Uncovering Autoregressive LLM Knowledge of Thematic Fit in Event Representation

Safeyah Khaled Alshemali
Imperial College London
London, UK
ska24@ic.ac.uk

Daniel Bauer
Columbia University
New York, NY, USA
db2711@columbia.edu

Yuval Marton
University of Washington
WA, USA
ymarton@uw.edu

Abstract

The thematic fit estimation task measures semantic arguments' compatibility with a given semantic role for a given predicate. We investigate if autoregressive LLMs have consistent, expressible knowledge of event arguments' thematic fit by experimenting with various prompt designs, manipulating input context, reasoning, and output forms. We set a new state-of-the-art on thematic fit benchmarks, but show that closed and open weight LLMs respond differently to our prompting strategies: Closed models achieve better scores overall and benefit from multi-step reasoning, but they perform worse at filtering out generated sentences incompatible with the given predicate, role, and argument. Our analysis shows that lemma tuple input and sentence input result in surprisingly different thematic fit score distributions.

1 Introduction

Do large language models (LLMs) have consistent, expressible knowledge of event arguments' thematic fit to a semantic role for a given predicate?

Thematic fit is the level of compatibility between the predicate (typically a verb), its argument (typically a noun phrase), and a semantic role assigned to the argument. For example, the verb 'eat' invokes a set of potential arguments for specific semantic roles. If the role is *Agent* (or *Arg0* in PropBank; see Section §3.2), then 'people', 'cat', etc., would be a good fit, as opposed to 'pizza', 'apple', etc., as the latter group would better fit a *Patient* role (often denoted as *Arg1* in PropBank). If the role is *Location*, then 'restaurant', 'kitchen', etc., would be a good fit due to their properties. If the role is *Instrument*, then 'fork', 'knife', etc. would be a good fit due to their use as tools while eating. Psycholinguistic experiments show that humans rely on such generalized, prototypical knowledge about events and their typical participants during language understanding, early in the process and prior to syntactic interpretation (McRae and Matsuki, 2009; Bicknell et al., 2010).

Thematic fit estimation (TFE) applies computational methods and event representations derived from data to mimic this type of human knowledge. This is challenging due to the lack of direct labeled training data. Thematic fit norms (see §3.2) summarize human ratings and provide a benchmark for evaluation, but are not large enough for direct supervised training. Instead, TFE models are traditionally trained on SRL datasets as proxy data.[1] Recently, pretrained language models have shown state-of-the-art performance without task-specific supervised training on many computational semantics tasks, such as synonym judgment (Levy et al., 2017), similarity judgments (Hill et al., 2015), and more. However, on TFE, these models, including masked LMs such as BERT (Devlin et al., 2019), have so far only yielded modest improvements over previous approaches (Pedinotti et al., 2021). In this work, we examine if autoregressive LLMs, such as GPT (Radford et al., 2018; Achiam et al., 2023), can produce better results via prompting. We design our experiments to elicit linguistic knowledge along the following three axes (see §3.1 for details):

Axis 1: Reasoning (Simple vs. Step-by-Step Prompting) LLMs perform remarkably well in challenging NLP tasks such as Planning (Wang et al., 2022), automatic scoring (Lee et al., 2024), and question answering (Wang et al., 2023), when using 'chain-of-thought prompting' (Wei et al., 2022). This tech-

[1] TFE is related to semantic role labeling (SRL). SRL outputs the semantic frames in the input sentence, using frameworks such as PropBank (Kingsbury and Palmer, 2002) or FrameNet (Johnson et al., 2016). The output includes the predicates, arguments, and their semantic roles in the frame. TFE takes the predicates, arguments, and roles as input, and outputs a score. Understanding the thematic fit of individual arguments/roles is thought to be required for SRL, but is not usually modeled explicitly in SRL systems. For theoretical and historical details, see Gruber (1965); Fillmore (1968); Dowty (1991); Parsons (1990); McRae et al. (1998) *inter alia*.

nique breaks a problem into sub-steps toward answering the core question. It seems to help LLMs in a similar way a scratchpad would help someone solve a math exercise.[2] We view TFE as a linguistic *reasoning* task because it can be broken into clear sub-steps. Thus, we compare the LLMs' performance using a "naive" single-prompt approach (simply asking the model to directly score a given lemma tuple; **Simple Prompting** hereafter), with the chain-of-thought approach (**Step-by-Step Prompting** hereafter).

Axis 2: Input (Generated Sentences vs. Lemma Tuples) The TFE norm data comes in a lemma tuple format containing a predicate lemma, lemma of the syntactic head of the predicate's argument, its thematic role, and the argument's thematic fit score (an average of several human raters' scores). For example, ⟨predicate: eat, argument: pizza, role: *Arg0/Agent*, score:1.3⟩. Since LLMs were pretrained on text rather than such tuples, we hypothesize that they could benefit from seeing full sentences that use the predicate and argument in the given role. We prompt the LLM to generate such sentences and use a novel approach for filtering out semantically incoherent and incompatible sentences.[3]

Axis 3: Output (Predefined Categories vs. Numeric Score) The thematic fit ratings in the TFE norms are Likert scale ratings (following McRae et al. (1998)) averaged over 7 human raters. A straightforward evaluation would prompt the model to output numeric scores as well. However, LLMs were designed to handle textual data, so outputting numeric scores may lead to inconsistent results. Therefore, we also prompt the LLMs to rate the thematic fit according to predefined categories ("Low", "High", etc.), which we then convert to corresponding predefined numeric values for evaluation purposes (see Section §3.2).[4]

Our experiments compare the performance of two closed-weight LLMs (GPT4.1 and GPT4-Turbo) and two open LLMs (Llama3.2 and Qwen2.5). We show closed LLMs achieve state-of-the-art results and demonstrate the benefits of Step-by-Step Prompting on the TFE task. In contrast, open LLMs demonstrate worse zero-shot performance on this task, but may have an edge on filtering out generated sentences if incompatible with the specified predicate, role, and argument.

Contributions Our main contributions are:

- First study to prompt autoregressive LLMs in order to estimate thematic fit from output tokens.
- New state-of-the-art thematic fit estimation.
- Comparison of prompting techniques along three axes: reasoning form, input form (with/out LLM-filtered generated sentences), and output form.
- An analysis of related strengths and weaknesses of various open and closed weight LLMs.

Source code is available at: `https://github.com/SafeyahShemali/LLM_Thematic_Fit_25`

2 Related Work

TFE is closely related to Selectional Association (Resnik, 1996), which is an information-theoretic measure of the difference in the probability that some noun class will appear as a specific syntactic argument of the given predicate, compared to the general probability of this class.[5]

Early distributional approaches to the TFE task compute the similarity between vector representations of an argument (usually for the argument's syntactic head word, obtained from a distributional semantic model or static word embeddings) and a prototype vector for the predicate/role pair (Padó et al., 2007; Erk et al., 2010; Baroni and Lenci, 2010; Santus et al., 2017; Sayeed et al., 2015; Greenberg et al., 2015). Our first baseline (hereafter **BG**) is the distributional approach by Greenberg et al., which is based on the syntactically structured distributional semantic model by Baroni and Lenci. Following approaches use neural networks to learn event representations. Tilk et al. (2016) describe a *non-incremental role filler* model (NN RF, baseline **B0**) trained to predict role fillers, given as input a predicate, target role, and optionally other contextual roles and their fillers. Hong et al. (2018) improve over Tilk et al. (2016) by adding role

[2] It proved so influential that several leading labs have developed 'thinking' models with 'test-time compute', incorporating this approach in the post-training and settings.

[3] Upon a close reading, we find that the human raters were indeed provided full sentences for the rating (McRae et al., 1998), but this fact seems to have been lost over the years, and only the lemma tuple format has been used (Tilk et al., 2016; Hong et al., 2018; Muthupari et al., 2022). We close here a full circle, bringing the conditions closer to the original.

[4] We leave for future work whether it is useful to use LLMs' logprobs for this task, and how to best do that.

[5] It requires a taxonomy (WordNet) for the noun classes and a syntactically parsed corpus for estimating probabilities. Thematic fit focuses on the cognitive load of understanding an utterance with the given predicate and argument(s) in specific thematic roles, *post-hoc*. TFE also uses a corpus, but with the advent of LLMs, no explicit parses or noun classes are needed, hence it is more applicable and fine-grained.

prediction as a multi-task objective, and improving pooling of context roles/fillers (ResRoFA-MT, baseline **B1**). Marton and Sayeed (2021) replicate Hong et al. (2018)'s model and train it on higher quality silver labels, experimenting with the quality/quantity tradeoff of annotations (baseline **B2**). Muthupari et al. (2022) focus on evaluating the capability of various models, or model elements, to capture thematic fit. They compare initializing the Hong et al. (2018) model with random vectors vs. pre-trained GloVe embeddings (Pennington et al., 2014), and other modifications. We add three of their higher performing models to our baselines: (**B3a**) Glove-shared-not-tuned, (**B3b**) Glove-shared-tuned, (**B3c**) Glove-shared-tuned-20%M. B3a and B3b report the average scores over several runs, whereas B3c is identical to B3b except for reporting the maximum score over several runs, all trained on the same 20% subset of the data.

Contemporary work using pretrained LLMs (without any additional supervised training) for TFE either examines if these models can differentiate plausible from implausible sentences ("the teacher bought the laptop" vs. "the laptop bought the teacher"; Kauf et al. (2023)), or used BERT-style models to predict role fillers for a masked argument in the context of a generated sentence (Metheniti et al., 2020; Pedinotti et al., 2021). Notably, Pedinotti et al. only report a moderate improvement over a distributional approach that uses a high quality structured distributional model (Chersoni et al., 2019). Vassallo et al. (2018) introduce DTFit, a dynamic evaluation of thematic fit as more words are added to a sentence already containing an *Agent* and a predicate (verb). Like us, they use autoregressive LLMs but focus on *Agent* with other roles, use LLM logprobs, not using tuples as input, and are constrained to specific sentence structures. Testa et al. (2023) construct ELLie, a dataset including various types of elliptical constructions with typical, atypical, anomalous completions, based on Culicover and Jackendoff (2005) along with DTFit, to examine the interaction between thematic fit and the ability of LLMs to resolve verbal ellipsis. Kauf et al. (2024) compare base and instruction-tuned LLMs' ability of estimating event plausibility.

We are the first to compare autoregressive LLMs' TFE over ⟨predicate, argument, role⟩ tuples vs. generated (and filtered) sentences; simple prompting vs. chain-of-thought reasoning; and prompting for numeric output vs. pre-defined categories.

3 Experiments

The experiments were structured along three axes (see §1) with two possible values each. Therefore, we conduct $2^3 = 8$ experiments. We refer to them as 'Exp $n.m$' (see Table 1). Figure 1 shows a schematic overview of the different methods. Each experiment has a prompting template according to the reasoning, input, and output settings, as detailed in Appendix §A.

3.1 Method: Reasoning, Input, and Output

Axis 1: Reasoning (Simple Prompting vs. Step-by-Step Prompting) The first axis is concerned with the effectiveness of Step-by-Step Prompting (Exp.3.x and 4.x) compared to Simple Prompting (Exp.1.x and 2.x) in evaluating the thematic fit. In the Simple Prompting approach we use a single zero-shot prompt to provide the model with a predicate, argument, and semantic role, and ask it to rate the thematic fit. Step-by-Step Prompting decomposes the task into multiple steps, each inferring properties needed to be considered for scoring the thematic fit of the given tuple ⟨predicate, argument, semantic role⟩. These steps are: **R1.** listing the argument's (salient or relevant) properties, **R2.** listing the (prototypical or necessary) properties required for the argument in the role "assigned" by the given predicate, **R3.** listing the (suitable or likely) roles for the argument given predicate, **R4.** listing the argument's properties that fit well or are missing for the specified role given the predicate, **R5.** and finally, estimating the argument's thematic fit to the assigned role, given these four lists. The last step is similar to the 'simple thematic fit prompt' in the Simple Prompting setting.

Axis 2: Input (Lemma Tuples vs. Generated Sentences) The input in each experiment can take one of two forms. The basic form (available in the TFE norm data and used in previous work including our baselines) is a ⟨predicate, argument, semantic role⟩ tuple, where the predicate is a verb lemma and the argument is represented by the lemma form of its

Reasoning Form	Input Form	Output Form	Exp
Simple Prompting	Head Lemma Tuples	Numerical	1.1
		Categorical	1.2
	Generated Sentences	Numerical	2.1
		Categorical	2.2
Step-by-Step Prompting	Head Lemma Tuples	Numerical	3.1
		Categorical	3.2
	Generated Sentences	Numerical	4.1
		Categorical	4.2

Table 1: Overview of the experiments.

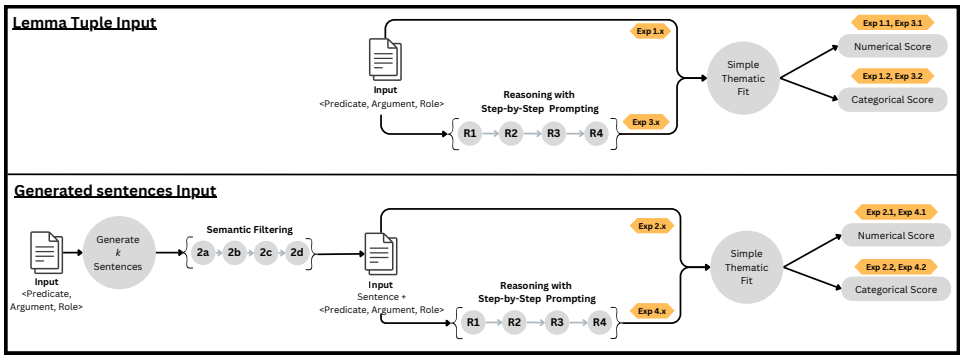

Figure 1: Experiment Method. For more details of the Step-by-Step Prompting and Semantic Filtering, see §3.1. The output of Exp.3.x - Exp.4.x contains the score's justification in addition to the score.

syntactic head. Adjuncts, subordinate clauses, etc., are ignored. The semantic role is taken from either PropBank or VerbNet. For example, for the sentence "I ate a pizza with my friends" we would see ⟨eat, I, *Agent*⟩ or ⟨eat, pizza, *Arg1*⟩. The **Lemma Tuples** setting (Exp.1.x and 3.x) uses only the lemmas, as in previous work[6] In the **Generated Sentences** setting (Exp.2.x and 4.x), a sentence generation step happens before the other prompts. We hypothesize that providing more context, similar to what the LLM presumably was trained on, will improve performance. Sentence generation consists of two steps: **1. Generation:** prompting the LLM to generate $k = 5$ candidate sentences with the given predicate, argument, and role (see §A), and **2. Filtering:** instructing the LLM to confirm each generated sentence's semantic cohesion and its use of the specified predicate and argument in its specified semantic role. The second step is needed, as the models (as tested in preliminary trials) seem to often generate semantically incoherent sentences, or use the given argument not in the given semantic role. Each sentence is given three trials to pass the filtering process, which consists of 4 waterfall Yes/No steps: **(2a)** is the given sentence semantically coherent, **(2b)** does the sentence contain the predicate, **(2c)** does the sentence contain the argument as the given predicate's argument, and **(2d)** is the argument in the required role for the predicate (see §A). The sentence must pass all steps of the filtering process in at least one trial, or it will be filtered out. The final fit score is an average over the remaining sentences' scores. If steps (1) or (2) fail, we back off to using the lemma tuples.

[6]In McRae et al. (1998), the raters were given full sentences, but they were not made public as far as we know (see footnote 3).

Axis 3: Output (Categorical vs. Numeric Scoring) Each experiment may generate one of two forms of output: a numeric thematic fit score (Exp.x.1 in Table 1) or a predefined thematic fit category (Exp.x.2). The numeric thematic fit score (numeric score, hereafter) is a numeric value in the range [0,1]. The predefined thematic fit categories (categorical score, hereafter) are 'Unlikely/Impossible', 'Low', 'Medium', 'High', or 'Perfect'. These output score categories are then converted to the following numeric values: (0.0, 0.25, 0.5, 0.75, 1.0), respectively, in a post-processing script, for evaluation purposes (see §3.2). In Exp.3.x and Exp.4.x, the model is asked to generate a justification for the score it assigns to the input, based on its previous Step-by-Step Prompting reasoning output.

3.2 Datasets and Evaluation

Training set: We used pre-trained LLMs (see Section §3.4) "off the shelf" without modification. In an ideal world, we would be able to point the reader to the full pre-training data. We did not add psycholinguistic (or other) labeled data to the training set, nor did we fine-tune the LLMs.

Test sets: We used the following datasets to evaluate our methods.[7] Each of these test sets comprises a list of ⟨predicate, argument, semantic role, score⟩ tuples. For example, ⟨eat, restaurant, *Location*, 0.9⟩. The first two are the representative lemma form of the syntactic head of the predicate and argument (in some sentence), the semantic role is as listed below, and the score is the ground truth (human rating). These tuples are given to the LLMs as input, except for the score, which is used to evaluate the models' output (details below).

[7]These datasets are too small to be split to training, validation and test sets, as is common in machine learning. We follow here previous TFE literature.

McRae et al. (1998), hereafter **'McRae',** contains $1,444$ predicate-argument-role-rating tuples, developed with the participation of 120 native English speakers undergraduate students at the University of Rochester. Students were asked to rate the role typicality (thematic fit) of a predicate (verb) and the argument (noun) in sentence fragments on a Likert scale of 1 to 7. The public dataset does not contain the sentence fragments. The roles are PropBank's *Arg0* and *Arg1*, which are roughly equivalent to *Agent* and *Patient*, respectively (Kingsbury and Palmer, 2002). We removed eight duplicates.

Padó et al. (2006), hereafter **'Pado'**, contains 414 tuples in the same form as McRae et al. (1998); however, the role could also be *Arg2*. To prevent bias in rating the same verb-noun with different roles, one hundred native English speaker volunteers judged tuples in four randomly ordered lists.

Ferretti et al. (2001) evaluate thematic fit for the roles *Instrument* (248 tuples; hereafter, the **Fer-Ins** dataset) and *Location* (277 tuples; hereafter, **Fer-Loc** dataset). Scores were collected from fifty-eight students. The rating scale is the same as in McRae et al. (1998). For *Instrument*, the participants were asked, "How common is it for someone to use each of the following to perform the *eating* action?" For *Location*, the question was, "How common is it for someone to *eat* in each of the following locations?". We removed indefinite articles in Fer-Loc arguments, in order to consider only the lemma of the syntactic head of the argument (not the phrase).

Evaluation: For each test set, we used Rank Correlation Coefficient (ρ) (Spearman, 1904) between two lists: the test set, sorted by the average human ratings for each item, and the corresponding model output. Categorical output was converted to numeric values, so it can be sorted similarly too before conducting this evaluation (see §3.1).

3.3 Hyperparameters

We conducted a preliminary study to determine good temperature and top_p parameter values for the LLMs. We used GPT-3.5 on a subset of the Fer-Ins dataset (20 samples $\sim 8\%$) and ran a parameter sweep over temperature values (0.0, 0.5, 0.9) and top_p values (0.5, 0.7, 0.95). Temperature values were chosen near-uniformly across the scale to examine a wide range of the model creativity setting. Higher values correspond to higher randomness (aka "creativity") in the output. High values of top_p limit the randomness in the output to more probable tokens. Since we were more interested in accurate output, and less interested in creative writing for our purposes here, we tested top-p threshold values on the higher half of the scale. Temperature 0.0 and top_p 0.95 showed the highest correlation coefficient with human judgments on our sample. We set the max_tokens parameter based on the length of the prompt and its expected output. For experiments with Lemma Tuples input, we set max_tokens to 100, whereas for Generated Sentences input, we set it to 300 (for Simple prompting). As Step-by-Step Prompting is expected to be lengthy to clarify the underlying reasons, we set max_tokens to 600 there. We used these settings in all experiments.

3.4 LLMs

We used closed and open LLMs, the latest available at the time of running the experiments. **Closed models** (often best performing, but less reproducible): OpenAI gpt-4-0125-preview (Achiam et al., 2023; OpenAI, 2024) with unconfirmed 1.76 trillion parameters (hereafter **'GPT4-Turbo'**) and *gpt-4.1-2025-04-14* (OpenAI, 2025) with unconfirmed 1.8 trillion parameters (hereafter **'GPT4.1'**). **Open models** (more controlled and reproducible, even if not fully open): *llama3.2:3b-instruct-q4_K_M* (Dubey et al., 2024; Meta, 2024), a quantized version of Meta Llama3.2, 3 billion parameters (hereafter **'Llama'**) and *qwen2.5:7b-instruct-q4_K_M* (Yang et al., 2024) with 7.61 billion parameters (hereafter **'Qwen'**). Closed model inference utilized the OpenAI API. Open model inference utilized Ollama (Ollama Developers, 2023).

4 Results

4.1 Closed Models

Overall, our results (Table 2) set a new state-of-the-art for all four tasks, especially with GPT4-Turbo using Step-by-Step Prompting. Both GPT4-Turbo and GPT4.1 outperformed the baselines by a large margin, except for Pado (where GPT4-Turbo tied with the highest baseline, and GPT4.1 was weaker than most baselines). GPT4-Turbo demonstrated superior performance across the board, outperforming also the newer GPT4.1. Along the three axes, the model performance differed as follows:

Axis 1: Reasoning Form Simple Prompting with GPT4-Turbo outdid all baselines (except Pado), while results with GPT4.1 were mixed, in spite of this model being newer.

Model \ Dataset	Fer-Loc	Fer-Ins	Pado	McRae
BG: GSD2015	.29	.42	.53	.36
B0: NN RF	.44	.45	.52	.38
B1: ResRoFA-MT	.46	.48	.53	.43
B2: 20% subset v2	-	-	.43	.44
B3a: Glove-shared-not-tuned	.30	.34	.49	.31
B3b: Glove-shared-tuned	.29	.29	.53	.33
B3c: Glove-shared-tuned with 20%M	.35	.43	**.59**	.43
Exp.1.1	.63	.64	.35	.54
Exp.1.2	.61	.65	.40	.58
Exp.2.1	.48	.49	.42	.55
Exp.2.2	.46	.54	.45	.56
Exp.3.1	**.69**	**.70**	.48	.65
Exp.3.2	.68	.68	.46	**.66**
Exp.4.1	.57	.66	.58	.58
Exp.4.2	.58	.66	**.59**	.59

Table 2: GPT4-Turbo, Spearman's Rank Correlation (ρ). BG = Greenberg et al. (2015), B0 = Tilk et al. (2016), B1 = Hong et al. (2018), B2 = Marton and Sayeed (2021), B3 = Muthupari et al. (2022) as baselines for our experiments. Subscripts a-c denote specific models there (Glove-shared-not-tuned, etc.). Exp.$m.n$ = see Table 1. All ρ's had p-values $< 10^{-13}$.

Model \ Dataset	Fer-Loc	Fer-Ins	Pado	McRae
BG: GSD2015	.29	.42	.53	.36
B0: NN RF	.44	.45	.52	.38
B1: ResRoFA-MT	.46	.48	.53	.43
B2: 20% subset v2	-	-	.43	.44
B3a: Glove-shared-not-tuned	.30	.34	.49	.31
B3b: Glove-shared-tuned	.29	.29	.53	.33
B3c: Glove-shared-tuned with 20%M	.35	.43	**.59**	.43
Exp.1.1	.51	.56	.40	**.64**
Exp.1.2	.50	.56	.40	.58
Exp.2.1	.32	.33	.33	.40
Exp.2.2	.26	.37	.33	.37
Exp.3.1	**.59**	**.59**	.45	**.64**
Exp.3.2	.56	**.59**	.42	.63
Exp.4.1	.56	.53	.42	.44
Exp.4.2	.53	.53	.42	.46

Table 3: GPT4.1, Spearman's Rank Correlation (ρ). Legend is as in Table 2. All ρ's had p-values $< 10^{-5}$.

For both GPT4-Turbo and GPT4.1, Step-by-Step Prompting (Exp.3.x and 4.x) yielded the best scores, demonstrating the effectiveness of this approach. For GPT4-Turbo, absolute improvement over the strongest baseline was .23 for Fer-Loc, .22 for Fer-Ins, .22 for McRae, and a tie with Pado.

Axis 2: Input Form For both closed models, using Generated Sentences (Exp.2.x and 4.x) rather than Lemma Tuples was detrimental. The only exception was for Pado with GPT4-Turbo, where Generated Sentences with Step-by-Step Prompting resulted in our highest score for Pado.

Axis 3: Output Form Using predefined categories compared to numeric output (Exp.x.2 vs. Exp.x.1) had zero, or small mixed effect, contrary to our hypothesis. With GPT4-Turbo, using predefined categorical output compared to numeric output (Exp.x.2 vs. Exp.x.1) leads to a mixed effect across the board. Regardless of input and reasoning forms, the categorical score is lower than the numerical score in Fer-Loc. For Fer-Ins and Pado, the categorical score is either the same or higher than the numerical score, as the gain in Fer-Ins was .04 in Exp2 and in Pado was .04 in Exp3. Using categorical output in McRae lowers the score in general, except for Exp4, where the gain was .02.

4.2 Open Models

The performance of the open models was lower than the closed models, and lower than several baselines across the board, except for Fer-Loc using Step-by-Step Prompting with Llama and McRae with Qwen (mostly in Exp.4.x). Details are below:

Axis 1: Reasoning Form Despite most results being lower than the baselines, Step-by-Step Prompting (Exp3.x and 4.x) had a positive effect in Fer-Loc and McRae. In Pado, the effect was detrimental with Llama (Table 4), but positive with Qwen (Table 5).

Axis 2: Input Form Using Generated Sentences (Exp.2.x and 4.x) instead of Lemma Tuples had mostly no effect with Llama, except for gains in McRae. With Qwen there were mostly gains except for lower scores in Exp.4.x in FerLoc.

Axis 3: Output Form Using predefined categories in the output, compared to numeric output (Exp.x.2 vs. x.1), yielded lower scores with Llama, except in Pado Exp.3.x and 4.x, but the correlation scores there are too low to trust these differences are meaningful. With Qwen results were worse in Fer-Ins and McRae, and mixed elsewhere.

5 Discussion

The core questions we aim to address in this work are: Do LLMs have the kind of linguistic knowledge required for TFE? If so, **(Q1)** does Step-by-Step Prompting help utilize the model's internal

Model \ Dataset	Fer-Loc	Fer-Ins	Pado	McRae
BG: GSD2015	.29	.42	.53	.36
B0: NN RF	.44	.45	.52	.38
B1: ResRoFA-MT	**.46**	**.48**	.53	.43
B2: 20% subset v2	-	-	.43	**.44**
B3a: Glove-shared-not-tuned	.30	.34	.49	.31
B3b: Glove-shared-tuned	.29	.29	.53	.33
B3c: Glove-shared-tuned with 20%M	.35	.43	**.59**	.43
Exp.1.1	.32	.33	.27	.16
Exp.1.2	.30	.23	.26	.13
Exp.2.1	.32	.33	.31	.25
Exp.2.2	.29	.22	.22	.23
Exp.3.1	**.46**	.29	.05*	.23
Exp.3.2	.41	.24	.10*	.20
Exp.4.1	**.46**	.28	.06*	.27
Exp.4.2	.41	.24	.12*	.21

Table 4: Llama, Spearman's Rank Correlation (ρ). Legend is as in Table 2. All ρ's had p-values $< 10^{-3}$ except cells with * for which p-values were between [.01, .22].

Model \ Dataset	Fer-Loc	Fer-Ins	Pado	McRae
BG: GSD2015	.29	.42	.53	.36
B0: NN RF	.44	.45	.52	.38
B1: ResRoFA-MT	**.46**	**.48**	.53	.43
B2: 20% subset v2	-	-	.43	.44
B3a: Glove-shared-not-tuned	.30	.34	.49	.31
B3b: Glove-shared-tuned	.29	.29	.53	.33
B3c: Glove-shared-tuned with 20%M	.35	.43	**.59**	.43
Exp.1.1	.29	.43	.28	.42
Exp.1.2	.38	.37	.19	.34
Exp.2.1	.31	.44	.31	.48
Exp.2.2	.37	.38	.22	.39
Exp.3.1	.41	.36	.32	.40
Exp.3.2	.41	.32	.36	.35
Exp.4.1	.31	.37	.33	**.51**
Exp.4.2	.22	.36	.37	.47

Table 5: Qwen, Spearman's Rank Correlation (ρ). Legend is as in Table 2. All ρ's had p-values $< 10^{-3}$.

linguistics knowledge for this task? **(Q2)** Would providing more context by adding sentences to the input help? (as opposed to providing only lemma tuples and roles) **(Q3)** Do predefined thematic fit output categories better elicit the model's internal linguistic knowledge compared to numeric output?

Overall, we conclude that closed models acquired much of the linguistic knowledge required for TFE. Results with both GPT models on each of the test sets achieved or even surpassed all previous baselines, which were trained on labeled linguistic

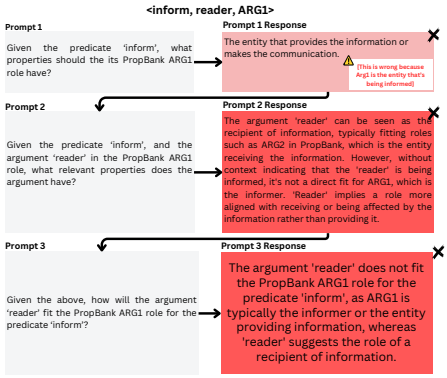

Figure 2: Effect of Early Bad Reasoning. The example was taken from preliminary experimentation.

data (even if not directly annotated for TFE, such as SRL). This is impressive,[8] assuming closed models were not trained on this task (sadly, we cannot know for sure because they are, well, closed). Next, we discuss the findings by the three axes (as defined in §3) and then move to analyzing additional aspects.

(Q1) Axis 1: Reasoning Form For GPT models (closed models), Step-by-Step Prompting fulfilled its intended goal of providing a "scratchpad" to help them calculate inferences given their internal knowledge (with the exception of GPT4.1 in Pado). In contrast, Step-by-Step Prompting didn't help with open models (except for Fer-Loc with Llama and for McRae with Qwen).

Why was there a difference in the Step-by-Step Prompting effect between the closed and open models? Step-by-Step Prompting is designed to help the models reason by first eliciting useful properties of arguments and roles, before the final TFE. However, our analysis on a sample of random tuples revealed that open models are weak at inferring the essential properties that are most expected for an argument, to serve well in a specific semantic role. Input from this step, containing invalid or irrelevant properties, often derailed the final reasoning output quality. Moreover, if a reasoning step failed early on, it often derailed the rest of the chain (see Figure 2) which may explain the score losses in some of the experiments.

Why did GPT4-Turbo outperform GPT4.1 in spite of being older? We observed that GPT4.1's ratings were often more 'cautious', meaning it rarely output very high or very low scores. We veri-

[8]The available inter-annotator agreement or correlation information for our testsets indicates the upper bound for these tasks is between the mid-60's and mid-70's. GPT4-Turbo's scores got close, but no single setting was always the best.

Acceptable Sentence?	Qwen	GPT4-Turbo	Qwen filtering GPT4-Turbo
True Positive	18	52	26
True Negative	40	15	20
False Positive	0	**6**	1
False Negative	42	27	53

Table 6: Semantic filtering of "bad" generated sentences, random sample size: 100. "Bad" = True Negatives + False Positives. Left 2 columns: model both generated and filtered.

fied this by calculating the variance of each model's output for each task and experimental setting, finding a negative correlation between the variance and the correlation score (ρ). The lower variance of GPT4.1 means it is more likely to do worse on inputs with very high or very low thematic fit.

(Q2) Axis 2: Input Form In principle, providing more context enhances the models' ability to comprehend complex tasks. This was generally reflected in our experiments with open models, but not closed models (with few exceptions).

Why do generated sentences only help open models? Although the open models did not outperform the closed models, they often better leveraged the input with generated sentences, and improved the correlation score (ρ) relative to the tuple input. But sentence input is likely to outperform tuple input only if these sentences are "good" (semantically coherent, with the given argument in the given role for the given predicate). Moreover, "bad" sentences are likely to derail the models' performance. Therefore, we analyzed a random sample of 20 tuples, 5 generated sentences for each, altogether 100 sentences per model. We found that the open Qwen did better in filtering out "bad" sentences (in spite of having fewer parameters): its semantic filtering let through no False Positives (FPs), thus successfully leveraging the sentence inputs. In contrast, the closed GPT4-Turbo let through 6 FPs (Table 6). We saw similar results comparing filtering by Qwen *vs.* GPT4-Turbo, both on GPT4-Turbo's sentences.[9] With it, closed models clearly generated more "good" sentences (80 vs. 60 True Positives + False Negatives in our sample). Therefore, we caution against jumping to conclusions which model gained more relevant knowledge. See also another potential factor in §5.1.

(Q3) Axis 3: Output Form There are arguments in favor of either categorical and numeric output. LLMs are primarily designed for text-based tasks, so we expect predefined output categories to yield better results than numeric output. However, LLMs have also shown impressive performance in math (even if not perfect). Also, the ground-truth thematic fit scores were averaged across the participants, providing a graded scale, and therefore perhaps a numeric output would better correlate with that. Indeed, output form had almost no effect for closed models, but categorical output gave mostly worse results with open models. We hypothesize this may be due to the differences in post-training between the closed and open models, but we currently have no solid explanation for this difference, and we leave this for future work.

5.1 Performance drivers[10]

What is the main performance driver? **(a)** Model knowledge (model size and specific weights), **(b)** user-provided knowledge (system prompt context), or **(c)** generated sentences (input context) quality?

Model size (a) is a prime suspect, and is a confound with open vs. closed models (the latter being x200 larger). But Qwen outperforming GPT4-Turbo on filtering "bad" sentences is a counterexample. While we cannot fully interpret and compare model weights, we can explore (b) and (c).

System prompt knowledge We tested removing semantic role definitions from the system prompt for GPT4-Turbo (Table 7). As expected, GPT4-Turbo did worse on Fer-Ins when *Instrument* definitions were absent, but did surprisingly better on McRae when *Agent, Patient* and related information was removed (especially in Exp.3.1). It is doubly surprising, as adding role definitions during our preliminary studies (with the older GPT model) improved performance by 8-9 points in Exp.3.x. We suspect fundamental differences between the 2 GPTs are the cause, and wish these models were open. McRae output inspection reveals mixed effects: sometimes without role definitions GPT misinterpreted what properties the PropBank semantic role should have; other times including role definitions made GPT assign low score for an object lacking an optional property (e.g., low fit for 'rabbit' in *Patient/Arg1* role due to not being stationary).

Generated Sentence Quality Was the sentences' generation and/or filtering quality a major factor?

[9] Note that Qwen had higher False Negative rate, but this would only result in regressing to tuple input (§3.1 Axis 2).

[10] We thank the anonymous reviewers for motivating us to add this subsection. OpenAI retired *gpt-4-0125-preview* after we used it for the main results (Tables 2-6), so for all GPT runs here (Tables 7-8), including the baselines, we use *gpt-4-turbo-2024-04-09*, the closest available successor.

	Fer-Ins		McRae	
Exp.	r **(WR → WoR)**	Δr	r **(WR → WoR)**	Δr
3.1	.721 → .718	-.003	.621 → .649	**.028**
3.2	.718 → .709	-.009	.630 → .635	.005

Table 7: Ablation test: correlation (r) with and without role definition in system prompt. Exp. = experiment, WR = with roles, WoR = without roles, Δ = WoR-WR.

	GPT4-Turbo		Qwen	
Exp.	r **(Base → Good)**	Δr	r **(Base → Good)**	Δr
2.1	.422 → .427	.005	.439 → .444	.005
2.2	.509 → .505	-.004	.379 → .370	-.009

Table 8: Using "good" sentences: manually fixing false positives and false negatives in a random sample. Correlation (r) for GPT4-Turbo and Qwen on Fer-Ins, Δ = Good-Base.

To assess this, we reused the 100-sentence Fer-Ins sample (§5, Axis 2) and manually fixed false positives (by providing a compliant sentence) and false negatives (by using the wrongly filtered out sentence), for both GPT4-Turbo and Qwen (Table 8).

We did not see evidence for the sentences being a large factor: the effect of fixing the sample sentences was tiny (mostly $\pm$.005), even though the sample size was 8% of the dataset.

These small (and sometimes negative) effects made us look further: Recall that instead of every filtered sentence, we back off to using the score from the counterpart tuple condition. But if tuple vs. sentence input result in very different score distributions, this back-off can lead to unexpected results! Indeed, Figure 3 shows that human scores are almost uniform (somewhat heavier near the high end), lemma tuple input scores are bipolar around .1 and .8-.9, sentence input scores are almost all above .8, where manually correcting the random sample sentences had a small smoothing effect between .9-1.0. Recall also that higher TFE scores do not necessarily mean better correlation with the ground truth. These findings are interesting for themselves, and should guide future work.

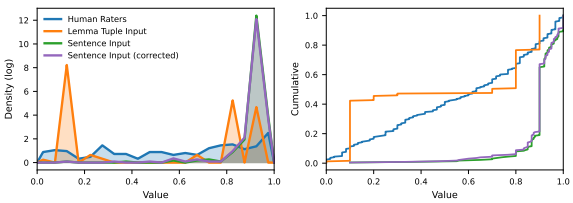

Figure 3: Fer-Ins TFE distributions of human raters (almost uniform), tuple input (bipolar), sentence input and corrected sentences (both right-heavy).

6 Conclusion and Future Work

We set out to discover if, or to what extent, autoregressive LLMs possess linguistic and other knowledge required to estimate the thematic fit of event participants. The tested LLMs set new state-of-the-art on each test set (and tied on Pado). Still, we conclude that TFE is not yet "solved". Closed models[11] yielded higher results than open models, indicating that closed models, especially GPT4-Turbo, acquired much of the linguistic knowledge required for TFE. This could be due to model size, training data and/or training regime. But open models were better at filtering "bad" generated sentences, and our back-off may have also greatly affected results, so we caution against rushing to interpret this as showing open models gained less relevant knowledge. No single axis setting consistently yielded the highest scores, so it is still unclear if models need additional knowledge or a different elicitation.

We tested three axes for eliciting this knowledge: **(1) Reasoning Form:** adding Step-by-Step Prompting, **(2) Input Form:** adding generated sentences as context, and **(3) Output Form:** predefined categories vs. numeric scores.

(1) Surprisingly, Step-by-Step Prompting only helped closed models; Our analysis showed that invalid or irrelevant reasoning not only was unhelpful but actually hurt performance. This happened more in open models.

(2) Generated sentences hurt closed models; We saw LLMs struggle with generating sentences where the given argument is a better fit to a semantic role different than the given one. To tackle this, we used novel semantic filtering, which, to our surprise, worked better for open models, because they were better at filtering out False Positives.

(3) Predefined categorical output (vs. numeric output) had zero-to-small effect with closed models and more negative effect with open models.

In future work, following our analysis, we hope to improve Step-by-Step Prompting, generate better sentences, better filter "bad" sentences, and better instruct the models to output more balanced estimation. If successful, LLMs may also be used to study linguistic issues in thematic fit, in line with Futrell and Mahowald (2025), BabyLM efforts[12], and others who believe that LLMs can be useful tools to study linguistic phenomena.

[11] We observed that GPT4.1 did worse than GPT4-Turbo, in spite of being newer, likely because 4.1 is too 'cautious' (§5).

[12] https://babylm.github.io

Limitations

Our experiments were evaluated on small datasets (a known limitation of this computational psycholinguistics subfield, as opposed to typical machine learning), so findings and conclusions should be assessed with caution. Our study considered only few LLMs, so one should be cautious about drawing conclusions for all LLMs.

LLMs can be very sensitive to prompt phrasing, so it is possible that different prompts may yield very different results. Also, prompt engineering is tricky; thus, tweaking the prompts may lead to different results. The richness of prompt variation (prompt engineering) depends on human resources (different people use language differently) and funding/time resources, all of which were limited. Future work may achieve better results with more resources, if available. Due to the scarcity of evaluation data, we could not use prompt optimization methods such as DSPy (Khattab et al., 2024) or fine-tune the models.

Last, all findings and conclusions here rely on data and annotations in English, as it is by far the most resource-rich. But thematic fit (and semantics in general) are thought of as universal, and of course relevant to all languages. Still, we would advise to be cautious of drawing universal conclusions before validating our findings on an additional, diverse set of languages.

Acknowledgments

The first author would also like to express gratitude to her sponsors, the Kuwait Chamber of Commerce and Industry and Abdullah Al Salem University, for granting her scholarships that made this research possible at Columbia University and Imperial College London.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. ArXiv preprint arXiv:2303.08774.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Klinton Bicknell, Jeffrey L Elman, Mary Hare, Ken McRae, and Marta Kutas. 2010. Effects of event knowledge in processing verbal arguments. *Journal of memory and language*, 63(4):489–505.

Emmanuele Chersoni, Enrico Santus, Ludovica Pannitto, Alessandro Lenci, Philippe Blache, and C-R Huang. 2019. A structured distributional model of sentence meaning and processing. *Natural Language Engineering*, 25(4):483–502.

Peter W. Culicover and Ray Jackendoff. 2005. *Simpler syntax*. Oxford Press.

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

David Dowty. 1991. Thematic proto-roles and argument selection. *language*, 67(3):547–619.

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

Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.

Todd R Ferretti, Ken McRae, and Andrea Hatherell. 2001. Integrating verbs, situation schemas, and thematic role concepts. *Journal of Memory and Language*, 44(4):516–547.

Charles J Fillmore. 1968. Lexical entries for verbs. *Foundations of language*, pages 373–393.

Richard Futrell and Kyle Mahowald. 2025. How linguistics learned to stop worrying and love the language models. *arXiv preprint arXiv:2501.17047*.

Clayton Greenberg, Asad Sayeed, and Vera Demberg. 2015. Improving unsupervised vector-space thematic fit evaluation via role-filler prototype clustering. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 21–31, Denver, Colorado. Association for Computational Linguistics.

Jeffrey Steven Gruber. 1965. *Studies in lexical relations.* Ph.D. thesis, Massachusetts Institute of Technology.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Xudong Hong, Asad Sayeed, and Vera Demberg. 2018. Learning distributed event representations with a multi-task approach. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 11–21.

Christopher R Johnson, Myriam Schwarzer-Petruck, Collin F Baker, Michael Ellsworth, Josef Ruppenhofer, and Charles J Fillmore. 2016. Framenet: Theory and practice. Technical report, International Computer Science Institute.

Carina Kauf, Emmanuele Chersoni, Alessandro Lenci, Evelina Fedorenko, and Anna Ivanova. 2024. Log probabilities are a reliable estimate of semantic plausibility in base and instruction-tuned language models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 263–277.

Carina Kauf, Anna A Ivanova, Giulia Rambelli, Emmanuele Chersoni, Jingyuan Selena She, Zawad Chowdhury, Evelina Fedorenko, and Alessandro Lenci. 2023. Event knowledge in large language models: the gap between the impossible and the unlikely. *Cognitive Science*, 47(11):e13386.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*.

Paul R Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*, pages 1989–1993.

Gyeong-Geon Lee, Ehsan Latif, Xuansheng Wu, Ninghao Liu, and Xiaoming Zhai. 2024. Applying large language models and chain-of-thought for automatic scoring. *Computers and Education: Artificial Intelligence*, 6:100213.

Joseph Patrick Levy, John Bullinaria, and Samantha McCormick. 2017. Semantic vector evaluation and human performance on a new vocabulary mcq test. In *Proceedings of the Annual Conference of the Cognitive Science Society: CogSci 2017 London:"Computational Foundations of Cognition"*. Cognitive Science Society.

Yuval Marton and Asad B. Sayeed. 2021. Thematic fit bits: Annotation quality and quantity interplay for event participant representation. In *International Conference on Language Resources and Evaluation*.

Ken McRae and Kazunaga Matsuki. 2009. People use their knowledge of common events to understand language, and do so as quickly as possible. *Language and linguistics compass*, 3(6):1417–1429.

Ken McRae, Michael J Spivey-Knowlton, and Michael K Tanenhaus. 1998. Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, 38(3):283–312.

AI Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI Blog. Retrieved December*, 20:2024.

Eleni Metheniti, Tim Van de Cruys, and Nabil Hathout. 2020. How relevant are selectional preferences for transformer-based language models? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1266–1278, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mughilan Muthupari, Samrat Halder, Asad Sayeed, and Yuval Marton. 2022. Where's the learning in representation learning for compositional semantics and the case of thematic fit. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 28–39. Association for Computational Linguistics.

Ollama Developers. 2023. Ollama: Run Large Language Models Locally. https://github.com/ollama/ollama. GitHub repository, Version 0.1.26 or later.

OpenAI. 2024. New embedding models and API updates. https://openai.com/index/new-embedding-models-and-api-updates/. Blog: 2025-10-30.

OpenAI. 2025. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1/. Blog: 2025-10-04.

Sebastian Padó, Ulrike Padó, and Katrin Erk. 2007. Flexible, corpus-based modelling of human plausibility judgements. In *Conference on Empirical Methods in Natural Language Processing*.

Ulrike Padó, Matthew W Crocker, and Frank Keller. 2006. Modelling semantic role plausibility in human sentence processing. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*, pages 1–8.

Terence Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press.

Paolo Pedinotti, Giulia Rambelli, Emmanuele Chersoni, Enrico Santus, Alessandro Lenci, and Philippe Blache. 2021. Did the cat drink the coffee? challenging transformers with generalized event knowledge. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 1–11, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. Technical report, OpenAI. Technical Report.

Philip Resnik. 1996. Selectional constraints: an information-theoretic model and its computational realization. *Cognition*, 61:127–159.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. ArXiv preprint arXiv:2308.12950.

Enrico Santus, Emmanuele Chersoni, Alessandro Lenci, and Philippe Blache. 2017. Measuring thematic fit with distributional feature overlap. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 648–658, Copenhagen, Denmark. Association for Computational Linguistics.

Asad Sayeed, Vera Demberg, and Pavel Shkadzko. 2015. An exploration of semantic features in an unsupervised thematic fit evaluation framework. *Italian Journal of Linguistics*, 27(1).

Charles Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

Davide Testa, Emmanuele Chersoni, and Alessandro Lenci. 2023. We understand elliptical sentences, and language models should too: A new dataset for studying ellipsis and its interaction with thematic fit. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3353.

Ottokar Tilk, Vera Demberg, Asad Sayeed, Dietrich Klakow, and Stefan Thater. 2016. Event participant modelling with neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 171–182.

Paolo Vassallo, Emmanuele Chersoni, Enrico Santus, Alessandro Lenci, Philippe Blache, et al. 2018. Event knowledge in sentence processing: A new dataset for the evaluation of argument typicality. In *Proceedings of LREC 2018 Workshop: Linguistic and Neuro-Cognitive Resources (LiNCR)*, pages 1–7. European Language Resources Association.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Annual Meeting of the Association for Computational Linguistics*.

Chaojie Wang, Yishi Xu, Zhong Peng, Chenxi Zhang, Bo Chen, Xinrun Wang, Lei Feng, and Bo An. 2023. keqing: knowledge-based question answering is a nature chain-of-thought mentor of llm. ArXiv preprint arXiv:2401.00426.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Baosong Yang, Zekun Wang, et al. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Appendices

A Prompt Design

Each experiment specified in §3.1 has a prompt design with specific input, output, and reasoning:

Simple Prompting with Lemma Tuples (Exp.1.1–Exp.1.2) A single prompt template ('simple thematic fit') that directly asks for the thematic fit score of a ⟨predicate, argument, role⟩ tuple (Fig. 1).

- `Given the predicate 'eat', how well does the argument 'pizza' fit the PropBank Arg1 role?`

We extend the prompt with two output settings:

- x.1: Numerical score where the model provides a numeric value between 0 and 1 to represent the thematic fit score:
  `Reply only with a valid JSON dictionary {"Score": numeric_value} where numeric_value is a float number from 0 to 1. Avoid adding any text outside this JSON dictionary.`
- x.2: Categorical score where the model selects one of the predefined labels that indicate the thematic fit score:
  `Reply only with a valid JSON dictionary {"Score": string_value} where string_value that is one of "Perfect", "High", "Medium", "Low" or "Unlikely/Impossible". Avoid adding any text outside this JSON dictionary.`

Sentence Generation and Semantic Filtering Before executing experiments 2 and 4, the model is asked to generate k diverse semantically coherent sentences containing an argument in a specific semantic role, given a predicate. Here we use $k = 5$.

- `Give me five sentences with the predicate 'eat' and argument 'pizza' in the PropBank Arg1 role. Make the sentences as diverse as possible, while using the predicate as the main verb and keeping the given argument in the specified role. Make sure that each sentence is semantically coherent and the argument is in the given role.`

LLMs may often generate a semantically incoherent or incompatible sentence with respect to the given role for the argument. Thus, we design

a semantic filtering process of 4 waterfall Yes/No Steps. If the sentence passed the whole waterfall at least once out of three trials, it is considered semantically acceptable. Here are the prompts we used:

- **(2a)** `Here's a sentence: 'I ate a pizza with my friends'. Is the given sentence semantically coherent?`
- **(2b)** `For the sentence: 'I ate a pizza with my friends'. Does the sentence contain the predicate 'eat'?`
- **(2c)** `For the sentence: 'I ate a pizza with my friends'. Does the sentence contain 'pizza' as an argument of the predicate 'eat'?`
- **(2d)** `For the same sentence: 'I ate a pizza with my friends'. Given the definitions, instructions, and answers above, is 'pizza' in the PropBank Arg1 role for the predicate 'eat'?`

Simple Prompting with Generated Sentences (Exp.2.1–Exp.2.2) We augment the prompt in Exp.1.x with sentences generated by the model from the same lemma tuples:

- `Given the following sentence, 'I ate a pizza with my friends', how well does the argument 'pizza' of the predicate 'eat' fit the PropBank Arg1 role?`

Step-by-Step Prompting with Lemma Tuples (Exp.3.1–Exp.3.2) Experiments 3.x are the enhanced version of Exp.1.x, where we break down the thematic fit task into a series of 4 prompts ('Step-by-Step Prompting'). See Fig. 1.

- `What salient essential properties or core characteristics best describe the essence of 'pizza'?`[13]
- `Given the predicate 'eat', and the argument of it in the PropBank Arg1 role, what salient essential properties or core characteristics are most expected for this argument, to serve well in the PropBank Arg1 role?`
- `Given the predicate 'eat', what PropBank roles are most suitable or likely for its argument 'pizza'?`
- `Given the predicate 'eat' with the argument 'pizza' in the PropBank Arg1 role, what essential properties does the argument have, properties which fit well its PropBank Arg1 role? And what essential properties this argument should have had (but are missing) in order for it to serve well in its PropBank Arg1 role?`

Step-by-Step Prompting with Generated Sentences (Exp.4.1–Exp.4.2) This experiment uses Step-by-Step Prompting as in Exp.3.x but with the addition of the generated sentence as in Exp.2.x:

- `Here's a sentence: 'I ate a pizza with my friends.', What salient essential properties or core characteristics best describe the essence of 'pizza'?`

B Earlier round of Experiment

A previous round of these experiments has been done on GPT4-Turbo and CodeLlama2 (codeLlama2-70B-instruct) (Roziere et al., 2023) with a previous version of the prompts we use. The results of the earlier round were mixed: sometimes higher (up to .77 on Fer-Loc) and other times lower than our new prompts. We take it as a positive sign that we didn't overfit the new prompts.

[13] In an early version of this experiment, only the semantic role name was mentioned. After analyzing it, we suspected that the LLMs may have not made the connection between roles such as '*Arg1*' and their PropBank sense in this context. Therefore we added the word 'PropBank' before the semantic role for Pado and McRae, as their semantic roles were PropBank-based, unlike the Ferretti datasets. This indeed improved the results on these sets by 2-3% (absolute).