convert them into the `tsv_to_tsv` format, and include them in the prompt as examples. In the sampling process, we ensure that the models have a sufficient number of examples to learn the task and format by repeating random sampling until at least two of the example sentences contain one or more MWEs.

Table 8 shows the results. FT greatly outperforms ZSL and FSL, demonstrating the effective-

		F1	P	R	Recall by MWE Type			
					NOUN (118)	VERB (154)	MOD/CONN (88)	CLAUSE (6)
FT	Llama-8B	$29.4_{\pm 2.1}$	$\mathbf{82.6}_{\pm 1.2}$	$17.9_{\pm 1.6}$	$5.9_{\pm 0.8}$	$26.4_{\pm 1.5}$	$24.2_{\pm 3.5}$	$0.0_{\pm 0.0}$
	Llama-70B	$38.4_{\pm 4.8}$	$74.5_{\pm 2.6}$	$26.1_{\pm 4.6}$	$19.2_{\pm 4.2}$	$35.7_{\pm 5.7}$	$25.4_{\pm 5.6}$	$0.0_{\pm 0.0}$
	Qwen-7B	$45.2_{\pm 0.6}$	$63.2_{\pm 0.9}$	$35.2_{\pm 0.8}$	$26.0_{\pm 2.0}$	$47.2_{\pm 1.0}$	$31.8_{\pm 2.0}$	$0.0_{\pm 0.0}$
	Qwen-72B	$\mathbf{55.5}_{\pm 0.5}$	$61.5_{\pm 2.6}$	$\mathbf{50.7}_{\pm 2.5}$	$\mathbf{44.4}_{\pm 3.2}$	$\mathbf{60.6}_{\pm 1.0}$	$\mathbf{50.4}_{\pm 5.1}$	$\mathbf{22.2}_{\pm 9.6}$
MaW	Rule	32.7	28.3	38.7	37.3	40.9	47.7	0.0
	Rule+BiEnc	$41.6_{\pm 0.1}$	$48.6_{\pm 0.3}$	$36.5_{\pm 0.3}$	$32.2_{\pm 0.0}$	$41.1_{\pm 0.7}$	$44.3_{\pm 0.0}$	$0.0_{\pm 0.0}$
	Rule+DCA	$42.0_{\pm 0.1}$	$48.4_{\pm 0.6}$	$37.1_{\pm 0.3}$	$33.3_{\pm 0.5}$	$40.9_{\pm 0.0}$	$45.8_{\pm 0.7}$	$0.0_{\pm 0.0}$

(a)

		Recall by Seen/Unseen		Recall by In WN or Not	
		Seen (138)	Unseen (247)	True (163)	False (222)
FT	Llama-8B	$37.2_{\pm 3.4}$	$7.2_{\pm 0.6}$	$29.2_{\pm 1.9}$	$9.6_{\pm 1.7}$
	Llama-70B	$35.0_{\pm 7.3}$	$21.1_{\pm 3.2}$	$46.8_{\pm 6.5}$	$10.8_{\pm 3.2}$
	Qwen-7B	$44.4_{\pm 0.8}$	$30.0_{\pm 0.8}$	$50.1_{\pm 1.4}$	$24.2_{\pm 0.5}$
	Qwen-72B	$\mathbf{58.2}_{\pm 5.8}$	$\mathbf{46.6}_{\pm 0.7}$	$63.6_{\pm 3.4}$	$\mathbf{41.3}_{\pm 1.9}$
MaW	Rule	47.8	33.6	**91.4**	0.0
	Rule+BiEnc	$44.2_{\pm 0.0}$	$32.1_{\pm 0.5}$	$86.1_{\pm 0.7}$	$0.0_{\pm 0.0}$
	Rule+DCA	$45.4_{\pm 0.8}$	$32.4_{\pm 0.0}$	$87.5_{\pm 0.7}$	$0.0_{\pm 0.0}$

(b)

Table 6: Results by MWEI system and metric/MWE category, as mean percentage scores of three runs with random training seeds. The numbers in parentheses are the MWE counts. ± denotes standard deviation. The bold font denotes the highest score. *Rule* stands for the rule-based baseline, and *WN* for WordNet.

Result	MWE	Context	Note
TP	*fire up*	*The allegations have **fired up** the opposition, . . .*	VERB MWE in WordNet
TP	*at least*	*. . . since **at least** the 1950s.*	MOD/CONN MWE in WordNet
FN	*real estate*	*. . . park their toxic **real estate** assets . . .*	NOUN MWE not in WordNet
FN	*you know*	***You know**, it's very old . . .*	CLAUSE MWE not in WordNet

Table 7: Examples of true positives (TPs) and false negatives (FNs), i.e., MWEs identified/missed in all three runs, of fine-tuned Qwen-72B.

	Llama-70B	Qwen-72B
FT	$38.4_{\pm 4.8}$	$55.5_{\pm 0.5}$
FSL	$4.3_{\pm 0.9}$	$14.1_{\pm 0.3}$
ZSL	6.9	2.8

Table 8: Ablation results by MWEI method and model, as mean percentage F1 scores of three runs (for FT the randomness arises in training, and for FSL in exemplar selection). ± denotes standard deviation.

ness of FT on the CoAM training set.

Comparison considering computational efficiency As shown in Table 6a, fine-tuned Qwen-72B outperforms the best MaW system in all of F1, precision, and recall. However, the LLM takes substantial memory and compute, and MaW has advantages in this regard. See Appendix D for more details.

7 Conclusion

In this paper, we constructed CoAM, a high-quality dataset of 1.3K sentences for MWEI covering all types of MWEs. Using a combination of human review and automated consistency checking, we addressed consistency issues that have been a problem for previous MWE datasets, enabling more accurate evaluation results for future work in MWEI. We used CoAM to evaluate two MWEI approaches: MaW and LLM fine-tuning. Our largest fine-tuned LLM performed the best, outperforming the current state-of-the-art, but all systems suffered from low recall. Consequently, we argue that MWEI overall remains a challenging task.

8 Limitations

Inter-Annotator Agreement (IAA) Our initial annotations on CoAM suffered from a low IAA of 37.3%. However, this IAA (F1 score) is computed in a strict way, which we chose for its clear interpretability. CoAM's IAA is comparable with those of other datasets when the same, more lenient methods are used. PARSEME 1.1 (Ramisch et al., 2018) computes F1 for exact span matches as we do, but they only report "the highest scores among all possible annotator pairs" (see Table 1, ibid.). This highest score for English PARSEME is 52.9%, similar to the highest score computed in the same way for CoAM: 52.2%. Meanwhile, DiMSUM was annotated by a single annotator, and thus no actual agreement was reported. Although Schneider et al. (2016) estimate DiMSUM's IAA to range from 60% to 75%, this is based on a very small subset (66 sentences) reannotated by the first author and computed as a more lenient F1 score based on partial span matches. Thus, while our reported IAA value is low, we emphasize that our IAA is in line with comparable MWEI datasets. Additionally, we addressed the low IAA of our dataset through the combination of human review (Section 4.6) and consistency checking (Section 4.7).

Dataset Size Because the objective of this work was to maximize dataset quality, we invested heavily in efforts to improve data quality, employing multiple annotators per sentence, manual review, and consistency checking. This focus on quality over quantity resulted in a slightly smaller dataset size (1.3K sentences) compared to some previous works like DiMSUM and PARSEME. However, the evaluation set of CoAM has many more sentences than the gold data of ID10M (Tedeschi et al., 2022), and is large enough for our purpose of reliably evaluating MWEI systems.

CAIGen We observed a usability issue with our annotation interface, CAIGen. It requires annotators to use separate table rows for each annotated MWE. Because annotators may forget to use separate rows, which happened in the construction of CoAM, researchers should urge them to confirm their annotations are as intended after finishing each sentence.

9 Ethical Considerations

All sources of CoAM are permitted at minimum for use in research, as described in Appendix A.1. CoAM itself will be released under the condition that the users do not publish the data on the open web to prevent its leakage to the training data of future models. All annotators of CoAM consented to the publication of their annotations. Throughout the dataset construction processes, we found no contents harmful to the environment, specific groups of people, or privacy.

10 Acknowledgments

To paraphrase or polish authors' original content, we used ChatGPT[9] as an assistant tool.

References

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In *Handbook of Natural Language Processing*, pages 267–292, Boca Raton. CRC Press.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English web treebank. LDC2012T13.

Eleftheria Briakou, Jiaming Luo, Colin Cherry, and Markus Freitag. 2024. Translating step-by-step: Decomposing the translation process for improved translation quality of long-form texts. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1301–1317, Miami, Florida, USA. Association for Computational Linguistics.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. In *EAMT*, pages 261–268.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning

[9] https://chatgpt.com

of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Goran Glavaš and Sanja Štajner. 2013. Event-centered simplification of news stories. In *Proceedings of the Student Research Workshop associated with RANLP 2013*, pages 71–78, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

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.

Jie Huang, Hanyin Shao, Kevin Chen-Chuan Chang, Jinjun Xiong, and Wen-mei Hwu. 2022. Understanding jargon: Combining extraction and generation for definition modeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3994–4004, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ray Jackendoff. 1995. The Boundaries of Lexicon. In M. Everaert, E.J. van der Linden, A. Schenk, R. Schreuder, and R. Schreuder, editors, *Idioms: Structural and Psychological Perspectives*, pages 133–166. Erlbaum, NJ.

Masahiro Kaneko and Naoaki Okazaki. 2023. Reducing sequence length by predicting edit spans with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10017–10029, Singapore. Association for Computational Linguistics.

Angelika Kirilin, Felix Krauss, and Yannick Versley. 2016. ICL-HD at SemEval-2016 task 10: Improving the detection of minimal semantic units and their meanings with an ontology and word embeddings. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 937–945, San Diego, California. Association for Computational Linguistics.

Ekaterina Kochmar, Sian Gooding, and Matthew Shardlow. 2020. Detecting multiword expression type helps lexical complexity assessment. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4426–4435, Marseille, France. European Language Resources Association.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

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.

Shuang Li, Jiangjie Chen, Siyu Yuan, Xinyi Wu, Hao Yang, Shimin Tao, and Yanghua Xiao. 2024. Translate meanings, not just words: Idiomkb's role in optimizing idiomatic translation with language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Nelson F. Liu, Daniel Hershcovich, Michael Kranzlein, and Nathan Schneider. 2021. Lexical semantic recognition. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 49–56, Online. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38(1):39–41.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Grace Muzny and Luke Zettlemoyer. 2013. Automatic idiom identification in Wiktionary. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1417–1421, Seattle, Washington, USA. Association for Computational Linguistics.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Graham Neubig, Katsuhiro Sudoh, Yusuke Oda, Kevin Duh, Hajime Tsukuda, and Masaaki Nagata. 2014. The NAIST-NTT TED talk treebank. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Papers*, pages 265–270, Lake Tahoe, California.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, Ion Androutsopoulos, Núria Bel, and Gülşen Eryiğit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International Workshop on Semantic Evaluation*.

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

Carlos Ramisch, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, Tunga Güngör, Abdelati Hawwari, Uxoa Iñurrieta, Jolanta Kovalevskaitė, Simon Krek, Timm Lichte, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Behrang QasemiZadeh, Renata Ramisch, Nathan Schneider, Ivelina Stoyanova, Ashwini Vaidya, and Abigail Walsh. 2018. Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 222–240, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Carlos Ramisch, Agata Savary, Bruno Guillaume, Jakub Waszczuk, Marie Candito, Ashwini Vaidya, Verginica Barbu Mititelu, Archna Bhatia, Uxoa Iñurrieta, Voula Giouli, Tunga Güngör, Menghan Jiang, Timm Lichte, Chaya Liebeskind, Johanna Monti, Renata Ramisch, Sara Stymne, Abigail Walsh, and Hongzhi Xu. 2020. Edition 1.2 of the PARSEME shared task on semi-supervised identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 107–118, online. Association for Computational Linguistics.

Omid Rohanian, Shiva Taslimipoor, Samaneh Kouchaki, Le An Ha, and Ruslan Mitkov. 2019. Bridging the gap: Attending to discontinuity in identification of multiword expressions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2692–2698, Minneapolis, Minnesota. Association for Computational Linguistics.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions : a pain in the neck for nlp.

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, Jolanta Kovalevskaite, Timm Lichte, Nikola Ljubešić, Johanna Monti, Carla Parra Escartín, Mehrnoush Shamsfard, Ivelina Stoyanova, Veronika Vincze, and Abigail Walsh. 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, Silvio Cordeiro, and Carlos Ramisch. 2019. Without lexicons, multiword expression identification will never fly: A position statement. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 79–91, Florence, Italy. 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, Emily Danchik, Chris Dyer, and Noah A. Smith. 2014a. Discriminative lexical semantic segmentation with gaps: Running the MWE gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.

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.

Nathan Schneider, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T Mordowanec, Henrietta Conrad, and Noah A Smith. 2014b. Comprehensive annotation of multiword expressions in a social web corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 455–461. European Language Resources Association (ELRA).

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Jeniya Tabassum, Mounica Maddela, Wei Xu, and Alan Ritter. 2020. Code and named entity recognition in stackoverflow. In *Annual Meeting of the 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.

Harish Tayyar Madabushi, Edward Gow-Smith, Marcos Garcia, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2022. SemEval-2022 task 2: Multilingual idiomaticity detection and sentence embedding. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 107–121, Seattle, United States. Association for Computational Linguistics.

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.

Abigail Walsh, Claire Bonial, Kristina Geeraert, John P. McCrae, Nathan Schneider, and Clarissa Somers. 2018. Constructing an annotated corpus of verbal MWEs for English. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 193–200, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. Ontonotes release 5.0. LDC2013T19.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations*.

Source	Paper	URL	License	Note
News	Glavaš and Štajner (2013); Yimam et al. (2018)	https://takelab.fer.hr/data/evsimplify/ https://sites.google.com/view/cwisharedtask2018/datasets	Creative Commons Attribution-NonCommercial-ShareAlike 3.0	We used the News_Train data from the CWI Shared Task 2018 datasets.
Commentary	Kocmi et al. (2023)	https://data.statmt.org/news-commentary/v18.1/training-monolingual/	Can be freely used for research purposes.	We used the English part of v18.1.
TED (NAIST-NTT)	Neubig et al. (2014)	https://ahcweb01.naist.jp/old/resource/tedtreebank/	Creative Commons ShareAlike-Attribution-NonCommercial	
TED (IWSLT)	Cettolo et al. (2012, 2017)	https://wit3.fbk.eu/2017-01 https://wit3.fbk.eu/2017-01-b	Creative Commons Attribution-NonCommercial-NoDerivs 3.0	
UD	Bies et al. (2012); Walsh et al. (2018)	https://gitlab.com/parseme/parseme_corpus_en	Creative Commons ShareAlike 4.0	We used the portion of PARSEME corpus from the English Web Treebank.

Table 9: Data sources of CoAM.

A Construction of CoAM

A.1 Data Sources

See Table 9.

A.2 Notes on Idiomaticity and (Non-)Compositionality

To judge whether a sequence is an MWE, we focus on (semantic) idiomaticity instead of non-compositionality, although non-compositionality has been considered an inherent property of MWEs in previous studies, such as Tedeschi et al. (2022). To discuss the difference between the two notions, let us consider the expression *spill the beans*. It is deemed compositional because it can be analyzed as being made up of *spill* in a "reveal" sense and the *beans* in a "secret(s)" sense, resulting in the overall compositional reading of "reveal the secret(s)" (Sag et al., 2002). Meanwhile, the "secret(s)" sense is unique to the expression and cannot be derived from the word *beans*, making the whole expression semantically idiomatic. We argue that *spill the beans* should be identified as an MWE because the meaning of *beans* depends on the whole expression and that any idiomatic sequences should be identified even if they are compositional.

A.3 Algorithm for MWE Type Tagging

We tag MWEs in CoAM by the algorithm shown in Algorithm 1.

A.4 Evaluation of MWE Type Tagging

The automated type tagging we employ (see Section 4.8) is based on the PoS of the syntactic head of the MWE. Both PoS tagging and syntactic parsing are performed using spaCy. As all three steps (PoS tagging, parsing, and the type assignment based on them) are potential sources of errors, we evaluated the accuracy of the tags.

We took a random sample of 30 instances per type (all 11 instances of the CLAUSE type), split 15:15 between the training set and the test sets, for a total sample size of 131 instances and manually verified the type of each MWE. The resulting accuracy was 89.3%, which is sufficient for the purposes of performance analysis (presented in Section 6.2). Some of the inaccuracies resulted from the whole expression having a different PoS than the head (e.g., "so-called" was assigned to VERB instead of MOD/CONN), some resulted from errors in the automated PoS tagging and parsing (e.g., "duck and cover" was assigned to NOUN instead of VERB). The inaccuracies were spread across NOUN, VERB, and OTHER. The corresponding incorrect types were spread across VERB, MOD/CONN, and OTHER, and they did not disproportionately affect any particular type.

A.5 Statistics Regarding MWE Continuity

Discontinuous MWEs are common in English and previous work has stressed the importance of iden-

Algorithm 1 Algorithm of MWE Type Tagging

Input: MWE $\boldsymbol{m}$
Output: Type t

```
for all w_i ∈ m do
    if w_i = w* then
        if PoS of w_i ∈ {NOUN, PRON, PROPN} then
            if children of w_i include a relative clause modifier that is verb then
                t ← VERB                                   ▷ E.g., . . . the price he pays . . .
            end if
            t ← NOUN                                       ▷ E.g., . . . born in the middle of nowhere . . .
        else if PoS of w_i ∈ {VERB, AUX} then
            if nominal subject of w_i ∈ m then
                t ← CLAUSE                                 ▷ E.g., . . . when it comes to climate change . . .
            else if PoS of w_i = VERB then
                t ← VERB                                   ▷ E.g., . . . the expression stands for . . .
            else if PoS of w_i = AUX then
                t ← OTHER (other PoS)                      ▷ E.g., . . . is off to an early start . . .
            end if
        else if PoS of w_i ∈ {ADP, ADJ, ADV, CCONJ, SCONJ} then
            t ← MOD/CONN                                   ▷ E.g., . . . for at least two decades . . .
        else
            t ← OTHER (other PoS)
        end if
    else
        t ← OTHER (head not in MWE)                        ▷ E.g., . . . , and so on . . .
    end if
end for
return t
```

MWE Type	Discontinuous (%)	Example
Noun	4.4	. . . *a* ***suicide*** *car* ***bomber*** *and Taliban militants* . . .
Verb	19.1	. . . ***turned*** *four aircraft* ***into*** *cruise missiles* . . .
Mod/Conn	6.1	. . . concentration of power **in** his own **hands**.
Clause	0.0	
Other	17.5	. . . *which* ***side*** *of these we'd like to* ***be on***.

Table 10: Ratio of discontinuous MWEs in CoAM by MWE type.

tifying them (Schneider et al., 2014a; Rohanian et al., 2019). Table 10 shows the ratio of discontinuous MWEs by MWE type. VERB MWEs have the highest proportion of discontinuous expressions, while MOD/CONN, NOUN, and OTHER MWEs reach lower values.

LoRA	r	64
	α	16
	Dropout	0.05
	Target modules	All linear layers
Training	Epoch	3
	Effective batch size	32
	Learning rate	2e-4
	Learning rate scheduler	constant
	Optimizer	paged_adamw_8bit (β_2 = 0.999)
	Max grad norm	0.3

Table 11: Hyperparameters for fine-tuning. The epoch was determined based on preliminary experiments, while other parameters are based on Dettmers et al. (2023).

Name	Example (Input → Output)		
dict_to_dict_list	`{1: 'ACL', 2: 'stands', 3: 'for', ...}`	→	`[{'surface': 'stands for', 'indices': [2, 3]}]`
str_to_str_ number_span	`ACL stands for Association for Computational Linguists .`	→	`ACL <1>stands for</1> Association for Computational Linguists .`
tsv_to_tsv	`ACL\n stands\n for\n` ⋮	→	`ACL\t\n stands\t1\n for\t1\n` ⋮

Table 12: Input-output formats for fine-tuning. Bold fonts denote the format employed for the main experiments. Suppose the sentence given as the example is *ACL stands for Association for Computational Linguistics*.

B Experimental Setup

B.1 LLM fine-tuning

Hyperparameters Table 11 shows hyperparameters used for training and evaluation. At inference, we perform greedy decoding.

Input-Output Format To investigate the optimal input-output format, we perform preliminary experiments. We prepare three input-output formats described in Table 12. All the output formats allow us to represent MWEs in any form, including discontinuous or overlapping MWEs. We train and evaluate Llama-8B and Qwen-7B with CoAM, using each input-output format. We use the hyperparameters described in Table 11 and the long MWE definition (same as the main experiments).

As a result, `tsv_to_tsv` turned out to be the only format the models can comply with. Employing other formats results in the violation of the format despite carefully written instructions. With `dict_to_dict_list`, the models produce wrong indices in the outputs. With `str_to_str_number_span`, the models delete spaces before punctuations.

MWE Definition To validate the efficacy of the long MWE definition, we perform an ablation study comparing the long definition to the short definition. They are both a summary of the full definition described in Appendix E. As shown in Table 13, the long definition has 162 words while the short is further contracted to 57 words. We perform experiments with Llama-8B and Qwen-7B, using the `tsv_to_tsv` format and the hyperparameters described in Table 11 (same as the main experiments).

Table 14 presents the result. For both models, the long definition achieves higher F1 scores than the short by more than 12 points, indicating the effectiveness of a detailed definition.

B.2 Evaluation Metrics

We use MWE-based (exact-match) precision, recall, and F1 score (Savary et al., 2023). Let G be the set of gold MWEs and H the set of predicted MWEs (hypothesis),

$$\text{Recall} = |G \cap H|/|G| \quad (1)$$

$$\text{Precision} = |G \cap H|/|H| \quad (2)$$

where each MWE is represented with the ID of the sentence and the IDs of its constituent tokens. F1 is the harmonic mean of precision and recall.

Name	Content
Long	`Here, an MWE is defined as a sequence that satisfies the following three conditions.\n 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.\n 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 fro", and the syntactically idiomatic include "long time no see".\n 3. It is not a multi-word named entity, i.e., a specific name of a person, facility, etc.`
Short	`Here, an MWE is defined as a sequence that satisfies the following three conditions.\n 1. It consists of multiple words that are always realized by the same lexemes.\n 2. It is idiomatic, that is, its meaning cannot be explicitly derived from its components.\n 3. It is not a multi-word named entity, i.e., a specific name of a person, facility, etc.`

Table 13: MWE definitions to be included in prompts for fine-tuning.

		F1	P	R	Recall by MWE Type			
					Noun (118)	Verb (154)	Mod/Conn (88)	Clause (6)
Long	Llama-8B	$29.4_{\pm 2.1}$	$82.6_{\pm 1.2}$	$17.9_{\pm 1.6}$	$5.9_{\pm 0.8}$	$26.4_{\pm 1.5}$	$24.2_{\pm 3.5}$	$0.0_{\pm 0.0}$
	Qwen-7B	$45.2_{\pm 0.6}$	$63.2_{\pm 0.9}$	$35.2_{\pm 0.8}$	$26.0_{\pm 2.0}$	$47.2_{\pm 1.0}$	$31.8_{\pm 2.0}$	$0.0_{\pm 0.0}$
Short	Llama-8B	$8.6_{\pm 14.8}$	$28.9_{\pm 50.0}$	$5.0_{\pm 8.7}$	$1.4_{\pm 2.4}$	$7.1_{\pm 12.4}$	$7.2_{\pm 12.5}$	$\mathbf{0.0}_{\pm 0.0}$
	Qwen-7B	$32.8_{\pm 0.2}$	$70.8_{\pm 1.6}$	$21.4_{\pm 0.1}$	$12.1_{\pm 0.5}$	$30.5_{\pm 0.6}$	$22.3_{\pm 0.7}$	$0.0_{\pm 0.0}$

Table 14: Ablation results, as mean percentage scores of three runs with random seeds. The numbers in parentheses are the MWE counts. $\pm$ denotes standard deviation.

C Computational Budgets

For experiments for MaW (Rule+BiEnc and Rule+DCA), we use a single NVIDIA RTX 2080Ti GPU with 11GB RAM. Each run of training and testing takes around 8 minutes. Consequently, the total GPU hours for MaW are estimated to be 0.8 hours.

For LLM fine-tuning, we use NVIDIA RTX 6000 GPU with 48GB RAM for smaller models (Llama-8B and Qwen-7B) and NVIDIA A100 PCIe with 80GB RAM for larger models (Llama-70B and Qwen-72B). Each run of training and testing takes around 66 minutes for the smaller models and 588 minutes for the larger models. Thus, the total GPU hours for fine-tuning experiments of smaller and larger models are estimated to be 6.6 hours and 58.8 hours, respectively.

D Analysis on Computational Costs

MaW's encoder model (`bert-base-uncased`) has only 110M parameters, taking approximately $2 \times 110\text{M} \times 4 = 880\text{M}$ bytes in total (2 is the number of encoders). 4-bit quantized Qwen-72B takes $72,000\text{M} \times 0.5 = 36,000\text{M}$ bytes, taking roughly 40 times more space than MaW. On the other hand, the performance of 4-bit quantized Qwen-7B, which takes roughly 4 times more space, is comparable with Rule+DCA, suggesting that MaW is more memory-efficient.

E Excerpt of Annotation Guidelines

E.1 Definition of MWEs

In our definition, MWEs are idiomatic sequences that (a) consist of multiple words, (b) display semantic, lexical, or syntactic idiomaticity, and (c) are not proper nouns.

a. **An MWE consists of at least two words that are always realized by the same lexemes**.[10] For example, the MWE *"break up"* is always realized by (1) *"break"* or its conjugated form such as *"broke"* and (2) *"up"*. Such words cannot be replaced without distorting the meaning of the expression or violating the language conventions.

b. **An MWE displays semantic, lexical, or syntactic idiomaticity**. The semantically idiomatic MWEs include *"break up"*, the lexically idiomatic include *"to and fro"*, and the syntactically idiomatic include *"long time no see"*.

   - Semantic idiomaticity occurs when **the meaning of an expression cannot be derived from its components**. That is, you cannot necessarily infer the meaning of the expression even if you know all the senses of its components. In other words, a semantically idiomatic expression takes on a meaning that is unique to that combination of words.
     The inferability differs from one expression to another. The meaning of idiomatic MWEs such as *"kick the bucket"* cannot be inferred from its components at all. The meaning of institutionalized phrases like *"traffic light"* is inferable to some degree. Yet *"traffic light"* is semantically idiomatic because it does not mean any type of light related to traffic but a specific type of light.
   - Lexical idiomaticity occurs when **one or more components of an expression are not used as stand-alone words in standard English**. Examples include *"bide one's time"*; *"bide"* rarely appears by itself in today's standard English.
   - Syntactic idiomaticity occurs when **the grammatical structure of an expression cannot be derived directly from that of its components**. It constitutes expressions whose grammar seems to go against standard English grammatical conventions. This includes expressions such as *"all of a sudden,"* where *"sudden"* appears in its archaic noun form.

c. **An MWE is not a proper noun, such as the name of a specific person, organization, and so forth.** In this project, do not annotate proper nouns unless they are part of an MWE (like *Pavlov* in *"Pavlov's dog"*). Proper nouns usually start with a capital letter, while exceptions like *"iPhone 15"* exist. Below is the full list of proper nouns in our definition.[11]

Type	Example
People's names	*Shohei Ohtani*
Nationalities or religious or political groups	*African American, Sunni Muslims*
Facilities	*Narita International Airport, Taipei 101*
Organizations	*Procter & Gamble, Kyoto University*
Geopolitical entities (GPE)	*Los Angeles, the United Kingdom*
Non-GPE locations	*Mount Fuji, Amazon River*
Products	*Toyota Prius, Samsung Galaxy*
Named events	*World War II, Olympic Games Tokyo 2020*
Works of art	*Norwegian Wood, Bohemian Rhapsody*
Named legal documents	*The Magna Carta*
Named languages	*Middle English, American Sign Language*

E.2 Notes

MWEs containing replaceable words

MWEs are not necessarily made of continuous words. Some MWEs contain open slots, that is, words that may be replaced with a large or open class of words. In annotation, we do not include open slots in MWEs, as illustrated by the following examples:

[10] Lexeme is a set of words related through inflection. Words belonging to the same lexeme share a common lemma.

[11] The 11 types derive from the named entity types of OntoNotes Release 5.0 (Weischedel et al., 2013).

- *You **took** me **by surprise**!*
- ***Pick** me **up** at the station.*

	1	2	3	4	5	6
MWE	Pick	me	up	at	the	station
Pick up	☑	☐	☑	☐	☐	☐
	☐	☐	☐	☐	☐	☐

Note that if a word is replaceable with only a very small number of alternatives, we consider it as part of the MWE. For example, we count the following as different MWEs:

- *Their food **leaves a lot to be desired**.*
- *The work **leaves much to be desired**.*

	1	2	3	4	5	6	7
MWE	The	work	leaves	much	to	be	desired
leaves much to be desired	☐	☐	☑	☑	☑	☑	☑
	☐	☐	☐	☐	☐	☐	☐

Overlapping MWEs

There is a chance that multiple MWEs share the same word. For example, *letting* in the following sentence belongs to both ***letting in*** and ***letting out***.

	1	2	3	4	5	6	7
MWE	They	were	letting	him	in	and	out
letting in	☐	☐	☑	☐	☑	☐	☐
letting out	☐	☐	☑	☐	☐	☐	☑
	☐	☐	☐	☐	☐	☐	☐

Rearranged MWEs

There is a chance that the order of component words in an MWE is different from its canonical form, but we still count such rearranged sequences as MWEs. In the following example, rearrangement happens in the MWE ***break** <one's>* ***heart***. As a side note, *my* here is not annotated because it is replaceable.

	1	2	3	4	5
MWE	My	heart	is	broken	.
heart broken	☐	☑	☐	☑	☐
	☐	☐	☐	☐	☐