Hybrid Human-LLM Corpus Construction and LLM Evaluation for the Caused-Motion Construction

Leonie Weissweiler, Abdullatif Köksal, Hinrich Schütze
LMU Munich & Munich Center for Machine Learning
weissweiler@cis.lmu.de

Abstract The caused-motion construction (CMC, "She sneezed the foam off her cappuccino") is one of the most well-studied constructions in Construction Grammar (CxG). It is a prime example for describing how constructions must carry meaning, as otherwise the fact that "sneeze" in this context takes two arguments and causes motion cannot be explained. We form the hypothesis that this remains challenging even for state-of-the-art Large Language Models (LLMs), for which we devise a test based on substituting the verb with a prototypical motion verb. To be able to perform this test at a statistically significant scale, in the absence of adequate CxG corpora, we develop a novel pipeline of NLP-assisted collection of linguistically annotated text. We show how dependency parsing and LLMs can be used to significantly reduce annotation cost and thus enable the annotation of rare phenomena at scale. We then evaluate OpenAI, Gemma3, Llama3, OLMo2, Mistral and Aya models for their understanding of the CMC using the newly collected corpus. We find that most models struggle with understanding the motion component that the CMC adds to a sentence.

1 Introduction

(1) She sneezed the foam off her cappuccino.

(2) They laughed him off the stage.

These are two examples of the caused-motion construction (CMC) in which the verb behaves unusually: *sneeze* and *laugh* typically do not take multiple arguments, nor do they typically convey that something was moved by sneezing/laughing. This poses a challenge to any naive form of lexical semantics: it would not make sense for someone writing a dictionary to include, for each intransitive verb, the meaning and valency of the CMC. Almost any verb can appear in the CMC as long as we can imagine a scenario in which the action it describes causes motion. The fact that humans easily understand the CMC showcases a main feature of Construction Grammar (Croft, 2001; Goldberg, 1995): the meaning is attached to the construction itself, and not the verb. Putting the verb into this construction adds the new meaning and valency. This is one reason that constructions pose a challenge to Large Language Models (LLMs), as they would have to learn to attach the meaning to this construction and retrieve it when necessary. Its extreme rarity and productivity makes it impossible to memorise all instances and memorisation would not be sufficient because the meaning shift to the verb is creative and is influenced by the specific context.

The research questions of this paper therefore are: Have LLMs learned the meaning of the CMC and how can we construct the resources needed to determine the status of CMC in LLMs?

We first address the second question, of collecting data for this at scale. This is challenging for several reasons. First, the CMC is a very rare phenomenon. Second, we are mostly interested in instances that are non-prototypical, i.e., where the verb does not typically encode motion, unlike e.g. 'kick' or 'throw'. Third, this construction cannot be automatically identified using only syntactic criteria: words might be in the correct syntactic slots required by the CMC, but not create a CMC reading if the semantics of the sentence do not fit. For example, "I would take that into account" is structurally identical to the examples above, but nothing is moving.

This shows that there is a crucial semantic component. The rarity makes it very costly to manually sift through a corpus to collect a dataset of the CMC, while the semantic complexity makes it infeasible to do so fully automatically.

In this way, we consider the CMC exemplary of rare phenomena of language that have been largely set aside in Computational Linguistics and in recent evaluation of LLMs in particular. This may be due to them being considered the *periphery* of language, rather than the core (Chomsky, 1993), or simply due to the described

https://doi.org/10.3384/nejlt.2000-1533.2025.5256

difficulty in finding appropriate data to investigate both the phenomena and their representation in LLMs. However, it is our point of view that as the performance of such models increases across the board, it is vital to turn to "edge cases" to accurately identify performance gaps. This is particularly important as rare phenomena may be indicators of systematic underlying problems of an NLP paradigm.

To study rare phenomena, we need natural data for them at scale. To this end, in section 3 we propose a novel annotation pipeline that combines dependency parsing with the use of LLMs. The aim of our pipeline is to minimise the cost of running the LLM and compensating human annotators, while maximising the number of positive, manually verified, linguistically diverse instances in the dataset.

After creating our corpus, we now return to our aim of evaluating state-of-the-art LLMs for their understanding of the CMC, as an example of a semantically challenging "edge case".

In Section 4, we therefore develop a test for different LLMs' understanding of the CMC, by giving an instance and asking if the direct object is physically moving. We then replace the verb (e.g., "sneeze") by a prototypical one that always encodes motion (e.g., "throw") and ask the model again if the direct object is moving. We expect models that do not fully understand the CMC to fail to consistently answer both questions with "yes". We observe that models struggle with this task to varying degrees.

We make three main contributions:

- We propose a hybrid human-LLM corpus construction method and show its effectiveness for the CMC, an extremely rare phenomenon. We discuss how our design and our guidelines can be applied to data collection needs for other linguistic phenomena.
- We release a corpus of manually verified instances of the CMC of 500 sentences.[1]
- We evaluate different sizes of Llama3, Gemma3, OLMo2, Mistral, Aya, and OpenAI models on their understanding of the CMC and find that most models struggle.

2 Related Work

Evaluation of LLMs' Understanding of Constructions. Tayyar Madabushi et al. (2020) conclude that BERT (Devlin et al., 2019) can classify whether two sentences contain instances of the same construction. Tseng et al. (2022) show that LMs have higher prediction accuracy on fixed than on variable syntactic slots and infer that LMs acquire constructional knowledge (i.e., they understand the "syntactic context" needed to identify a fixed slot). Weissweiler et al. (2022) find that LLMs reliably discriminate instances of the English Comparative Correlative (CC) from superficially similar contexts. However, LLMs do not produce correct inferences from them, i.e., they do not understand its meaning.

Zhou et al. (2024) evaluate LMs' understanding of the causal excess construction by contrasting it with two constructions of similar structure, and using the LMs' ability to distinguish between them as a proxy for measuring their understanding. They find that even large models like GPT-4 perform poorly on this. By contrast, Rozner et al. (2025a), using the same dataset among others, investigate smaller masked language models. They do not test understanding but rather probe the internal representations of the output layer to recover systematic differences between the constructions, showing that distinguishing between them is possible. Rozner et al. (2025b) repeat this experiment with BabyLM models and find that even they are capable of picking up many constructions, providing valuable evidence about construction learning with developmentally plausible amounts of data.

Bonial and Tayyar Madabushi (2024) compile a corpus of examples from several constructions, including the 52 caused-motion sentences collected from the Abstract Meaning Representation (AMR) dataset (Banarescu et al., 2013). They evaluate GPT-4 and GPT-3.5 on their ability to pick out three caused-motion sentences from among a larger set, and find that performance does not exceed 60%. However, it should be noted that this was metalinguistic prompting, relying on a model's understanding of the term 'caused-motion', which many humans may also be unfamiliar with.

Most related to this work, Li et al. (2022) probe for LMs' handling of four Argument Structure Constructions (ASCs): ditransitive, resultative, caused-motion, and removal. They adapt the findings of Bencini and Goldberg (2000), who used a sentence sorting task to determine whether human participants perceive the argument structure or the verb as the main factor in the sentence meaning. They find that, while human participants prefer sorting by the construction more if they are more proficient English speakers, language models show the same effect in relation to training data size. In a second experiment, they then insert random verbs that are incompatible with one of the constructions, and measure the Euclidean distance between the verbs' contextual embedding and that of a verb that is prototypical for the construction. They demonstrate that

[1] Code and data are provided on https://github.com/LeonieWeissweiler/CausedMotion

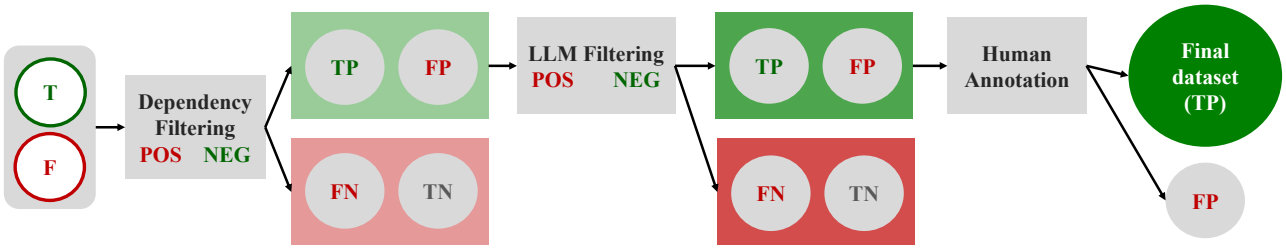


Figure 1: Flowchart of our annotation pipeline. For details of each step refer to §3.

construction information is picked up by the model, as the contextual embedding of the verb is brought closer to the corresponding prototypical verb embedding.

Mahowald (2023) investigates GPT-3's (Brown et al., 2020) understanding of the English Adjective-Article-Numeral-Noun construction (AANN), assessing its grasp of the construction's semantic and syntactic constraints. Utilising a few-shot prompt based on the CoLA corpus of linguistic acceptability (Warstadt et al., 2019), he creates artificial AANN variants as probing data. GPT-3's performance on the linguistic acceptability task is found to align with human judgments across most conditions. More recently, Misra and Mahowald (2024) investigate the same construction for smaller models trained on the BabyLM corpus (Warstadt et al., 2023) and show how its learning is supported by more frequent, smaller constructions. In a similar vein, Scivetti et al. (2025) investigate how well BabyLM size models acquire the let-alone construction.

Linguistic Annotation with LLMs Since the release of ChatGPT, numerous papers have proposed to use it or similar LLMs as an annotator. Gilardi et al. (2023) find that ChatGPT outperforms crowd-workers on tasks such as topic detection. Yu et al. (2023) and Savelka and Ashley (2023) evaluate the accuracy of GPT-3.5 and GPT-4 against human annotators, while Koptyra et al. (2023) annotate a corpus of data labelled for emotion by ChatGPT, but acknowledge its lower accuracy compared to a human-annotated version. In the area of Construction Grammar, Torrent et al. (2023) use ChatGPT to generate novel instances of constructions.

Most related to our work are papers that propose a cooperation between the LLM and the human annotator. Holter and Ell (2023) create a small gold standard for industry requirements by generating an initial parse tree with GPT-3 and then correcting it with a human annotator. Pangakis et al. (2023) investigate LLM annotation performance on 27 different tasks in two steps. First, annotators compile a codebook of annotation guidelines, which is then given to the LLM as help for annotation, and then the codebook is refined by the annotators in a second step. However, they find little to no improvement from the second step. Gray et al. (2023) make an LLM pre-generate labels for legal text analytics tasks which are then corrected by human annotators, but find that this does not speed up the annotation process.

In contrast, our work proposes a hybrid human-LLM pipeline that minimizes the cost of dataset creation. We emphasise prompt design and engineering, a critical factor in effective use of LLMs.

Computational Approaches to Argument Structure Constructions. In addition to the probing work discussed above, ASCs have also been studied from a computational perspective. Kyle and Sung (2023) leverage a UD-parsed corpus as well as FrameNet (Fillmore et al., 2012) semantic labelling to annotate a range of ASCs.

Hwang and Palmer (2015) identify CMCs and four different subtypes based on linguistic features. Some of these are automatically generated, but others are gold annotations. This limits the applicability to large, unannotated corpora.

Hwang and Kim (2023) conduct an automatic analysis of constructional diversity to predict ESL speakers' language proficiency. Similar to our first filtering step, they perform an automatic dependency parse and then identify a range of constructions, including the CMC, using a decision tree built on the parse. They do not employ any further filtering.

3 Data Collection

Concept of the CMC In collecting a dataset of CMC instances, we must first find a working definition of the CMC to guide our automatic and manual annotation. While we base our definition on that of Goldberg (1992), we also restrict it further to include only sentences in which the object is physically moving. This is not meant as a universal definition of the CMC, but rather as one that suits the needs of our project, as we later ask LLMs if the direct object is moving and where. We therefore make no definitive statement as to whether metaphorical movement (*I laughed myself off the chair*), the electronic movement of data (*I sent him an email*), or movement involving a metaphysical location (*She sneezed herself out of existence*) constitute

instances of the CMC.

Data Collection Pipeline Our aim is to investigate how well the caused-motion construction is learned by LLMs, for which we require a dataset of caused-motion sentences, which should be natural and therefore sourced from text. The simplest version of this would be to have human annotators sift through a corpus and extract all caused-motion sentences. This would be very expensive, as we assume caused-motion sentences to be quite rare. On the other hand, they are so semantically complex that we cannot simply use automated filtering, e.g. based on dependencies. We therefore propose a hybrid approach combining linguistic resources, an LLM, and an expert annotator.

Our key idea is that data collection will proceed in a pipeline, where a corpus is first filtered using dependency parsing and the syntactic constraints of the CMC, the output set of sentences is further filtered with prompt-based classification using an LLM, and the sentences which it labels as positive are then manually annotated by a human. Each step in the pipeline is meant to further concentrate the rate of instances in the corpus that will then be manually annotated, therefore reducing total annotation effort.

The main cost of data collection is the cost of the LLM API and for human annotators. We assume that any expenses for linguistic resources and the computational infrastructure (not relevant to running LLMs) at our disposal are negligible in comparison. *Our aim is to minimise the cost for the LLM and annotators while maximising the number of positive, manually verified, diverse instances.*

We propose a way of computing the cost for this problem setting and a pipeline for producing a novel linguistic resource while minimising cost.

Our main goal is to minimise the cost per confirmed CMC sentence; however, we also have a secondary goal: the final set of sentences should be diverse. Regardless of the specific goals of the linguistic researcher, it is unlikely that they would be served by a set of sentences that do not represent the true diversity of the CMC. Extreme cost-minimising measures – such as making the dependency filtering rules described in §3.1 too strict or asking the LLM to provide examples of the CMC – would therefore be counterproductive.

The baseline here is to take an annotator, give them a corpus, set them on the task of reading through it and marking all sentences that contain instances of the CMC. As the corpus contains very few true positives, this would be highly costly. We therefore turn to dependency parsing with spaCy (Honnibal et al., 2020) for prefiltering. We select the reddit corpus (Baumgartner et al., 2020), with the motivation that it will contain a high rate of creative language usage, aiding our goal of finding as many non-prototypical CMC instances as possible.

class	PR	RE	F1	n
True	79.76	97.10	87.58	69
False	75.00	26.09	38.71	23
Avg	77.38	61.59	63.15	92

Table 1: Accuracies of the dependency filtering based on the total set of positive and negative instances from Goldberg (1992). We focus on maximising Recall (RE) of the True class, to minimise the number of potential CMC sentences that are lost before human annotation, achieving 97%.

3.1 Step 1: Dependency Parsing

Figure 1 shows our pipeline. In the first step, we dependency-parse the corpus and apply a pattern to filter out all sentences that, with high likelihood, are not instances of the phenomenon.

For this dependency annotation, we could rely on annotated treebanks such as Universal Dependencies (de Marneffe et al., 2021). But to find a diverse and sufficiently large set of instances, particularly in languages other than English, available treebanks may not be large enough for the rare phenomenon that we are targeting.

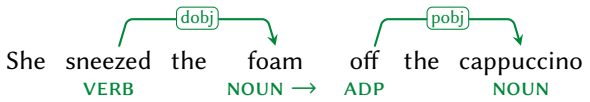
We therefore turn to automated dependency parsing to annotate large amounts of data, which we can run by using minimal computational resources without the need for GPUs.

After dependency parsing, we want filters that preserve the diversity of the found sentences. We therefore design subtree filters that preserve recall above all else. This is especially advisable as parsing will lead to some parsing errors that we want to be tolerant of, and as CMC sentences are rare, they are more likely to be parsed incorrectly.

To design the pattern, we start with a list of gold instances taken from Goldberg (1992), which we parse with the spaCy toolkit.[2] The instances are positive and negative examples for the CMC. On the basis of their dependency parses, we develop dependency constraints as a filter for our dependency-parsed sentences. Specifically, we iterate over the verbs in a sentence, then look for a direct object or a recursive dependent of the direct object, e.g. an adjective, immediately following the verb. In the position immediately following, we check for an adposition, while taking into account that it may comprise several tokens. We do not impose constraints on the dependency between adposition and

[2] version 3.2.0

prepositional object, as we have found these to be especially vulnerable to parsing errors. We then look for a pobj-dependent of this adposition.



We design the subtree to optimise recall with reasonable precision, following the overall goal of losing as few sentences as possible in the pipeline to maximise final dataset diversity.

We then evaluate its recall and precision on this small development set, comprising the total sum of positive and negative CMC instances given in Goldberg (1992), and report on the results in Table 1. Our filter achieves 97.10 % recall for true CMC instances, minimising the number of sentences lost in this step.

This filtering step also allows us to extract the location of the potential CMC instance and its parts as a side product of the filtering step: We extract the sentence, the lemmatised verb, direct object, preposition, and prepositional object, as well as their positions in the sentence.

3.2 Step 2: Selection of Sentences for Classification

Given that we now have a lot of dependency-filtered data and limited resources for classification, we want to select the optimal set of sentences for this classification, in order to optimise several criteria for our final dataset. As the dataset will form a challenging evaluation set for LLMs, the most important of these criteria is that the dataset contains as many verbs as possible that do not usually contain motion. Even though we consider sentences like "I throw the ball" instances of the CMC, they would not challenge a model's understanding, as "throw" already encodes motion. As a proxy for this, we sort verbs by how frequently they are used intransitively, with the idea that these would make for less prototypical CMC sentences.

We compute statistics about the verbs with UD. Specifically, we merge the English treebanks EWT (Silveira et al., 2014), GUM (Zeldes, 2017), GUM reddit (Behzad and Zeldes, 2020), LinES (Ahrenberg, 2007), partTUT (Sanguinetti and Bosco, 2015), PUD (Zeman et al., 2017), and GENTLE (Aoyama et al., 2023), and then for each verb, we compute the ratio of how often that verb has an object. We then go through the dependency-filtered dataset from the last step and sort by this ratio. This has the added benefit of removing verbs that never appeared in UD as lemmata, which removes noise from the reddit dataset.

3.3 Step 3: Prompt-based Few-shot Classification with an LLM

Goals Even after dependency-based filtering, the positive instances would still be very rare in the output, and it is therefore not feasible that the output is directly annotated by a human. We therefore introduce a further filtering step with an LLM to "concentrate" the positive instances even more, i.e. we want the LLM to remove most negative instances while keeping as many positive instances as possible. The remaining data can then be cost-effectively annotated by the human annotator. The aim is to reduce the cost per instance (i.e., cost per true positive, TP) as much as possible.

There are two components of the cost: the cost of querying the LLM and the cost of human annotation. Our two key ideas are:

- We consider the two costs jointly and optimise the pipeline for overall lowest cost per TP.
- Design and selection of the prompting setup (including the prompt, the choice of model, how many times it's run, etc.) used with the API is a major determinant for the cost of the pipeline. We propose a workflow for creating effective prompting setups.

A particular prompting setup may require many tokens in total, thereby incurring a higher API cost. But it may also have high accuracy, thereby reducing the cost of human annotation. We jointly consider both cost components when designing and selecting prompting setups.

Development Set For creating the development set V, we manually annotate 500 (183 positive, 317 negative) sentences from the output of the dependency filtering step. To ensure that V is both diverse and relevant, we group the prefiltered dataset by verb, and starting with the highest-frequency verbs, take at most 5 positive and 5 negative sentences from every verb, where no preposition appears twice in either the positive or the negative sentences selected. We choose 25 shots from each class to be included as examples in the prompt, which are not used for V.

Minimising the cost per true positive Given this development set, let $J(C_{\text{HR}}, C_{\text{API}}, i)$ be the cost per true positive where C_{HR} is the human annotation cost per sentence, C_{API} is the cost of processing an input/output token with the API and i (for instruction) is a prompting setup. We can then estimate $J(C_{\text{HR}}, C_{\text{API}}, i)$, the cost per true positive, as follows:

$$\frac{C_{\text{API}} t(V, i) + C_{\text{HR}}(\text{TP}(V, i) + \text{FP}(V, i))}{\text{TP}(V, i)} \quad (1)$$

				Sent's to Annotate			Total Cost		
P	**Details**	**Prec.**	**Rec.**	**LLM**	**Human**	**API**	C_{HR}=**\$.002**	C_{HR}=**\$.006**	C_{HR}=**\$.5**
1	Base (4o-mini)	0.486	0.582	3535	1719	0.01	3.46	10.3	860
2	1 + repeat sentence with json	0.459	0.656	3320	1524	0.04	3.13	9.2	762
3	2 + reason	0.470	0.662	3217	1512	0.07	3.15	9.2	756
4	3 + structured information	0.648	0.621	2483	1610	0.07	3.33	9.8	805
5	4 + sentence	0.519	0.664	2900	1505	0.07	3.13	9.2	753
6	4 + cmc string	0.393	0.462	5507	2167	0.09	4.44	13.1	1083
7	4 + cmc string continuous	0.536	0.681	2744	1469	0.06	3.06	8.9	735
8	6 + sentence	0.536	0.658	2839	1520	0.06	3.15	9.2	760
9	7 + sentence	0.579	0.658	2622	1519	0.06	3.14	9.2	760
10	4 + few shots	0.557	0.600	2990	1667	0.08	3.46	10.1	833
11	10 + explanations	0.694	0.608	2371	1646	0.06	3.39	10.0	823
12	11 + all shots	0.710	0.653	2155	1531	0.07	3.18	9.3	766
13	**12 + only 10 samples**	0.721	0.714	1943	1402	0.11	**3.02**	8.6	701
14	12 + only 1 sample	0.552	0.789	2296	1267	0.76	4.17	9.2	635
15	12 + only 5 samples	0.639	0.713	2192	1402	0.19	3.18	8.8	701
16	12 + only 25 samples	0.738	0.678	1998	1474	0.08	3.09	9.0	737
17	14 + new few-shots	0.552	0.789	2296	1267	0.76	4.17	9.2	635
18	17 + alternating shots	0.588	0.805	2114	1243	0.70	4.04	9.0	623
19	17 + grouped shots	0.448	0.796	2803	1256	0.93	4.53	9.6	630
20	19 + majority vote	0.486	0.840	2449	1191	2.44	7.97	12.7	601
21	19 on o3-mini	0.913	0.856	1280	1168	4.91	13.83	18.5	595
22	21 + 100 samples	0.760	0.874	1506	1144	0.67	3.90	8.5	574
23	21 + 250 samples	0.820	0.806	1513	1240	0.52	3.64	8.6	621
24	21 + 50 samples	0.803	0.865	1440	1156	0.80	4.20	8.8	580
25	21 + 25 samples	0.798	0.864	1451	1158	0.83	4.27	8.9	581
26	24 + majority vote	0.803	0.891	1397	1122	2.42	8.13	12.6	567
27	24 on 4o	0.814	0.837	1467	1195	0.75	4.10	8.9	599
28	24 - sentence	0.787	0.878	1447	1139	0.75	4.07	8.6	571
29	27 - sentence	0.803	0.821	1516	1218	0.54	3.65	8.5	610
30	**28 - reason**	0.803	0.891	1397	1122	0.70	3.96	**8.4**	563
31	29 - reason	0.760	0.790	1667	1266	0.60	3.82	8.9	634
32	**22 on o1**	0.880	0.920	1235	1087	5.79	16.72	21.1	**558**
33	32 + 50 samples	0.891	0.916	1226	1092	7.10	19.94	24.3	564
34	33 + majority vote	0.869	0.952	1209	1050	22.30	60.10	64.3	583
-	Human only	-	-	-	2732	0.00	5.46	16.4	1366

Table 2: A comparison of all prompting setups for different values of C_{HR}. **P** = Prompting Setup. We give numbers (sentences that need to be annotated by LLM/human) for a scenario in which the desired size of the final resource (output of pipeline when applied to the raw corpus) is $N = 1000$. The human baseline depends solely on the rate of TPs (which is higher here than for the raw corpus to be processed by the pipeline as the development set contains more positive instances). The different values of C_{HR} were chosen to highlight the different scenarios in which the three best prompting setups, 13, 30, and 32, are each optimal.

where we process the development set using the API and prompting setup i and record: $\text{TP}(V, i)$, the number of true positives, $\text{FP}(V, i)$), the number of false positives, and $t(V, i)$, the sum of the number of tokens input to the API and the number of tokens returned by the API.

We create a variety of different prompting setups (where with prompting setup we refer to a combination of prompt, model, and other configurations like majority voting) i and then select our final prompting setup i' as the one with the lowest per-TP cost:

$$i' = \text{argmin}_i J(C_{\text{HR}}, C_{\text{API}}, i)$$

Determining the size of the input corpus To compile our CMC dataset, we set a target number of $\text{TP}_{\text{req}} = 292$ instances of the CMC, to bring the total up to 500 by later adding the manually annotated positive development instances and the positive few-shots. After selecting a prompting setup i and determining $\text{TP}(V, i)$ on the development set, we can estimate the size N of the

input corpus that will result in a set of TP_{req} instances to be output by the pipeline as:

$$N := |V| \frac{TP_{req}}{TP(V, i)}$$

Iterative Prompting Setup Development We start with a simple base prompting setup and iteratively attempt improvements to it. The total cost of this experimentation was about $22. The full details of all attempted prompting setups are given in the appendix in Section A. We test four models from OpenAI of those available in February 2025: 4o-mini, 4o, o3-mini, and o1. For this experiment, we use sampling with temperature=1.0 and top_p=1.0.[3]

During prompt development, we do not have a good estimation of the human annotator cost, as we will ultimately annotate the sentences ourselves. We, however, assume that C_{HR} should be at least $0.001, which means that we can determine many prompting setup improvements to be clear improvements and only have to consider the cost tradeoff for some.

We start with a simple prompting setup that gives no few-shot examples and asks for sentence IDs and classifications in a csv codeblock, classifying 50 sentences at a time with 4o-mini. The instruction remains the same throughout and can be seen in the prompt example in Table 3. We achieve straightforward improvements by making the model repeat the sentence (and therefore giving the output as a json object to avoid confusion over commas), but not with having 4o-mini give a reason for its decision. We then try out different combinations of giving the entire sentence, only the substring containing the core CMC, and the structured information given by the dependency parsing step. We add few shots and hand-written explanations for our labels for them. We also vary the number of samples, increase the number of few-shots, and reorder them. We then add majority voting after running each sentence 3 times, and try out different numbers of sentences to be classified for each prompt. During this process, we also switch to the more expensive models o3-mini, 4o, and o1. The final optimal prompting setup depends on the human annotation cost. In Figure 2, we visualise with grey vertical lines where one prompting setup "overtakes" another, meaning the human annotation cost per sentence where the optimal prompting setup changes. We then show example total cost figures for three reasonable values in between these change points in Table 2, revealing that the best prompting setups are 13, 30, and 32, depending on human annotation cost.

As our **final prompting setup**, we select prompting setup 30 as it is a good tradeoff between API cost and human cost.

[3]The specific models used were `gpt-4o-mini-2024-07-18`, `gpt-4o-2024-11-20`, `o3-mini-2025-01-31` and `o1-2024-12-17`.

3.4 Final Dataset Collection

In combination with the 183 positive instances from the development set, and an additional 25 positive instances from the few shots, we now set out to annotate additional data using our pipeline, to reach a final dataset of 500 hand-annotated CMC instances. To this end, we classify an additional 9,046 sentences with prompting setup 30, with approximately 3.6 USD in API costs. 598 of these (6.6%) are classified as positive by the model. We annotate these by hand, resulting in 292 positive and 396 negative instances, which gives the prompting setup a precision of 48.83% in practice. We see the reason for this lower precision mostly in the fact that the concentration of true positives was likely much lower in the data processed here, than in the development set, which was chosen to have many diverse CMC instances. Examples for sentences in the final dataset are given in Table 4.

4 Evaluation of LLMs' Understanding of the CMC

4.1 Methods

The goal of our evaluation is to assess different LLMs for their understanding of the CMC. The performance reached by the prompts in the data collection phase is not a suitable measure for this, since it relied on metalinguistic prompting and few-shots.

Our LLM evaluation setup in this section differs from prompting setup evaluation as we do not explicitly refer to the "caused-motion construction", but rather prompt implicitly for the model's understanding of the situation described. The key idea is that in a CMC sentence, something is always physically moving, even if the verb (e.g., "sneeze") does not indicate this. The distinction between prototypical vs. non-prototypical instances is crucial here: for prototypical CMC instances ("throw", "kick"), the verb already conveys the meaning component of motion while for non-prototypical CMC instances ("sneeze", "laugh") it does not and the LLM has to infer the additional meaning component of motion from the construction.

Our setup is to ask "In the sentence "...", is *direct_object* moving, yes or no?". If a model were to answer this with "yes", we would feel confident that it has understood the CMC; however, if it answered with "no", we could not be sure that the model has failed specifically in its understanding of the CMC, and not of the sentence or situation in general. We therefore construct a control question, for which we replace the verb of the CMC with the appropriately inflected form of "throw",

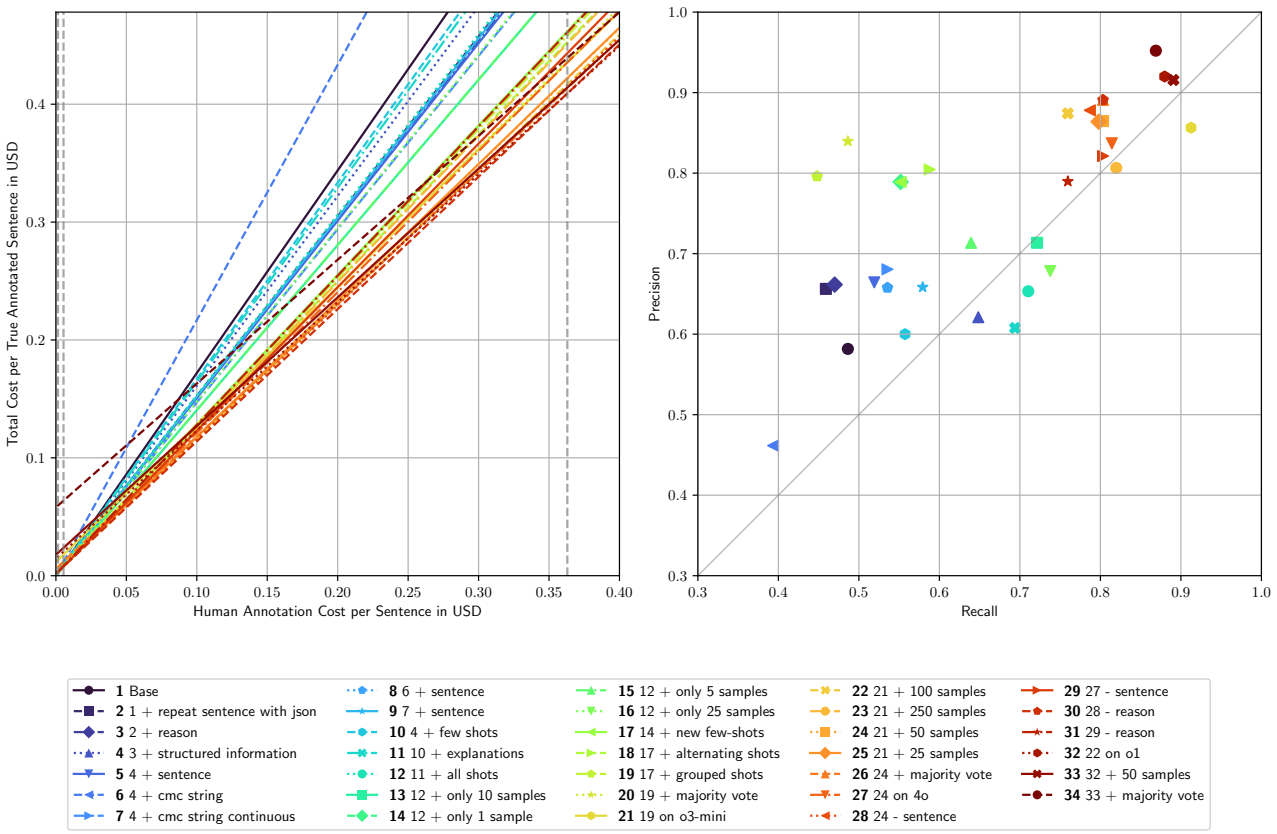


Figure 2: A comparison of all prompting setups that were considered in development. On the left, the total cost per true annotated sentence is shown dependent on the human annotation cost, in USD. On the right, prompts are compared by recall and precision.

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.

Table 3: An example prompting setup (30)

I crumble them into the bowl one at a time .
I just wept a single tear into my beard .
He hissed air through his clenched teeth .
did people really crane grand pianos to upper floors ?
Gently swirl it into the batter .

Table 4: Examples from the final dataset. Verbs are highlighted in green, direct objects in purple, prepositions in blue, and prepositional objects in red.

and ask the same question again, using the structural information extracted by the dependency filtering step. This is intended to test if the model is having a general problem understanding the sentence (which would still be an issue, but not the one we set out to find), or specifically with the CMC. While the sentence variants with “throw” are still instances of the CMC, they are now prototypical ones, which we expect to require no deeper understanding of the semantics of the CMC, as

Question Type	Example Sentence
original	In the sentence 'did people really [crane|throw] grand pianos to upper floors ?', did pianos really move, yes or no?
original_prep	In the sentence 'did people really [crane|throw] grand pianos to upper floors ?', did pianos really move to floors, yes or no?
medium	In the sentence 'People [crane|throw] grand pianos to upper floors .', do pianos move, yes or no?
medium_prep	In the sentence 'People [crane|throw] grand pianos to upper floors .', do pianos move to floors, yes or no?
short	In the sentence 'You [crane|throw] pianos to floors .', do pianos move, yes or no?
short_prep	In the sentence 'You [crane|throw] pianos to floors .', do pianos move to floors, yes or no?

Table 5: An overview of the prompt formats for LLMs, for the example sentence 'did people really crane grand pianos to upper floors?'. For each prompt, the main verb 'crane' is optionally replaced with the appropriate form of 'throw'. Each question exists once with the direct object and once without. The sentence itself is modified with two stages of simplification (medium and short).

the verb is behaving in a prototypical and frequently observed way. We expect that models with no understanding of the CMC would answer "yes" both times only for prototypical instances, and switch from "no" to "yes" for non-prototypical ones. Models with a perfect understanding of the CMC would always answer "yes".

As this only covers the most basic element of understanding the CMC sentence, the presence of motion, we also expand the evaluation paradigm to also query the destination of the caused motion. This results in a question of the format "In the sentence "...", is *direct_object* moving *prep prep_obj*, yes or no?". This is a more challenging version of the question, which will allow us to test the models on all aspects of the CMC's meaning.

Some of the sentences in our corpus contain modal verbs (e.g., *I may sneeze the foam off the cappuccino*), questions (e.g. *Did you sneeze the foam off the cappuccino?*), or other hypotheticals (e.g. *I nearly sneezed the foam off the cappuccino.*). Asking if the foam moved off the cappuccino in any of these sentences should be correctly answered with 'no', or at least with a lengthy explanation, which introduces noise into our evaluation. We therefore automatically modify each sentence using the existing dependency parse to form simpler sentences in the present tense and indicative mood, which we call "medium" sentences. In a more radical edit, we also form a "short" version, which consists only of the verb, direct object, preposition, and prepositional object, forming a sentence together with a pronoun. This is meant to evaluate if additional context helps or hinders the models in answering the question. Examples for all sentence and question types are given in Table 5.

We conduct this experiment on our corpus of 500 hand-annotated sentences. As API-based LLM, we investigate OpenAI's 4o-mini (OpenAI, 2022). From the family of open LLMs, we further choose Llama3 (Touvron et al., 2023) in sizes 8B and 70B from version 3.1, and 1B and 3B from version 3.2, Mistral 7b (Jiang et al., 2023), OLMo2 in sizes 7B and 13B (OLMo et al., 2025), Gemma3 in sizes 1B, 4B, 12B, and 27B (Team et al., 2025), as well as Aya Expanse 8B (Dang et al., 2024). Models generate a sentence in response, which we then parse for versions of "yes" and "no". We use temperature 0 for all models, i.e. greedy decoding.

4.2 Results

Figure 3 presents the results in three groups. (i) Green: the model answers "yes" both times and therefore demonstrates that it understands the CMC. (ii) Red: The model answers with "no" for the original sentence but changes its answer to "yes" when the verb is changed to "throw", meaning that it does not understand the CMC. (iii) Grey: Even with "throw", the model does not answer correctly that the direct object is moving. We consider these to be general failures of the model to understand the instruction, rather than the CMC specifically.

Indicative Present Sentences On this subgroup, titled 'medium' and 'medium_prep' in the plot, performance is higher for all models than on the questions formed with original sentences. This fits well with our intuition that the original sentences sometimes consider modals and hypotheticals, and can therefore not straightforwardly be answered with 'yes', and we therefore consider these to be the main LLM results.

Context-Free Sentences For this minimal version of the evaluation, models overall perform as well or slightly worse than for the indicative present variants. This indicates that the lack of additional context only minimally hurts model performance, and consequently, that models were only utilising the context to answer the question to a small degree.

Destination of Caused Motion If we ask only if the direct object is moving, we cannot take any model's accuracy as a direct measure of its understanding of the entire construction. It is possible that a model might understand that the direct object is moving in some way, but not precisely in which direction, and therefore wouldn't have entirely captioned the boundaries of the

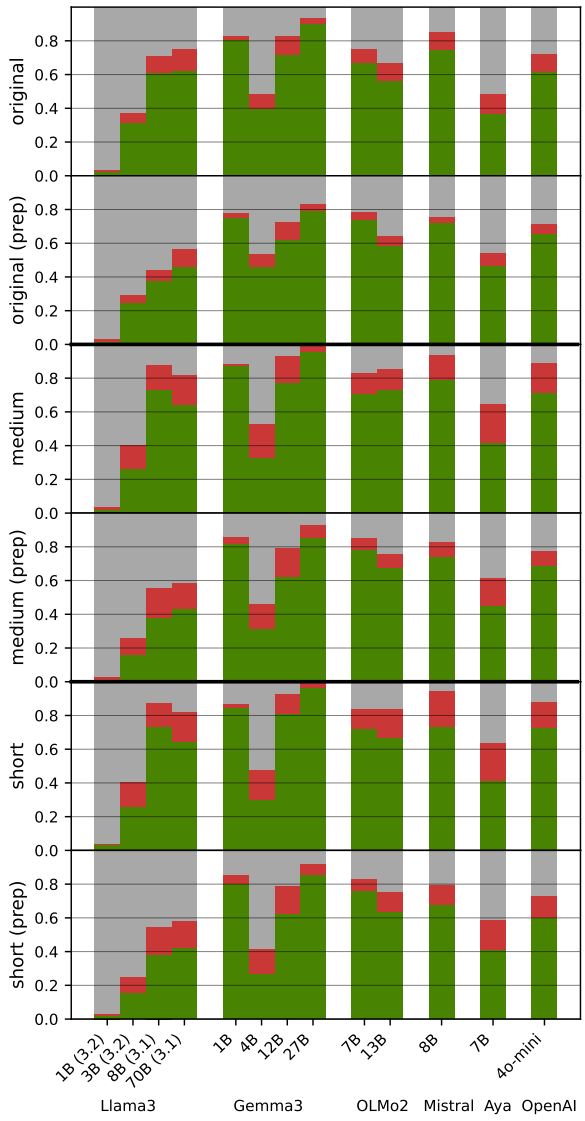


Figure 3: Results for each model and evaluation type. Examples for the evaluation types are given in Table 5. Correct answers are coloured in green, incorrect in red, and invalid results in grey.

Neither is it ever going to vibrate itself out of place .
I chop up the bacon and crumble it on top .
Do not squat the bar off the ground .
We thin the weak from the heard .
It rained arrows from the sky at any rate .

Table 6: Examples from the final dataset which were wrongly classified as negative instances by prompt 30. Verbs are highlighted in green, direct objects in purple, prepositions in blue, and prepositional objects in red.

CMC. To test this, we design a second question that includes the prepositional object, examples for which can be seen in Table 5, where the question types are suffixed with _prep.

Across the board, models give fewer correct answers to these questions than to the ones which do not include the destination (always directly above in Figure 3). However, the rate of false answers mostly stays the same or decreases, while the rate of invalid answers increases, meaning that models are more likely to answer 'no' when asked the question, including the destination of 'throw'. This may indicate that models are having general trouble interpreting these complex sentences. The pattern holds even when considering the short_prep category, where nothing else in the sentence could interfere with the model's understanding.

Results by Model Comparing different models, we find that Gemma3 perform best, with the 27B variant consistently in the range of 90%. The performance of Llama3 is correlated with model size, while that of Gemma3 is not. Gemma3 1B stands out in particular with performance almost rivalling that of the 27B version, for unknown reasons. The high performance of Gemma3 27B indicates that our questions are solvable for models, but remain a challenge for most of them. This is further supported by the fact that the only sentence types where this model falls below 90% is in the original and original_prep categories, which may include sentences where 'yes' is not the correct answer, as explained above.

4.3 Results on False Negatives

Even though our pipeline to create the test corpus included manual verification of all sentences, there is still a possibility that the automated steps introduced bias, i.e. mistakenly filtered out a set of sentences that would have significantly altered the results of our LLM evaluation. To investigate this, we repeat the same evaluation using specifically the false negatives from our corpus collection. While it would be infeasible to collect false negatives from the dependency filtering step due to the

very low concentration of CMC sentences in raw data, we can take a sample of the false negatives of the LLM filtering step simply by using the false negatives from the development set that we hand-annotated earlier. With the final prompt 30, this was a set of 36 sentences that had been hand-annotated as CMC sentences, but were wrongly missed by the prompt. If the results of running the LLM evaluation on these were identical to running it on the entire collected dataset, this would tell us that the LLM filtering does not systematically exclude sentences that are more or less challenging for other LLMs to answer questions about than a random sample would have been. While we cannot find any obvious patterns in the set of false negatives, we provide some example sentences from it in Table 6.

We present the results of this in Figure 4. The results are striking: all models perform significantly worse on this set of 36 false negatives. Most interestingly, the largest change is the increase in false answers and decrease in invalid answers. This leads us to two conclusions. First, the LLMs overlap in their notion of difficulty of a CMC sentence: while the false negatives come from prompt 30, which used GPT-4o, the sentences that it misclassified were not only more difficult for 4o-mini, but also for all other models. Second, the results in the previous section, while more robust because they were based on 500, not just 36 sentences, overestimated all models' understanding of the CMC. Interestingly, the previously best model, Gemma3 27B, is now rivalled by its much smaller variant, Gemma3 1B, and neither performs as well as on the full dataset. On the other hand, specifically the short variant, which are minimal sentences where we do not ask for the destination of movement, were still almost fully solved by Gemma3 27B. It should also be noted, however, that the general relative trends between models are very similar to those of the full evaluation. This control set is, of course, also not a representation of the true distribution; it is likely that it represents exactly the most difficult subset of CMC sentences from an LLM perspective.

Overall, this has shown that while our hybrid pipeline is not perfect, the evaluation based on it still shows the general trend that most language models have large deficits in understanding the CMC, even though they are slightly underestimated.

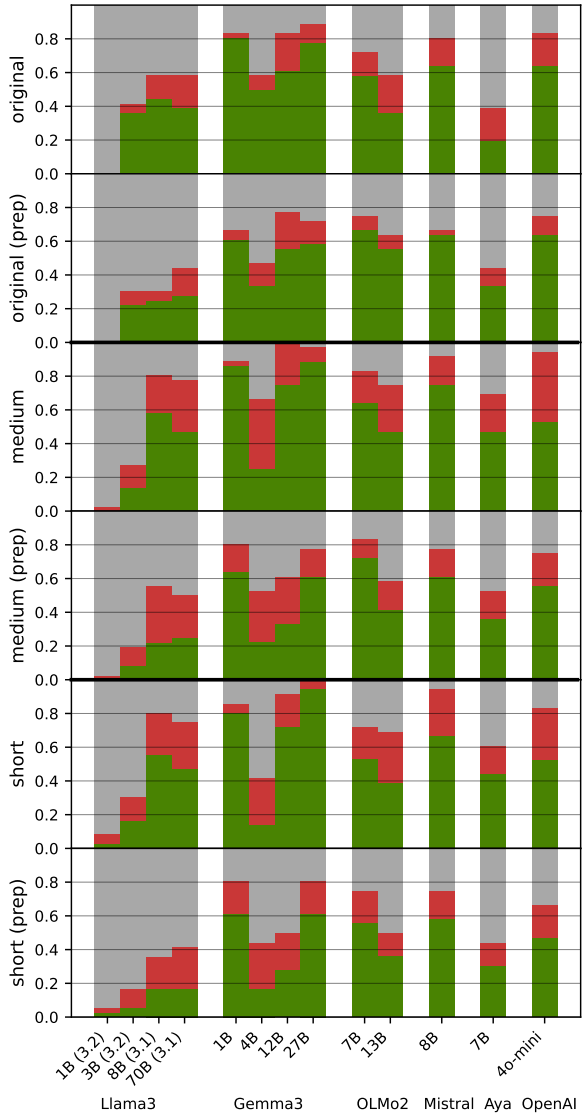


Figure 4: Results for each model and evaluation type. Examples for the evaluation types are given in Table 5. Correct answers are coloured in green, incorrect in red, and invalid results in grey.

5 Conclusion

We have introduced an annotation pipeline aided by dependency parsing and prompting LLMs, which can be specifically used for phenomena that are so rare that little to no corpora have been created, as the human annotation effort would be too great. We have demonstrated this pipeline on the example of the caused-motion construction, and a corpus of 500 caused-motion sentences.

We have used the manually annotated corpus to evaluate state-of-the-art LLMs for their understanding of the CMC, and found that many have high error rates when asked to interpret situations described with a non-prototypical CMC.

We hope that our work will inspire more computational and corpus-based studies of rare linguistic phenomena. We note that even though prompt engineering is complex, large gains can be achieved by using intermediate-complexity prompting setups and basic knowledge of LLMs. We are confident that further advances in instruction-tuned LLMs will make the cost-benefit ratio of incorporating them into this hybrid annotation pipeline even stronger.

We see several opportunities for interesting future work in both halves of the paper. For the data collection part, it is a promising engineering direction to develop tools that automate parts of this process so that it becomes available to linguists without the need for complex prompt engineering. Continued progress in LLMs is likely to make the process even more efficient.

Concerning the evaluation of LLMs' understanding of constructions, a straightforward direction for future work would be to expand to the other three Argument Structure Constructions described in Goldberg (1992).

Limitations

Due to cost reasons, the evaluation experiments were limited to replacing the verbs only with "throw". A further validation of the results could be achieve by repeating the experiment with several other prototypical motion verbs.

Because the evaluation prompts as shown in Table 5 are automatically generated, the resulting sentences might occasionally be slightly unnatural, which could affect how models reply to them.

Acknowledgements

We thank the anonymous reviewers for their thorough and constructive feedback.

References

Ahrenberg, Lars. 2007. LinES: An English-Swedish parallel treebank. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 270–273, Tartu, Estonia. University of Tartu, Estonia.

Aoyama, Tatsuya, Shabnam Behzad, Luke Gessler, Lauren Levine, Jessica Lin, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. 2023. GENTLE: A genre-diverse multilayer challenge set for English NLP and linguistic evaluation. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 166–178, Toronto, Canada. Association for Computational Linguistics.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Baumgartner, Jason, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.

Behzad, Shabnam and Amir Zeldes. 2020. A cross-genre ensemble approach to robust Reddit part of speech tagging. In *Proceedings of the 12th Web as Corpus Workshop (WAC-XII)*, pages 50–56.

Bencini, Giulia ML and Adele E Goldberg. 2000. The contribution of argument structure constructions to sentence meaning. *Journal of Memory and Language*, 43(4):640–651.

Bonial, Claire and Harish Tayyar Madabushi. 2024. A construction grammar corpus of varying schematicity: A dataset for the evaluation of abstractions in language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 243–255, Torino, Italia. ELRA and ICCL.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Chomsky, Noam. 1993. *Lectures on government and binding: The Pisa lectures.* 9. Walter de Gruyter.

Croft, William. 2001. *Radical construction grammar: Syntactic theory in typological perspective.* Oxford University Press, USA.

Dang, John, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, Acyr Locatelli, Hangyu Lin, Dwarak Talupuru, Bharat Venkitesh, David Cairuz, Bowen Yang, Tim Chung, Wei-Yin Ko, Sylvie Shang Shi, Amir Shukayev, Sammie Bae, Aleksandra Piktus, Roman Castagné, Felipe Cruz-Salinas, Eddie Kim, Lucas Crawhall-Stein, Adrien Morisot, Sudip Roy, Phil Blunsom, Ivan Zhang, Aidan Gomez, Nick Frosst, Marzieh Fadaee, Beyza Ermis, Ahmet Üstün, and Sara Hooker. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier.

Devlin, Jacob, 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.

Fillmore, Charles J, Russell Lee-Goldman, and Russell Rhodes. 2012. The framenet constructicon. In Hans Christian Boas and Ivan A Sag, editors, *Sign-based construction grammar*, pages 309–372. CSLI Publications Stanford.

Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Goldberg, Adele E. 1995. Constructions: A construction grammar approach to argument structure. *Chicago UP*.

Goldberg, Adele Eva. 1992. *Argument structure constructions*. University of California, Berkeley.

Gray, Morgan, Jaromir Savelka, Wesley Oliver, and Kevin Ashley. 2023. Can GPT alleviate the burden of annotation? In *Legal Knowledge and Information Systems*, pages 157–166. IOS Press.

Holter, Ole Magnus and Basil Ell. 2023. Human-machine collaborative annotation: A case study with GPT-3. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 193–206.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Hwang, Haerim and Hyunwoo Kim. 2023. Automatic analysis of constructional diversity as a predictor of efl students' writing proficiency. *Applied linguistics*, 44(1):127–147.

Hwang, Jena D. and Martha Palmer. 2015. Identification of caused motion construction. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 51–60, Denver, Colorado. Association for Computational Linguistics.

Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Koptyra, Bartłomiej, Anh Ngo, Łukasz Radliński, and Jan Kocoń. 2023. Clarin-emo: Training emotion recognition models using human annotation and ChatGPT. In *International Conference on Computational Science*, pages 365–379. Springer.

Kyle, Kristopher and Hakyung Sung. 2023. An argument structure construction treebank. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 51–62, Washington, D.C. Association for Computational Linguistics.

Li, Bai, Zining Zhu, Guillaume Thomas, Frank Rudzicz, and Yang Xu. 2022. Neural reality of argument structure constructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7410–7423, Dublin, Ireland. Association for Computational Linguistics.

Mahowald, Kyle. 2023. A discerning several thousand judgments: GPT-3 rates the article + adjective + numeral + noun construction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 265–273, Dubrovnik, Croatia. Association for Computational Linguistics.

de Marneffe, Marie-Catherine, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Misra, Kanishka and Kyle Mahowald. 2024. Language models learn rare phenomena from less rare phenomena: The case of the missing AANNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 913–929, Miami, Florida, USA. Association for Computational Linguistics.

OLMo, Team, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. 2 olmo 2 furious.

OpenAI. 2022. ChatGPT: Optimizing language models for dialogue.

Pangakis, Nicholas, Samuel Wolken, and Neil Fasching. 2023. Automated annotation with generative ai requires validation. *arXiv preprint arXiv:2306.00176*.

Rozner, Joshua, Leonie Weissweiler, Kyle Mahowald, and Cory Shain. 2025a. Constructions are revealed in word distributions.

Rozner, Joshua, Leonie Weissweiler, and Cory Shain. 2025b. Babylm's first constructions: Causal interventions provide a signal of learning.

Sanguinetti, Manuela and Cristina Bosco. 2015. Parttut: The turin university parallel treebank. In Roberto Basili, Cristina Bosco, Rodolfo Delmonte, Alessandro Moschitti, and Maria Simi, editors, *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*, pages 51–69. Springer International Publishing, Cham.

Savelka, Jaromir and Kevin D Ashley. 2023. The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts. *Frontiers in Artificial Intelligence*, 6.

Scivetti, Wesley, Tatsuya Aoyama, Ethan Wilcox, and Nathan Schneider. 2025. Unpacking let alone: Human-scale models generalize to a rare construction in form but not meaning.

Silveira, Natalia, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tayyar Madabushi, Harish, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets construction grammar. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Team, Gemma, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot,

Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 technical report.

Torrent, Tiago Timponi, Thomas Hoffmann, Arthur Lorenzi Almeida, and Mark Turner. 2023. *Copilots for Linguists: AI, Constructions, and Frames*. Cambridge University Press.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. Preprint, arXiv 2302.13971.

Tseng, Yu-Hsiang, Cing-Fang Shih, Pin-Er Chen, Hsin-Yu Chou, Mao-Chang Ku, and Shu-Kai Hsieh. 2022. CxLM: A construction and context-aware language model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6361–6369, Marseille, France. European Language Resources Association.

Warstadt, Alex, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Warstadt, Alex, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Weissweiler, Leonie, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your syntax, the better your semantics? probing pretrained language models for the English comparative correlative. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yu, Danni, Luyang Li, Hang Su, and Matteo Fuoli. 2023. Assessing the potential of llm-assisted annotation for corpus-based pragmatics and discourse analysis: The case of apologies. *International Journal of Corpus Linguistics*.

Zeldes, Amir. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Zeman, Daniel, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Zhou, Shijia, Leonie Weissweiler, Taiqi He, Hinrich Schütze, David R. Mortensen, and Lori Levin. 2024. Constructions are so difficult that Even large language models get them right for the wrong reasons. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3804–3811, Torino, Italia. ELRA and ICCL.

A Full details for each prompt

We report in Tables 7 to 24 the details of the prompt, along with the change that it represents from a previous prompt.

B Few Shots

In Table 41, we give the five shots from each class given to ChatGPT as examples.

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Reply with a csv codeblock (wrapped in three backticks), with the headers 'id' and 'label'. label should be either True or False. Label all 50 sentences.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	Base
Shot Strategy	all

Table 7: Prompt 1

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	1 + repeat sentence with json
Shot Strategy	all

Table 8: Prompt 2

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	2 + reason
Shot Strategy	all

Table 9: Prompt 3

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	3 + structured information
Shot Strategy	all

Table 10: Prompt 4

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	4 + sentence
Shot Strategy	all

Table 11: Prompt 5

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	4 + cmc string
Shot Strategy	all

Table 12: Prompt 6

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	4 + cmc string continuous
Shot Strategy	all

Table 13: Prompt 7

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	6 + sentence
Shot Strategy	all

Table 14: Prompt 8

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	0
Sentences	50
Model	4o_mini
Majority Vote	No
Change	7 + sentence
Shot Strategy	all

Table 15: Prompt 9

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 10 positive examples: . Here are 10 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	10
Sentences	50
Model	4o_mini
Majority Vote	No
Change	4 + few shots
Shot Strategy	first of each verb and class

Table 16: Prompt 10

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 10 positive examples: . Here are 10 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	10
Sentences	50
Model	4o_mini
Majority Vote	No
Change	10 + explanations
Shot Strategy	first of each verb and class

Table 17: Prompt 11

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	4o_mini
Majority Vote	No
Change	11 + all shots
Shot Strategy	all

Table 18: Prompt 12

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	10
Model	4o_mini
Majority Vote	No
Change	12 + only 10 samples
Shot Strategy	all

Table 19: Prompt 13

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	No
Change	12 + only 1 sample
Shot Strategy	all

Table 20: Prompt 14

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	5
Model	4o_mini
Majority Vote	No
Change	12 + only 5 samples
Shot Strategy	all

Table 21: Prompt 15

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	25
Model	4o_mini
Majority Vote	No
Change	12 + only 25 samples
Shot Strategy	all

Table 22: Prompt 16

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	No
Change	14 + new few-shots
Shot Strategy	all

Table 23: Prompt 17

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: . Here are 50 negative examples: . Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	No
Change	17 + alternating shots
Shot Strategy	all_alternating

Table 24: Prompt 18

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	No
Change	17 + grouped shots
Shot Strategy	all_grouped

Table 25: Prompt 19

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	4o_mini
Majority Vote	Yes
Change	19 + majority vote
Shot Strategy	all_grouped

Table 26: Prompt 20

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	1
Model	o3_mini
Majority Vote	No
Change	19 on o3-mini
Shot Strategy	all_grouped

Table 27: Prompt 21

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	100
Model	o3_mini
Majority Vote	No
Change	21 + 100 samples
Shot Strategy	all_grouped

Table 28: Prompt 22

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	250
Model	o3_mini
Majority Vote	No
Change	21 + 250 samples
Shot Strategy	all_grouped

Table 29: Prompt 23

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	o3_mini
Majority Vote	No
Change	21 + 50 samples
Shot Strategy	all_grouped

Table 30: Prompt 24

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 25 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 25 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	25
Sentences	50
Model	o3_mini
Majority Vote	No
Change	21 + 25 samples
Shot Strategy	all_grouped

Table 31: Prompt 25

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	o3_mini
Majority Vote	Yes
Change	24 + majority vote
Shot Strategy	all_grouped

Table 32: Prompt 26

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	4o
Majority Vote	No
Change	24 on 4o
Shot Strategy	all_grouped

Table 33: Prompt 27

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	o3_mini
Majority Vote	No
Change	24 - sentence
Shot Strategy	all_grouped

Table 34: Prompt 28

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	4o
Majority Vote	No
Change	27 - sentence
Shot Strategy	all_grouped

Table 35: Prompt 29

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	o3_mini
Majority Vote	No
Change	28 - reason
Shot Strategy	all_grouped

Table 36: Prompt 30

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes. Before you give the label, justify your decision with a reason.
Few-Shots	50
Sentences	50
Model	4o
Majority Vote	No
Change	29 - reason
Shot Strategy	all_grouped

Table 37: Prompt 31

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes.
Few-Shots	50
Sentences	100
Model	o1
Majority Vote	No
Change	22 on o1
Shot Strategy	all_grouped

Table 38: Prompt 32

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes.
Few-Shots	50
Sentences	50
Model	o1
Majority Vote	No
Change	32 + 50 samples
Shot Strategy	all_grouped

Table 39: Prompt 33

Instruction	The task is to classify whether the sentences contain instances of the caused-motion construction. The caused-motion construction is a construction where an agent causes an object to move. This motion has to be literal, not metaphorical. Each sentence that you will be given includes a subject, a verb, a direct object, and a prepositional phrase. In the caused motion instances, the verb causes the motion of the direct object, in the direction specified by the prepositional phrase. The action does not need to actually happen, it could be only mentioned or hypothetical or occur in the past or future.
Input Format	Here are 50 positive examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Here are 50 negative examples: { "sentence": "...", "verb": "...", "direct_object": "...", "preposition": "...", "prepositional_object": "...", "reason": ..., "label": ... }. Classify the following sentences: { "id": "...", "sentence": "..." }.
Output Format	Respond with a jsonl codeblock (wrapped in three backticks) using double quotes.
Few-Shots	50
Sentences	50
Model	o1
Majority Vote	Yes
Change	33 + majority vote
Shot Strategy	all_grouped

Table 40: Prompt 34

Sentence	Verb	Dir Obj	Prep	P-Obj	Lab.	Explanation
I actually giggled myself to tears .	giggle	myself	to	tear	False	This is a negative example because being 'in tears' is a state, not a location, so the subject here didn't move but rather changed state.
Nope , they just giggle their microscopic excretions into the air .	giggle	excretion	into	air	True	This is a positive example because the act of giggling is causing the excretions to move
I 'll stop it from repeating and fade it into a single background color .	fade	it	into	color	False	This is a negative example which describes the act of fading a color so that it can't be told apart from the background color, which means that nothing moved.
Just hover your mouse over it	hover	mouse	over	it	False	This is a negative example because the mouse is hovering over it, but it is not moving, it is staying in place while hovering.
Once she was strapped back in he started to hover her out of the room .	hover	she	out of	room	True	This is a positive example because they are moving her out of the room by hovering.
They tug him to the ground and start jumping on him and licking his face .	tug	he	to	ground	True	This is a positive example because someone was tugged and that moved him to the ground.
I gulped it from the bottle while watching old movies .	gulp	it	from	bottle	True	This is a positive example because what was in the bottle moved from the bottle because the person was drinking it.
I would rather take the spoon , I can gulp it in one go .	gulp	it	in	go	False	This is a negative example because one go is not a destination, it specifies the manner of gulping.
Cruz was trailing Clinton in basically every poll .	trail	Clinton	in	poll	False	This is a negative example because no movement is happening, the sentence describes the relative position of two politicians in a poll.
She began trailing a finger down his chest .	trail	finger	down	chest	True	This is a positive example because she is moving the finger down his chest.
He stopped for ten minutes while wheezing himself to death .	wheeze	himself	to	death	False	This is a negative example because death is a state, not a physical location.
It is not cute to watch your dog wheeze himself to the floor because he was so excited you picked up his tug of war rope .	wheeze	himself	to	floor	True	This is a positive example because the wheezing is causing the dog to move to the floor.
I bounced it off the wall .	bounce	it	off	wall	True	This is a positive example because the ball moved off the wall.
We bounce ideas off each other .	bounce	idea	off	other	False	This is a negative example because an idea can't physically move.
I was in bed for about a week and thought I was going to shiver myself to death .	shiver	myself	to	death	False	This is a negative example because death is a state, not a destination of a physical movement.
She needs to stop darting her eyes to the side every time she says something	dart	eye	to	side	False	This is a negative example because her eyes are rotating but they're not moving.
He nervously darted his tongue into her mouth .	dart	tongue	into	mouth	True	This is a positive example because his tongue is moving into her mouth.
For some reason every time i overflow the sink in Dalia 's bathroom , the Sheik always comes up to investigate ...	overflow	sink	in	bathroom	False	This is a negative example because the sink is not moving.
Most importantly the toilet was overflowing water into the pan , almost on constant flush .	overflow	water	into	pan	True	This is a positive example because the water is moving into the pan.
You 're trying to wriggle your way out of it now !	wriggle	way	out of	it	False	This is a negative example because while something is moving, it is not the direct object way.
At one point he wriggles himself into position to block a soccer ball with his head while Latin on the street .	wriggle	himself	into	position	True	This is a positive example because he is moving himself into position.
I swim laps in the pool .	swim	lap	in	pool	False	This is a negative example because while I am moving, the laps are not moving.
My wife lapped me on the scoring track .	lap	I	on	track	False	This is a negative example because I am moving, but my wife is not causing me to move.
He will nibble you to death !	nibble	you	to	death	False	This is a negative example because death is a state, not a location.
I eat my Twix by nibbling the chocolate off the sides , then off the top , then eat the caramel and cookie .	nibble	chocolate	off	side	True	This is a positive example because the chocolate is moving off the sides.
I aimed at her , and gazed her in her eyes before I successfully hit her face with the snowball .	gaze	she	in	eye	False	This is a negative example because a gaze is not something that can physically move.
I choose to be the one that goes hiking with friends into waterfalls , out galloping horses in open fields , and having fun times with my SO .	gallop	horse	in	field	False	This is a negative example because the horses are moving, but they are not moving in the direction of the field, they are already in it.
I can confirm that galloping a horse through an open field is amazing .	gallop	horse	through	field	True	This is a positive example because the horse is moving through the field.
I scramble them in the hot pan .	scramble	they	in	pan	False	This is a negative example because the eggs are not moving in the direction of the pan, they are already in it.
Once it firms a little , scramble it into the rice .	scramble	it	into	rice	True	This is a positive example because the eggs are moving into the rice.
To continue with your explanation , we see not only that this man here can afford to encrust rare and obviously expensive jewels onto his box of ' Fruity Pebbles ' brand breakfast cereal , but also that he can afford the ' Family Size ' box .	encrust	jewel	onto	box	True	This is a positive example because the jewels are moving onto the box.
I peel paint off walls .	peel	paint	off	wall	True	This is a positive example because the paint is moving off the wall.
I peel bananas from the bottom	peel	banana	from	bottom	False	This is a negative example because the banana is not moving, only the peel is, and it is not moving from the bottom.
In my defense I was actually very drunk when I plowed my car into that crowd of pedestrians .	plow	car	into	crowd	True	This is a positive example because I caused tha car to move into the crowd.
I plow snow in the winter	plow	snow	in	winter	False	This is a negative example because in the winter is a time, not a location.
And drag queens cake themselves in makeup .	cake	themselves	in	makeup	False	This is a negative example because the drag queens are not moving.
I would cake makeup on my face to hide it .	cake	makeup	on	face	True	This is a positive example because the makeup is moving onto the face.
Whereas WWE charred it to a crisp and drowned it in A-1 sauce .	char	it	to	crisp	False	This is a negative example because it is changing state to a crips, not moving.
I fermented it in a 3 gallon food grade plastic bucket .	ferment	it	in	bucket	False	This is a negative example because it is staying in the bucket and not moving.
When the child collapsed , the mother hurried him to the hospital , where he died .	hurry	he	to	hospital	True	This is a positive example because the child is moving to the hospital.
I will take my time or hurry you through a meal , there are no rules against that .	hurry	you	through	meal	False	This is a negative example because the meal here is an action, not a destination
I love blackening it in a roasting pan .	blacken	it	in	pan	False	This is a negative example because it is not moving, it is staying in the pan.
I rarely use them but my girlfriend is crocheting them into reusable shopping bags ...	crochet	they	into	bag	False	This is a negative example because the bags are not moving, they are being made into something else.
When " nice guys " change their MO to target " nice girls " the equilibrium will tilt the earth off its axis and hurtle us into space , thus settling this tired argument for all eternity .	hurtle	we	into	space	True	This is a positive example because we are moving into space.
Then you drip juice into it and vape .	drip	juice	into	it	True	This is a positive example because the juice is moving into it.
As in you literally gnaw it off the bone .	gnaw	it	off	bone	True	This is a positive example because it is moving off the bone
I twitch my head to the side .	twitch	head	to	side	True	This is a positive example because the head is moving to the side.
He snorted coke off my ass	snort	coke	off	ass	True	This is a positive example because the coke moved off my ass.
I ca n't tell if she 's smiling or is she 's about to sneeze the sand off of her nose .	sneeze	sand	off of	nose	True	This is a positive example because the sand moves off her nose.
It was like a little rocket that tried to burrow itself into the ground .	burrow	itself	into	ground	True	This is a positive example because the rocket moves into the ground.

Table 41: Few shots. P-Obj stands for Prepositional Object, Dir Obj for Direct Object.