EcomScriptBench: A Multi-task Benchmark for E-commerce Script Planning via Step-wise Intention-Driven Product Association

Weiqi Wang[*♠♣], **Limeng Cui**[♣], **Xin Liu**[♣], **Sreyashi Nag**[♣], **Wenju Xu**[♣], **Chen Luo**[♣], **Sheikh Muhammad Sarwar**[♣], **Yang Li**[♣], **Hansu Gu**[♣], **Hui Liu**[♣], **Changlong Yu**[♣], **Jiaxin Bai**[♠], **Yifan Gao**[♣], **Haiyang Zhang**[♣], **Qi He**[♣], **Shuiwang Ji**[†♡♣], **Yangqiu Song**[†♠♣]

[♠]Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China

[♣]Amazon.com Inc, Palo Alto, CA, USA

[♡]Department of Computer Science & Engineering, Texas A&M University, Texas, USA

{wwangbw, yqsong}@cse.ust.hk, {culimeng, xliucr, cheluo}@amazon.com

Abstract

Goal-oriented script planning, or the ability to plan coherent sequences of actions toward specific goals, is commonly used by humans to plan for daily activities. In e-commerce, customers increasingly seek LLM-based assistants to plan for them with a script and recommend products at each step, thereby facilitating convenient and efficient shopping experiences. However, this capability remains underexplored due to several challenges, including the inability of LLMs to simultaneously conduct script planning and product retrieval, difficulties in matching products caused by semantic discrepancies between planned actions and search queries, and a lack of methods and benchmark data for evaluation. In this paper, we step forward by formally defining the task of E-commerce Script Planning (EcomScript) as three sequential subtasks. We propose a novel framework that enables the scalable generation of product-enriched scripts by associating products with each step based on the semantic similarity between the actions and their purchase intentions. By applying our framework to real-world e-commerce data, we construct the very first large-scale EcomScript dataset, EcomScriptBench, which includes 605,229 scripts sourced from 2.4 million products. Human annotations are then conducted to provide gold labels for a sampled subset, forming an evaluation benchmark. Extensive experiments reveal that current (L)LMs face significant challenges with EcomScript tasks, even after fine-tuning, while injecting product purchase intentions improves their performance.

Figure 1: An example of a ***product-enriched script*** for planning the objective of *plan an autumn-themed party with friends and family*, with relevant products associated with some steps. Note that for simpler steps, such as the first two, no products are needed.

1 Introduction

In our daily lives, humans commonly plan a sequence of general prototypical actions, usually in the form of step-by-step instructions, to achieve a specific objective (Abbott et al., 1985). This capability, also known as ***goal-oriented script planning*** (Bower et al., 1979; Schank and Abelson, 1975), forms the foundation of situationally grounded planning for complex scenarios, which is crucial for intelligent agents. With recent advances in Large Language Models (LLMs; OpenAI, 2024b,a; Dubey et al., 2024), recent works (Yuan et al., 2023; Wang et al., 2023a; Chan et al., 2024a; Deng et al., 2024) have demonstrated the strong capabilities of LLMs in script planning. This has led to the development of LLM-based script planners across downstream domains (Hao et al., 2023; Hazra et al., 2024).

[*]Work done during his internship at Amazon.com Inc.

[†]Visiting academic scholar at Amazon.com Inc.

In e-commerce, there is a growing trend of customers querying LLM-empowered shopping assistants to create scripts tailored to their specific needs or objectives, with each step featuring relevant products. For example, as illustrated in Figure 1, LLM assistant is expected to generate an eight-step script toward the user's objective, *plan an autumn-themed party with friends and family*, while also associating products that customers may find useful for achieving each step, such as items related to food and decoration in this example. However, for simple actions that can be completed without additional items, no products are recommended. A goal-oriented script with product recommendations at certain steps that may necessitate product purchases, which we term as **product-enriched script**, facilitates a convenient shopping experience, saves customers from multiple rounds of searches, and ultimately fosters new concierge-style applications for LLM shopping assistants to promote business.

Despite its significant potential, the exploration of this ability has faced several challenges. First, while current LLMs excel in script planning, they struggle to retrieve relevant products from the vast pool in e-commerce platforms. Although some LLMs have been pre-trained with e-commerce product knowledge, prior studies (Li et al., 2024; Peng et al., 2024) indicate that they still face difficulties in generating precise product titles that accurately align with specific items in the pool for further retrieval. Additionally, while a generate-then-retrieve approach–where LLMs first plan the script and then use the steps as corresponding search queries for product searches using traditional search engines–might be feasible, there is often a semantic gap between the planned steps (the actions users should take sequentially) and the search queries intended for search engines (see Section 3.3). This gap arises because queries typically include product features and descriptions provided by users, which are matched against product metadata. When users search for actions that a product can facilitate, this discrepancy compromises the matching mechanism, further undermining the effectiveness of search engines and limiting their ability to address the shortcomings of LLMs in both planning and product retrieval. Finally, there is a notable lack of methods and benchmark datasets that incorporate both script plans and products at the step level, which are necessary to evaluate the current capabilities of LLMs in this area.

To bridge these gaps, in this paper, we formally define the process of **E-commerce Script Planning** (EcomScript) as a three-step discriminative process consisting of three sequential sub-tasks. We then introduce a novel framework for automatically guiding LLMs in generating product-enriched scripts by incorporating product keyword filtering at each step to narrow the search scope for relevant products. To address semantic discrepancies, we search for products based on their *purchase intentions*, which represent a customer's underlying motivation to buy the product (Chang and Wildt, 1994), and then filter out those whose intentions don't align closely with the action at each step.

By applying our framework to Amazon Review data (Hou et al., 2024), we construct a large-scale knowledge base, EcomScriptBench, that includes 605,229 product-enriched scripts derived from real user purchase reviews, alongside 2.4 million products, each linked to ten distinct purchase intentions. Within each script, we associate up to three products with each step by applying our intention alignment strategy (§ 4.3). Human annotations are then conducted to provide gold labels for 15,000 randomly sampled entries across three subtasks, thereby constructing an evaluation benchmark. We then experiment with over 20 (L)LMs, applying both fine-tuning and advanced prompting techniques to EcomScript tasks. Our findings reveal that all LMs encounter significant challenges in addressing these tasks. Further analysis identifies potential reasons for their underperformance and demonstrates that injecting purchase intentional knowledge significantly enhances LLMs' performance.

2 Related Works

2.1 Goal-oriented Script Planning

Goal-oriented scripts refer to a coherent and appropriate sequence of steps, usually in the form of actions, as instructions for achieving a goal (Regneri et al., 2010). They are a common reflection of language planning capabilities, often observed in embodied AI (Gan et al., 2022) and robotics (Zhang et al., 2024a). In the era of LLMs, various works have explored their script planning capabilities. Yuan et al. (2023) proposed an over-generate-then-filter framework to improve the constraint language planning capabilities of LLMs and distilled a knowledge base from it. Sun et al. (2023); Wang et al. (2023a) attempted generative script learning in a multimodal manner to enhance the planning

abilities of large vision-language models. Joshi et al. (2023) designed an interactive text-based gaming framework that consists of daily real-world human activities as another benchmark. Nevertheless, none of the prior works have explored script planning in the context of e-commerce, which holds significant potential for customers wishing to plan toward their desired objectives and purchase necessary products at every step all at once.

2.2 Purchase Intention Understanding

Purchase intention represents the implicit mental state that motivates customers' purchase behaviors (Anscombe, 2000), which simulates the underlying reasons a customer wishes to achieve with the purchase of a product (Chan et al., 2024b). Various existing works have already examined the impact of consumer shopping intentions on downstream applications (Dai et al., 2006; Zhang et al., 2016; Hao et al., 2022; Lu et al., 2024). Specifically, Ni et al. (2019) collected real-world customer reviews to investigate the underlying purchase intentions in consumer purchase behavior and created a large-scale review dataset based on Amazon. Yu et al. (2023, 2024); Bai et al. (2024) then leveraged this data and proposed a semi-supervised intention generation framework to obtain purchase intentions at scale (FolkScope and COSMO) by distilling OPT (Zhang et al., 2022). Xu et al. (2024) further strengthened this approach by incorporating visual signals from product images to guide the generation of more feature-oriented purchase intentions that align with stronger human preferences. Ding et al. (2024) transformed FolkScope into an evaluation benchmark and demonstrated that LLMs cannot effectively utilize intention for product recommendation. In our work, we share a similar aspiration of using purchase intention as the key to match products that best align with each actionable step in every script, enabling LLMs to implicitly leverage intention for product retrieval.

3 Problem Definition

3.1 EcomScript Task Definitions

We first introduce our definition of the proposed e-commerce script planning tasks (EcomScript). Since both asking an LLM to generate a script with products associated with each step and evaluating such generations are difficult to accomplish directly, it is challenging to formulate the task simply as a one-step generative task and evaluate it in an open-ended manner. To this end, we propose three sequential discriminative tasks to emulate the process, with the aspiration that an LLM can perform these three tasks to build a generate-then-discriminate paradigm that fully enables automated e-commerce script planning. Initially, the model is given a user objective o, a script consisting of k steps toward this objective $S_o = \{s_1, s_2, ..., s_k\}$ (collected from the user or generated by the LLM), and a pool of n e-commerce products $P = \{p_1, p_2, ..., p_n\}$.

Task 1: Script Verification: The first task asks the model to determine the plausibility and feasibility of the script based on the given objective. It gives the model o and S_o as input and requires the model to output a binary score $T_1(o, S_o) \in \{0, 1\}$ as the indicator where 1 indicates that the script is plausible and feasible, and 0 indicates otherwise.

Task 2: Step-Product Discrimination: The second task aims to determine whether a step in the script requires the purchase of a product to assist the user in accomplishing that step. If so, the model will then be provided with a product and asked to determine whether purchasing this product can help with the step. Formally, the model takes o, s_i, and p_i as inputs and is required to output a binary score $T_2(o, s_i, p_i) \in \{0, 1\}$, where 0 indicates that the step does not require any product purchase or that the product cannot help, and 1 indicates that the product is a good match to contribute to the step.

Task 3: Script-Products Verification: The final task aims to determine the overall feasibility of a product-enriched script by providing the model with the objective, the script, and products associated with each step. Formally, the model takes o, $S_o = \{s_1, s_2, ..., s_k\}$, and the products at each step $P_{s_1}, P_{s_2}, ..., P_{s_k}$ as inputs and is expected to output a binary score $T_3(o, S_o, (P_{s_1}, P_{s_2}, ..., P_{s_k})) \in \{0, 1\}$ where 0 indicates that there are internal conflicts between different steps and products, while 1 means that all products are suitable for each step and can collaborate within the entire script.

The rationale behind this task design is that, with the filtering models associated with these three tasks and an LLM as the core shopping assistant, we can automate the process of e-commerce script planning. This is achieved by first asking the LLM to generate a script based on the user's provided objective. Then, the **script verifier** (T_1) can determine the plausibility of the script and guide the LLM to improve it if necessary. Products will be retrieved according to our proposed step-intention alignment strategy, as explained later in Section 4.3.

The **step-product discriminator** (T_2) can verify the results of each retrieved product associated with each step and remove unnecessary products for simple steps. Finally, the **script-products verifier** (T_3) will check the product-enriched script and ensure that all products can coordinate smoothly within the script to be recommended to the customer.

3.2 Datasets

To ensure the practicality of applying our framework to real-world scenarios, we collect real world products from purchases made at Amazon.com. To manage the overwhelming size of the product pool P and reduce product redundancy, we randomly sample 10% of unique products from each category while maintaining the original distribution. As a result, 2.4 million products and 3.7 million associated reviews are used for constructing EcomScriptBench.

3.3 Semantic Gaps in Product Retrieval

Prior to this work, we conducted a preliminary pilot study to investigate the effectiveness of using search engines—considered a traditional alternative method—for retrieving products based on user-provided steps in the context of e-commerce script planning. In this study, we selected 200 scripts at random and used their individual steps as search queries. The results showed that roughly 68% of these queries returned only a limited assortment of products, indicating that search engines struggled to align product titles and metadata with the nuanced, natural language requirements expressed in the user queries. We also observe that most of retrieved products are identical or very similar to each other, limiting the divergence of product association results.

For example, when searching for "a reusable bottle that is easy to clean and suitable for carrying both hot and cold beverages," generic listings of reusable bottles were returned, with little emphasis on the specific attributes mentioned. This illustrates a semantic gap between how users describe their needs and the structured metadata currently used to index and retrieve products. Addressing this gap will require incorporating richer contextual signals and better capturing user intent. By enhancing the information associated with products—such as their features, use cases, and suitability—systems can more effectively match user needs with relevant product recommendations. This supports the main motivation of our study, which is to compensate for the weaknesses of semantic retrieval.

4 EcomScriptBench Construction

In this section, we introduce our method for synthesizing product-enriched scripts to construct an evaluation benchmark. An overview of the framework is shown in Figure 2. Specifically, our framework consists of four main stages: (1) user objective and script collection, (2) product purchase intention mining, (3) script-product association through step-intention alignment, and (4) human annotation.

To enable scalable data collection, we use GPT-4o-mini (OpenAI, 2024a), a powerful yet cost-efficient proprietary LLM, as the generator to collect user objectives, scripts, and product purchase intentions. Following Brown et al. (2020) and Wang et al. (2024a), we guide each generation stage with a few-shot prompt as described below:

```
<TASK-PROMPT>
<INPUT_1><OUTPUT_(1,1)> ... <OUTPUT_(1,N_1)>
<INPUT_2><OUTPUT_(2,1)> ... <OUTPUT_(2,N_2)>
...
<INPUT_5><OUTPUT_(5,1)> ... <OUTPUT_(5,N_5)>
<INPUT_6>
```

where we modify **<TASK-PROMPT>** at each stage to provide different instructions that inform the LLM of the generation objective and incorporate five <INPUT$_i$> and <OUTPUT$_i$> pairs as few-shot exemplars for demonstration (prompts in Appendix A.1).

4.1 User Objective and Script Collection

We start by collecting user objectives by instructing the LLM to extract and infer them from user purchase reviews, as these are most practically aligned with real-world use cases. To achieve this, we let **<TASK-PROMPT>** clarify the goal to the LLM, which involves generating an objective that the customer is trying to achieve based on a series of purchases and their reviews. Note that we explicitly ask the LLM to avoid generating overly simplistic objectives and to aim for complex ones that require multiple steps to complete, in order to facilitate further script planning. We then populate <INPUT$_i$> and <OUTPUT$_i$> with five pairs of user purchase reviews and a list of comma-separated objectives inferred from the reviews by experts. The LLM is then expected to infer a list of user objectives from the last given customer purchase review (<INPUT$_6$>). If the LLM believes that no objective can be inferred, "None" will be generated instead. To ensure high

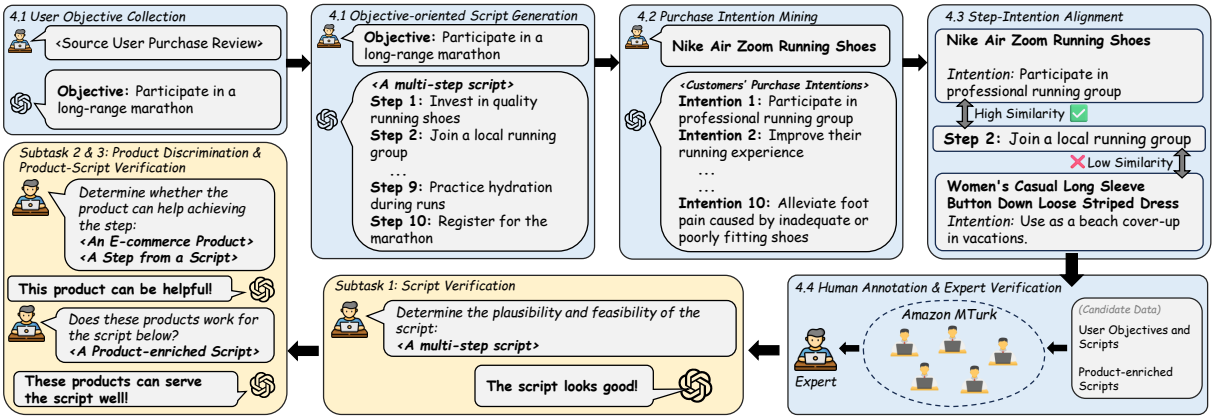

Figure 2: An overview of our benchmark curation and evaluation pipeline for EcomScriptBench.

quality, we discard reviews that are too short or contain excessive punctuation or hashtags.

With these objectives, we further instruct the LLM to generate goal-oriented scripts based on them. We similarly modify the prompt to achieve this by changing **<TASK-PROMPT>** to instruct the LLM to devise a coherent and sequential plan of steps, in the form of a script, with all steps being actions that are commonly seen in usual cases. Specifically, we require the LLM to avoid generating overly simple actions and to maximize the necessity of purchases in each step by generating actions that may require items to complete, whenever possible. We then populate $\texttt{<INPUT}_i\texttt{>}$ and $\texttt{<OUTPUT}_i\texttt{>}$ with five pairs of user objectives and their corresponding actionable scripts, written by experts, as exemplars. The LLM is then expected to generate the script for the last given user objective ($\texttt{<INPUT}_6\texttt{>}$). For simplicity, we ask the LLM to generate scripts that contain no more than 10 steps, and longer scripts will be truncated to a maximum of 10 steps. The mined objectives and generated scripts will be candidate data for the first subtask.

4.2 Purchase Intention Mining

We then collect purchase intentions for e-commerce products, aiming to leverage these intentions as a key to bridge products and actionable steps. The rationale behind this approach is that intentions commonly reflect what customers wish to achieve with their purchases, which intuitively aligns with the semantics of actionable steps. This alignment helps overcome the semantic discrepancy found in traditional search queries, which typically focus on product features and metadata.

To collect purchase intentions, we follow Yu et al. (2023) and utilize LLMs to distill intentional knowledge. Specifically, we modify **<TASK-PROMPT>** to instruct the LLM to infer purchase intentions by reasoning about the customer's motivations and desires. We emphasize modeling the customer's mental state, using phrases like "PersonX wants to buy this because" or "PersonX believes buying this can" to guide the generation. We then populate $\texttt{<INPUT}_i\texttt{>}$ and $\texttt{<OUTPUT}_i\texttt{>}$ with five pairs of purchased product metadata and expert-drafted customer intentions as examples. The model is then asked to generate purchase intentions for the last given product ($\texttt{<INPUT}_6\texttt{>}$). For each product, we collect 10 intentions, resulting in a total of 24 million intentions for 2.4 million products.

4.3 Step-Intention Alignment

In this stage, we first ask the LLM to determine whether a product purchase is necessary for each step, in order to filter out trivial actions that can be performed directly by the user without additional support from any product, such as "invite friends" and "check the calendar." If the LLM believes that additional product purchases are necessary, we further ask it to generate a list of keywords to describe the product as thoroughly as possible. We will then filter products that contain any of these keywords to narrow down our search scope.

To achieve this, we modify the **<TASK-PROMPT>** to include the descriptions above and populate $\texttt{<INPUT}_i\texttt{>}$ and $\texttt{<OUTPUT}_i\texttt{>}$ with five pairs of actionable steps in a script, along with their associated purchase necessity and relevant product keywords. The model will then infer the purchase necessity and relevant product keywords for the last provided step in the script ($\texttt{<INPUT}_6\texttt{>}$).

For every step deemed necessary for product purchases and their filtered products, we use Sentence-

Type	#Data (Unlabeled)	#Token	Expert.
Scripts	605,229	71.5	94.0%
Steps	5,928,271	7.48	94.0%
w. products	3,018,276	6.98	-
w.o. products	2,909,995	7.98	-
Products	2,401,087	19.31	-
Intentions	24,010,870	10.27	98.5%
Task 1	5,000 (592,729)	-	95.5%
Task 2	5,000 (5,919,278)	-	96.5%
Task 3	5,000 (589,801)	-	97.0%

Table 1: Statistics of the EcomScriptBench benchmark. #Token refers to average number of tokens used. Expert. refers to expert acceptance rate.

BERT (Reimers and Gurevych, 2019) to calculate the average embedding similarity between each step and the purchase intentions of each product. For each step, we rank all filtered products to select the top three that best align with the actionable step in the script, using a lower-bound similarity threshold of $\tau = 0.45$ to control for relevance, which is determined based on our observation of the similarity distribution. We limit our selection to a maximum of three products to reduce overlap and maintain a manageable dataset size. Each step and its selected products form the candidate data for the second task, while the entire script and all retrieved products are used for the third subtask.

4.4 Human Annotations

Benchmark Annotation: We finally conduct human annotations via Amazon Mechanical Turk (AMT) to provide gold labels for a sampled proportion of data and build them into an evaluation benchmark. For each task, 5,000 data entries are randomly sampled for annotation. We qualify 56 (18.67%) workers from a pool of 300 candidates with excellent annotation records and provide them with detailed instructions to complete each subtask. They are then tasked with annotating (1) the plausibility and feasibility of a given script towards an objective as generated in §4.1, (2) the necessity of purchasing a given product for a specific step in the script, as collected in §4.3, and (3) the overall feasibility of a product-enriched script given the user objective, the entire script, and all retrieved products. Note that only scripts that passed the plausibility annotation are used as candidate data in further tasks. We collect five votes for each entry, and the majority vote is used as the final label. The overall inter-annotator agreement (IAA) is 78% in terms of pairwise agreement, and the Fleiss Kappa (Fleiss, 1971) is 0.53, indicating sufficient agreement. More details are in Appendix B.

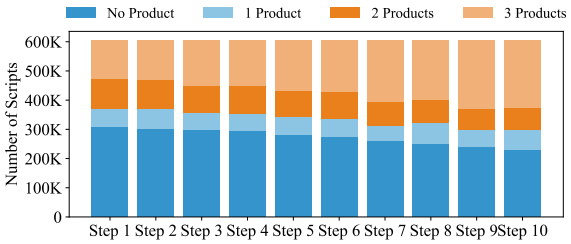

Figure 3: Distribution of the number of retrieved products at each step in EcomScriptBench.

Expert Verification: To verify the quality of our collected labels, we invite three additional experts in e-commerce NLP to perform an extra round of annotation verification. Each expert is asked to annotate a sample of 200 data entries for each task, following the same instructions provided to the AMT annotators. Results in Table 1 show that, on average, 96.33% of the labels collected from AMT annotations align with the experts' majority vote, demonstrating the reliability of our collected labels.

5 Experiments and Analyses

5.1 EcomScriptBench Statistics

We first present the statistics of EcomScriptBench in Table 1. In total, we collect 605K scripts with 5.9 million steps. Of a sample of 200 scripts, 94% were annotated as plausibly correct by expert annotators. Among them, approximately 3 million steps are deemed necessary for product purchases by the LLM, while the others do not necessitate products. We also collect 24 million intentions based on 2.4 million products, of which 98.5% from a sample of 200 are deemed plausible by expert annotators, demonstrating the high quality of our dataset. For each task, we collect labels for 5,000 sampled data entries and leave the rest unlabeled. We partition the annotated data into train, dev, and test sets according to an 8:1:1 ratio. Each task follows a high expert acceptance rate, demonstrating the reliability of EcomScriptBench. We further visualize the product distributions of our collected scripts by steps in Figure 3. We observe that as the script progresses (the number of steps increases), more steps are required to associate with products, indicating the need for e-commerce script planning in real-world scenarios.

Methods	Backbone	Script Verification			Product Discrimination			Product-Script Veri.		
		Acc	AUC	Ma-F1	Acc	AUC	Ma-F1	Acc	AUC	Ma-F1
Random	N/A	50.00	-	50.00	50.00	-	50.00	50.00	-	50.00
Majority	N/A	60.98	-	60.05	57.67	-	57.10	56.46	-	56.24
PTLM *(Zero-shot)*	RoBERTa-Large *340M*	52.04	51.79	51.21	50.80	50.74	50.68	51.39	51.37	51.32
	DeBERTa-Large *435M*	51.98	52.06	51.82	52.00	51.96	51.23	52.34	52.59	51.81
	CAR *435M*	52.77	52.75	51.95	51.98	52.10	51.88	53.06	53.25	52.90
	CANDLE *435M*	53.76	53.61	53.20	52.89	53.10	52.28	52.40	52.37	51.91
	VERA-xl *3B*	53.63	53.50	53.18	52.94	52.87	52.21	52.18	52.09	51.94
	VERA-xxl *11B*	55.77	55.66	54.79	54.49	54.61	53.92	54.90	54.94	54.34
LLM *(Zero-shot)*	Meta-Llama-3-8B	70.05	-	69.98	64.83	-	64.36	61.16	-	60.22
	Meta-Llama-3-70B	71.74	-	71.52	66.02	-	65.05	62.00	-	61.33
	Meta-Llama-3.1-8B	71.45	-	71.30	65.74	-	65.69	61.63	-	60.96
	Meta-Llama-3.1-70B	72.65	-	72.42	66.15	-	65.54	62.50	-	62.22
	Meta-Llama-3.1-405B	75.26	-	74.97	68.16	-	67.33	65.66	-	65.65
	Gemma-2-2B	66.82	-	66.80	60.56	-	60.22	58.95	-	58.10
	Gemma-2-9B	71.27	-	70.98	65.14	-	64.15	61.07	-	60.40
	Gemma-2-27B	71.77	-	71.27	66.86	-	66.20	63.15	-	62.70
	Phi-3.5-mini *4B*	68.18	-	68.05	61.92	-	61.15	60.36	-	59.79
	Falcon2 *11B*	71.73	-	71.68	65.70	-	65.12	61.89	-	61.65
	Mistral-7B-v0.3	72.38	-	71.49	66.42	-	65.77	62.18	-	61.47
	Mistral-Nemo *12B*	73.18	-	72.51	66.98	-	66.78	62.95	-	62.71
	Mixtral-8x7B-v0.1	75.06	-	74.25	66.39	-	65.59	63.64	-	62.84
PTLM & LLM *(Fine-tuned)*	RoBERTa-Large *340M*	79.18	79.27	78.86	72.26	72.32	71.74	70.26	70.38	69.83
	DeBERTa-v3-Large *435M*	81.10	80.76	81.03	74.26	74.56	73.78	72.00	71.93	71.99
	Meta-LLaMa-3-8B	83.48	83.38	82.64	75.75	75.52	**75.73**	73.06	73.33	72.84
	Meta-LLaMa-3.1-8B	85.24	85.07	84.64	**76.44**	**76.51**	75.53	**74.48**	**74.44**	**74.38**
	Gemma-2-2B	81.06	80.95	80.82	73.43	73.51	73.09	69.61	69.79	68.78
	Gemma-2-9B	82.04	82.20	81.35	73.58	73.94	73.15	71.65	71.41	71.44
	Mistral-7B-v0.3	**85.72**	**85.61**	**85.51**	75.63	75.61	75.33	73.18	73.09	72.62
LLM *(API)*	GPT4o-mini	74.30	-	73.54	69.03	-	68.47	69.68	-	69.16
	GPT4o-mini (5-shots)	74.56	-	73.61	71.56	-	71.09	71.39	-	71.04
	GPT4o-mini (COT)	71.66	-	71.59	69.31	-	68.63	70.62	-	70.23
	GPT4o-mini (SC-COT)	72.74	-	72.38	71.13	-	70.79	70.93	-	70.26
	GPT4o-mini (SR)	73.32	-	72.35	72.46	-	71.89	71.08	-	70.43
	GPT4o	77.50	-	77.23	73.04	-	72.06	71.50	-	71.33
	GPT4o (5-shots)	77.92	-	76.93	73.90	-	73.68	72.85	-	72.83
	GPT4o (COT)	74.89	-	74.12	71.05	-	70.58	70.32	-	69.68
	GPT4o (SC-COT)	73.84	-	73.16	71.08	-	70.67	69.26	-	68.67
	GPT4o (SR)	76.22	-	76.13	71.97	-	71.28	71.90	-	70.96

Table 2: Evaluation results (%) of various (L)LMs on the annotated testing sets of EcomScriptBench. The best performances within each method are underlined and the best among all methods are **bold-faced**.

5.2 Benchmarking Experiments

Setup: We experiment with a selection of (L)LMs to investigate their performance on our proposed tasks. Each task, as defined in §3.1, is evaluated as a binary classification task using accuracy, AUC, and Macro-F1 scores as evaluation metrics. The evaluation of different models is categorized into three types: **(1) Zero-shot:** We first evaluate several (L)LMs in a zero-shot manner on the full annotated testing set. For small-sized Pre-Trained Language Models (PTLMs), we assess RoBERTa (Liu et al., 2019), DeBERTa-v3 (He et al., 2023), CAR (Wang et al., 2023b), CANDLE (Wang et al., 2024a), and VERA (Liu et al., 2023) following the zero-shot question answering evaluation paradigm (Ma et al., 2021). For LLMs, we evaluate Llama3, Llama3.1 (Touvron et al., 2023; Dubey et al., 2024), Gemma2 (Mesnard et al., 2024; Riviere et al., 2024), Phi3.5 (Abdin et al., 2024), Falcon2 (Malartic et al., 2024), Mistral (Jiang et al., 2023), and Mixtral (Jiang et al., 2024) via direct zero-shot prompting (Qin et al., 2023). **(2) Finetuning:** Next, we assess the performance of LLMs when fine-tuned on EcomScriptBench. We fine-tune RoBERTa, DeBERTa, Llama3, Llama3.1, Gemma2, Falcon2, and Mistral and evaluate them on the partitioned testing set. LLMs are fine-tuned using LoRA (Hu et al., 2022). **(3) LLM API:** Finally, we evaluate the performance of GPT4o (OpenAI, 2023, 2024b) and GPT4o-mini (OpenAI, 2024a), which represent proprietary LLMs, using zero-shot, few-shot (Brown et al., 2020), Chain-of-Thought (COT; Wei et al., 2022), Self-Consistent COT (SC-

COT; Wang et al., 2023d), and Self-Reflection (SR; Shinn et al., 2023) promptings on the full annotated testing set. We also include Random and Majority voting to illustrate the characteristics of our benchmark. See more details in Appendix A.3.

Results: The evaluation results are presented in Table 2. Our observations include: **(1) Challenges with EcomScript Tasks:** (L)LMs struggle with all tasks in e-commerce script planning, particularly in those involving e-commerce products. All models achieved only moderately satisfactory performance across the three subtasks. For instance, the best open-source LLM, LLAMA-3.1-405B, attained accuracy scores of 75%, 68%, and 65% on the respective tasks. This underscores the inherent difficulty of the EcomScriptBench. Notably, the latter two subtasks are considerably more challenging than script verification, likely due to the complexities associated with e-commerce products and the requisite product knowledge. **(2) Impact of Fine-tuning and Advanced Prompting:** While fine-tuning and advanced prompting methods yield some performance improvements, there remains significant room for enhancement. We observed a notable boost in performance when LLMs are fine-tuned on annotated product-enriched scripts. For example, the performance of LLAMA-3.1-8B improved by 12%, 11%, and 13% across the three tasks, respectively. Similarly, GPT series models benefited from advanced prompting techniques, such as few-shot prompting and self-reflection. COT prompting, on the other hand, cannot help, which may be due to its reliance on the model's internal reasoning paths rather than incorporating additional external product-related signals or domain-specific annotations that align closely with the given tasks. **(3) Effects of Model Training Paradigms and Scale:** Enhancing the training paradigm and increasing the number of parameters positively impacted performance. In the LLAMA series, both increasing parameters and updating training data and methods led to improved results. The performance trend associated with increasing the number of parameters is also clear and highlights the significance of model scale in achieving better outcomes on our tasks. **(4) Complexity of Tasks:** The poor performances on both the step-product discrimination and script-product verification tasks demonstrate that EcomScript is a complex and challenging problem for LLMs, revealing the limitations of current models in flexibly integrating e-commerce product knowledge into planning tasks. Greater efforts should be directed along this direction in order to achieve automated e-commerce script planning in a single-step generative manner.

5.3 The Effect of Injecting Intentions

From the evaluation results in Table 2, we observe that a key weakness in current LLMs is their difficulty in associating products with each step in a script and verifying whether the entire script can work. To improve this, we hypothesize that injecting intentional knowledge into LLMs may help, as it provides a better understanding of what e-commerce products can help or how they can assist the customer, thereby promoting the linking of products with script planning. To achieve this, we select two intention knowledge bases based on products from Amazon, FolkScope (Yu et al., 2023), and MIND (Xu et al., 2024) as sources of intentions. We use a natural language prompt to concatenate product metadata (title, features, descriptions) as the input with their purchase intentions as the output, and train LLMs under a generative objective using LoRA (Hu et al., 2022). They are then sequentially fine-tuned on training set of EcomScriptBench. Another group of LLMs, after fine-tuning on FolkScope and MIND, is directly evaluated for comparison. All models are evaluated on the testing set of EcomScriptBench, and the results are reported in Table 3. From the results, we observe a significant improvement across all tasks when the models are sequentially fine-tuned on FolkScope and MIND, then on EcomScriptBench, compared to being solely fine-tuned on either one. This indicates that aligning LLMs with more e-commerce products' use cases or purchase motivations enhances their ability to identify useful products for users' desired actions or steps in scripts. Since intentions from both resources are distilled from LLMs, this opens up a scalable yet cost-efficient paradigm for improving LLMs' performance on e-commerce script planning tasks.

5.4 Error Analysis of GPT-Series Models

Finally, for a more fine-grained error analysis, we manually inspect the causes of errors in 200 sampled COT responses generated by GPT-4o across all tasks and categorize their mistakes into three categories: **(1) Wrong understanding of products:** 68% of errors are caused by the LLM's false understanding of a specific usage or feature of a product

Backbone	Training Data	Script Verification			Product Discrimination			Product-Script Veri.		
		Acc	AUC	Ma-F1	Acc	AUC	Ma-F1	Acc	AUC	Ma-F1
Llama-3.1 ***8B***	Zero-shot	71.45	-	71.30	65.74	-	65.69	61.63	-	60.96
	ECOMSCRIPTBENCH	83.86	83.94	83.05	77.70	77.87	77.59	75.88	75.58	75.58
	FolkScope + MIND	67.74	67.63	67.38	66.79	66.43	66.11	64.91	64.87	64.42
	+ ECOMSCRIPTBENCH	84.65	84.84	84.13	78.60	78.83	78.27	76.35	76.50	76.08
Mistral-v0.3 ***7B***	Zero-shot	72.38	-	71.49	66.42	-	65.77	62.18	-	61.47
	ECOMSCRIPTBENCH	85.72	85.61	85.51	75.63	75.61	75.33	73.18	73.09	72.62
	FolkScope + MIND	69.77	70.00	69.56	67.78	67.75	67.39	63.70	63.41	63.66
	+ ECOMSCRIPTBENCH	**85.87**	**85.80**	**86.37**	**81.18**	**80.96**	**80.54**	**78.94**	**78.94**	**78.66**

Table 3: Evaluation results (%) of transfering knowledge from FolkScope and MIND to aid ECOMSCRIPTBENCH. The best performances among each method is underlined and best ones among all methods are **bold-faced**.

that conflicts with the steps or the entire script. For example, when a step requires controlling the user's non-compatible smart light bulbs using the virtual assistant, the LLM might incorrectly suggest voice commands that only work for compatible devices. To address this issue, multi-modal product images or more detailed attributes can be incorporated. **(2) Conflict in reasoning across steps:** 27% of errors occur due to the model's failure to reason about the feasibility of collaborating products associated with different steps, where the model may mistakenly deem it infeasible to purchase two products simultaneously. **(3) Internal conflict and annotation errors:** 5% of errors are due to internal conflicts, such as inconsistencies between the binary predictions made and the corresponding reasoning rationales, as well as annotation errors, potentially caused by overzealous annotators.

5.5 Category-wise Performance Analysis

We then conduct a detailed analysis of GPT-4o's performance in the product discrimination task across a variety of product categories. Table 4 presents the accuracy scores obtained for each major product category. We observe that GPT-4o performs best in categories like "Toys and Games," "Patio Lawn and Garden," "Grocery and Gourmet Food," and "Cell Phones and Accessories," often surpassing 80% accuracy. These categories tend to have clearer, more distinct product descriptors, making it easier to distinguish between items. In contrast, performance dips in more ambiguous or heterogeneous categories like "Beauty and Personal Care" and "Health and Household." Products in these domains often share overlapping descriptors or subtle differences (e.g., similar lotions or vitamins), making text-only differentiation challenging. Intermediate results in categories like "Electronics and Office Products" suggest that while technical specifications are helpful, the sheer diversity of items can still obscure product distinctions. Integrating additional modalities, such as images or structured product metadata, might help address these difficulties and improve the model's discrimination capabilities across a broader range of categories.

Category	Accuracy
Automotive	64.58
Beauty and Personal Care	63.95
Cell Phones and Accessories	82.31
Clothing Shoes and Jewelry	78.99
Electronics	66.15
Health and Household	62.08
Home and Kitchen	65.63
Grocery and Gourmet Food	82.49
Industrial and Scientific	79.51
Office Products	67.42
Patio Lawn and Garden	82.84
Sports and Outdoors	76.68
Tools and Home Improvement	65.57
Toys and Games	84.37

Table 4: Accuracy (%) of GPT-4o on product discrimination task by product categories.

6 Conclusions

In conclusion, this paper proposes the task of e-commerce script planning and introduces a novel framework for collecting product-enriched scripts. By applying the framework to Amazon product data, we construct a sibling large-scale knowledge base and build the very first evaluation benchmark upon it. Extensive experiments demonstrate the challenges of our task and potential solutions to improve the performance of LLMs on ECOMSCRIPT. We hope that our task and benchmark can serve as an important cornerstone to advance the e-commerce shopping experience by creating more intelligent and personalized shopping assistants with e-commerce script planning capability that ultimately benefit the community and the world.

Limitations

We discuss three main limitations of our work.

First, our data construction process relies significantly on GPT-4o-mini, a proprietary LLM, for data collection, as well as human annotation for label collection and verification. This raises concerns about the reproducibility and the high costs associated with our dataset. However, the expense of using GPT-4o-mini is relatively low compared to other proprietary LLMs; for instance, we spent only around $250 USD to collect 24 million intentions. The quality of the output remains outstanding, with fast generation speeds that effectively simulate a real-world LLM-powered shopping assistant. We also experimented with using LLAMA-3.1-405B as the core generator for data collection, which also yields exceptional data quality. However, hosting the model and using it for inference proved to be computationally and time-intensive, leading us to ultimately choose GPT-4o-mini.

Next, we assign the verification of product compatibility between different steps to a human-annotated task and do not implement any strategies within our data collection framework. This decision is made because detecting conflicting products is a complicated task that requires consideration of many features, some of which cannot be determined solely based on product metadata. We leave this verification to future industrial efforts to ensure that products retrieved at different stages can accommodate each other and collectively contribute to successful execution.

Finally, we defer the exploration of practical solutions to assist LLMs in solving ECOMSCRIPT, as well as the deployment of these solutions to deliver real-world benefits, to future work. We can also implement knowledge editing techniques to address this, as done by Lau et al. (2024); Zhang et al. (2024b). In the long run, we envision a model capable of accurately understanding a customer's needs and recommend all products at once via e-commerce script planning can promote purchase decision-making and increase e-commerce revenue.

Ethics Statement

Since our dataset curation pipeline involves prompting LLMs, it is important to implement stringent measures to ensure the absence of offensive content in both the prompts and the generated responses. We first explicitly state in the prompt that the LLM should not generate any content that contains personal privacy violations, promotes violence, racial discrimination, hate speech, sexual content, or self-harm. Then, we manually inspect a random sample of 500 data entries from all tasks in ECOMSCRIPTBENCH for offensive content. Based on our observations, we have not detected any offensive content. Therefore, we believe that our dataset is safe and will not yield any negative societal impact.

Due to data privacy issues, our dataset will not be made public. As for language models, we access all open-source LMs via the Hugging Face Hub (Wolf et al., 2020). The number of parameters is presented in Table 2. All associated licenses permit user access for research purposes, and we have agreed to follow all terms of use.

We conduct large-scale human annotations on the Amazon Mechanical Turk (AMT) platform. We invite annotation workers from the US, Europe, and India due to their proficiency in English. The annotators are paid an average hourly rate of $17.50, which is comparable to the minimum wage in their local jurisdictions. The selection of these annotators is solely based on their performance on the evaluation set, and we do not collect any personal information about the participants from AMT. The expert annotators agree to participate as their contribution to the paper without compensation.

Acknowledgements

We thank the anonymous reviewers and the area chair for their constructive comments. The authors of this paper were supported by the ITSP Platform Research Project (ITS/189/23FP) from the ITC of Hong Kong, SAR, China, as well as the AoE (AoE/E-601/24-N), the RIF (R6021-20), and the GRF (16205322) from the RGC of Hong Kong, SAR, China. We also thank the Amazon Search Experience Science team for supporting this intern project.

References

Valerie Abbott, John B Black, and Edward E Smith. 1985. The representation of scripts in memory. *Journal of memory and language*, 24(2):179–199.

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav

Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219.

GEM Anscombe. 2000. *Intention*. Harvard University Press.

Jiaxin Bai, Zhaobo Wang, Junfei Cheng, Dan Yu, Zerui Huang, Weiqi Wang, Xin Liu, Chen Luo, Qi He, Yanming Zhu, Bo Li, and Yangqiu Song. 2024. Intention knowledge graph construction for user intention relation modeling. *CoRR*, abs/2412.11500.

Gordon H Bower, John B Black, and Terrence J Turner. 1979. Scripts in memory for text. *Cognitive psychology*, 11(2):177–220.

Tom B. Brown, 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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chunkit Chan, Cheng Jiayang, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2024a. Exploring the potential of chatgpt on sentence level relations: A focus on temporal, causal, and discourse relations. In *Findings of the Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, March 17-22, 2024*, pages 684–721. Association for Computational Linguistics.

Chunkit Chan, Cheng Jiayang, Yauwai Yim, Zheye Deng, Wei Fan, Haoran Li, Xin Liu, Hongming Zhang, Weiqi Wang, and Yangqiu Song. 2024b. Negotiationtom: A benchmark for stress-testing machine theory of mind on negotiation surrounding. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 4211–4241. Association for Computational Linguistics.

Tung-Zong Chang and Albert R Wildt. 1994. Price, product information, and purchase intention: An empirical study. *Journal of the Academy of Marketing science*, 22:16–27.

Honghua (Kathy) Dai, Lingzhi Zhao, Zaiqing Nie, Ji-Rong Wen, Lee Wang, and Ying Li. 2006. Detecting online commercial intention (OCI). In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 829–837. ACM.

Zheye Deng, Chunkit Chan, Weiqi Wang, Yuxi Sun, Wei Fan, Tianshi Zheng, Yauwai Yim, and Yangqiu Song. 2024. Text-tuple-table: Towards information integration in text-to-table generation via global tuple extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 9300–9322. Association for Computational Linguistics.

Wenxuan Ding, Weiqi Wang, Sze Heng Douglas Kwok, Minghao Liu, Tianqing Fang, Jiaxin Bai, Xin Liu, Changlong Yu, Zheng Li, Chen Luo, Qingyu Yin, Bing Yin, Junxian He, and Yangqiu Song. 2024. Intentionqa: A benchmark for evaluating purchase intention comprehension abilities of language models in e-commerce. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 2247–2266. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon

Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, and Kevin Stone. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Tianqing Fang, Weiqi Wang, Sehyun Choi, Shibo Hao, Hongming Zhang, Yangqiu Song, and Bin He. 2021a. Benchmarking commonsense knowledge base population with an effective evaluation dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8949–8964. Association for Computational Linguistics.

Tianqing Fang, Hongming Zhang, Weiqi Wang, Yangqiu Song, and Bin He. 2021b. DISCOS: bridging the gap between discourse knowledge and commonsense knowledge. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2648–2659. ACM / IW3C2.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Chuang Gan, Siyuan Zhou, Jeremy Schwartz, Seth Alter, Abhishek Bhandwaldar, Dan Gutfreund, Daniel L. K. Yamins, James J. DiCarlo, Josh H. McDermott, Antonio Torralba, and Joshua B. Tenenbaum. 2022. The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied AI. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pages 8847–8854. IEEE.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8154–8173. Association for Computational Linguistics.

Zhenyun Hao, Jianing Hao, Zhaohui Peng, Senzhang Wang, Philip S. Yu, Xue Wang, and Jian Wang. 2022. Dy-hien: Dynamic evolution based deep hierarchical intention network for membership prediction. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 363–371. ACM.

Rishi Hazra, Pedro Zuidberg Dos Martires, and Luc De Raedt. 2024. Saycanpay: Heuristic planning with large language models using learnable domain knowledge. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 20123–20133. AAAI Press.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian J. McAuley. 2024. Bridging language and items for retrieval and recommendation. *CoRR*, abs/2403.03952.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6384–6392. AAAI Press.

Albert Q. Jiang, 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. *CoRR*, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Abhinav Joshi, Areeb Ahmad, Umang Pandey, and Ashutosh Modi. 2023. Scriptworld: Text based environment for learning procedural knowledge. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 5095–5103. ijcai.org.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kelvin J. L. Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. 2024. Learning to generate explainable stock predictions using self-reflective large language models. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 4304–4315. ACM.

Ching Ming Samuel Lau, Weiqi Wang, Haochen Shi, Baixuan Xu, Jiaxin Bai, and Yangqiu Song. 2024. Ecomedit: An automated e-commerce knowledge editing framework for enhanced product and purchase intention understanding. *CoRR*, abs/2410.14276.

Yangning Li, Shirong Ma, Xiaobin Wang, Shen Huang, Chengyue Jiang, Haitao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2024. Ecomgpt: Instruction-tuning large language models with chain-of-task tasks for e-commerce. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18582–18590. AAAI Press.

Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. 2023. Vera: A general-purpose plausibility estimation model for commonsense statements. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1264–1287. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Feihong Lu, Weiqi Wang, Yangyifei Luo, Ziqin Zhu, Qingyun Sun, Baixuan Xu, Haochen Shi, Shiqi Gao, Qian Li, Yangqiu Song, and Jianxin Li. 2024. Miko: Multimodal intention knowledge distillation from large language models for social-media commonsense discovery. In *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 3303–3312. ACM.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13507–13515. AAAI Press.

Quentin Malartic, Nilabhra Roy Chowdhury, Ruxandra Cojocaru, Mugariya Farooq, Giulia Campesan, Yasser Abdelaziz Dahou Djilali, Sanath Narayan, Ankit Singh, Maksim Velikanov, Basma El Amel Boussaha, Mohammed Al-Yafeai, Hamza Alobeidli, Leen Al Qadi, Mohamed El Amine Seddik, Kirill Fedyanin, Réda Alami, and Hakim Hacid. 2024. Falcon2-11b technical report. *CoRR*, abs/2407.14885.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Cristian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, and et al. 2024. Gemma: Open models based on gemini research and technology. *CoRR*, abs/2403.08295.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 188–197. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence. *OpenAI*.

OpenAI. 2024b. Hello gpt-4o. *OpenAI*.

Bo Peng, Xinyi Ling, Ziru Chen, Huan Sun, and Xia Ning. 2024. ecellm: Generalizing large language models for e-commerce from large-scale, high-quality instruction data. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? In *Proceedings of the 2023*

Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 1339–1384. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. 2010. Learning script knowledge with web experiments. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 979–988. The Association for Computer Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3027–3035. AAAI Press.

Roger C Schank and Robert P Abelson. 1975. Scripts, plans, and knowledge. In *IJCAI*, volume 75, pages 151–157. New York.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Chenkai Sun, Tie Xu, Chengxiang Zhai, and Heng Ji. 2023. Incorporating task-specific concept knowledge into script learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 3018–3032. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Qingyun Wang, Manling Li, Hou Pong Chan, Lifu Huang, Julia Hockenmaier, Girish Chowdhary, and Heng Ji. 2023a. Multimedia generative script learning for task planning. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 986–1008. Association for Computational Linguistics.

Weiqi Wang, Tianqing Fang, Wenxuan Ding, Baixuan Xu, Xin Liu, Yangqiu Song, and Antoine Bosselut. 2023b. CAR: conceptualization-augmented reasoner for zero-shot commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 13520–13545. Association for Computational Linguistics.

Weiqi Wang, Tianqing Fang, Chunyang Li, Haochen Shi, Wenxuan Ding, Baixuan Xu, Zhaowei Wang, Jiaxin Bai, Xin Liu, Cheng Jiayang, Chunkit Chan, and Yangqiu Song. 2024a. CANDLE: iterative conceptualization and instantiation distillation from large language models for commonsense reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2351–2374. Association for Computational Linguistics.

Weiqi Wang, Tianqing Fang, Haochen Shi, Baixuan Xu, Wenxuan Ding, Liyu Zhang, Wei Fan, Jiaxin Bai, Haoran Li, Xin Liu, and Yangqiu Song. 2024b. On the role of entity and event level conceptualization in generalizable reasoning: A survey of tasks, methods, applications, and future directions. *CoRR*, abs/2406.10885.

Weiqi Wang, Tianqing Fang, Baixuan Xu, Chun Yi Louis Bo, Yangqiu Song, and Lei Chen. 2023c. CAT: A contextualized conceptualization and instantiation framework for commonsense reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13111–13140. Association for Computational Linguistics.

Weiqi Wang and Yangqiu Song. 2024. MARS: benchmarking the metaphysical reasoning abilities of language models with a multi-task evaluation dataset. *CoRR*, abs/2406.02106.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023d. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yutong Wang, Jiali Zeng, Xuebo Liu, Fandong Meng, Jie Zhou, and Min Zhang. 2024c. Taste: Teaching large language models to translate through self-reflection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6144–6158. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Baixuan Xu, Weiqi Wang, Haochen Shi, Wenxuan Ding, Huihao Jing, Tianqing Fang, Jiaxin Bai, Xin Liu, Changlong Yu, Zheng Li, Chen Luo, Qingyu Yin, Bing Yin, Long Chen, and Yangqiu Song. 2024. MIND: multimodal shopping intention distillation from large vision-language models for e-commerce purchase understanding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 7800–7815. Association for Computational Linguistics.

Changlong Yu, Xin Liu, Jefferson Maia, Yang Li, Tianyu Cao, Yifan Gao, Yangqiu Song, Rahul Goutam, Haiyang Zhang, Bing Yin, and Zheng Li.

2024. COSMO: A large-scale e-commerce common sense knowledge generation and serving system at amazon. In *Companion of the 2024 International Conference on Management of Data, SIGMOD/PODS 2024, Santiago AA, Chile, June 9-15, 2024*, pages 148–160. ACM.

Changlong Yu, Weiqi Wang, Xin Liu, Jiaxin Bai, Yangqiu Song, Zheng Li, Yifan Gao, Tianyu Cao, and Bing Yin. 2023. Folkscope: Intention knowledge graph construction for e-commerce commonsense discovery. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1173–1191. Association for Computational Linguistics.

Siyu Yuan, Jiangjie Chen, Ziquan Fu, Xuyang Ge, Soham Shah, Charles Robert Jankowski, Yanghua Xiao, and Deqing Yang. 2023. Distilling script knowledge from large language models for constrained language planning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4303–4325. Association for Computational Linguistics.

Chenwei Zhang, Wei Fan, Nan Du, and Philip S. Yu. 2016. Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 1373–1384. ACM.

Jiatao Zhang, Lanling Tang, Yufan Song, Qiwei Meng, Haofu Qian, Jun Shao, Wei Song, Shiqiang Zhu, and Jason Gu. 2024a. FLTRNN: faithful long-horizon task planning for robotics with large language models. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 6680–6686. IEEE.

Liyu Zhang, Weiqi Wang, Tianqing Fang, and Yangqiu Song. 2024b. Conke: Conceptualization-augmented knowledge editing in large language models for commonsense reasoning. *CoRR*, abs/2412.11418.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Yaowei Zheng, Richong Zhang, Junhao Zhang, YeYanhan YeYanhan, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Appendices

A Implementation Details

A.1 Dataset Construction Prompts

We first present the prompts used in each step to sequentially instruct GPT-4o-mini to generate candidate data for EcomScriptBench. Special tokens, such as <example>, </example>, are added whenever necessary throughout the prompts.

A.1.1 User Objective and Script Collection

To collect real-world user objectives and their associated scripts, we use the following prompt to instruct the LLM.

> **Given a product and an user review about it, infer some potential action goals of purchasing the product. The goals should be what the user wants to do with the product. They do not have to be explicitly stated in the review, but can be reasoned from the context. You may think of big goals of what the user wants to achieve with the help of the product. Action goals should be specific and actionable objectives that take multiple steps to achieve, and the product may contribute to one step of them. Do not generate goals that are too simple. Do not generate buying a product again, recommend the product or brand to others, reliable customer service, etc. They shouldn't be very long-term goals, do not generate being successful or making a lot of money. Separate each goal with || and make each goal specific by describing it in detail. Follow these examples:**
>
> ...
>
> **Product <i>: Nike Air Zoom Running Shoes**
>
> **Review <i>: This is the best pair of running shoes I've ever owned. They are comfortable and provide great support.**
>
> **Goals: participate in a marathon || stay a healthy lifestyle || start running regularly**
>
> ...
>
> **Product <N>: Samsung 65-Inch 4K Smart TV**
>
> **Review <N>: This TV has great picture quality and sound. It's perfect for**

watching movies and shows.

We drop reviews that are less than 10 tokens or contain fewer than 3 unique tokens. Additionally, we exclude reviews with more than 5 hashtags, as these are sometimes misleading. These thresholds were determined based on our prior experience in processing e-commerce reviews and have proven to provide the best trade-off in retaining the maximum number of valid reviews. When reviews are of poor quality or unavailable, our offline experiments show that LLMs can still infer potential user objectives from the product title and metadata alone. This ensures that the framework remains functional and capable of reasoning about product use cases, even without relying on user reviews.

We then use the following prompt to generate a goal-oriented script based on the objectives collected above. A brief explanation of each step is required for clarification.

> **Given an actionable goal, generate a script of steps that can be used to achieve the goal. The script should be detailed and specific, and each step should be actionable and achievable sequentially. Limit the script to within 10 steps and each step to within 20 words. For each step, a short explanation of the step should be provided. The steps should be in the correct logical and temporal order and should be detailed enough to be executed sequentially by someone who is not familiar with the goal. For each step, try to ensure that some E-commerce products can be purchased to help achieve the step.**
>
> ...
>
> Objective <i>: Participate in a marathon.
>
> Step 1: Choose a marathon to participate in (Research and select a marathon that fits your schedule and location preference)
>
> Step 2: Register for the marathon (Complete the registration form and pay any associated fees)
>
> Step 3: Create a training plan (Develop a schedule with incremental mileage increases and rest days)
>
> Step 4: Purchase proper running gear (Buy running shoes, moisture-wicking clothing, and a water bottle)
>
> Step 5: Start your training program (Follow your schedule, gradually increasing your running distance each week)
>
> Step 6: Maintain a balanced diet (Eat a mix of carbohydrates, proteins, and fats to fuel your training)
>
> Step 7: Stay hydrated (Drink plenty of water daily and during your runs)
>
> Step 8: Practice long runs (Include one long run per week to build endurance, following your training plan)
>
> Step 9: Get enough rest (Ensure adequate sleep and recovery time to avoid overtraining and injuries)
>
> Step 10: Plan race day logistics (Prepare your transportation, know the race course, and plan post-race recovery)
>
> ...
>
> Objective <N>: Learn to play the guitar.

For scripts that are more than 10, we drop the outlier steps. In total, we recorded only 3,098 cases where truncation occurs, which is very rare. In most cases, LLM follows our instruction precisely.

A.1.2 Purchase Intention Mining

We then distill product purchase intentions from the LLM by following Yu et al. (2023, 2024) with the prompt below. This type of intention distillation has been proven effective and can support downstream applications. Thus, our knowledge distillation-based method is justifiable and enables large-scale benchmark construction.

> **Given a product retrieved from Amazon, you are required to generate 10 possible intentions that a user may have that motivates them to purchase the product. The intention should be describing what the user wants to do with the product, believe the product can help them achieve, or the problem the product can solve. It should be specific and not too general. For example, "like the product", "wants to buy it", "good product" are invalid intentions. Best intentions describe the user's goal, desire, or action to be taken with the product. Generate the intentions**

directly with one intention per line. Follow these examples:

...

Product <i>: Nike Air Zoom Running Shoes

Intention 1: Improve their running performance with better cushioning and support.

Intention 2: Train for a marathon or other long-distance race.

Intention 3: Increase comfort during daily jogging sessions.

Intention 4: Reduce the risk of injuries by using shoes with advanced technology.

Intention 5: Enhance their athletic appearance with stylish and modern footwear.

Intention 6: Replace worn-out running shoes with a high-quality, durable option.

Intention 7: Experience the benefits of lightweight shoes for faster running times.

Intention 8: Participate in a running club or group with appropriate gear.

Intention 9: Transition to a more serious and dedicated running routine.

Intention 10: Alleviate foot pain caused by inadequate or poorly fitting shoes.

...

Product <N>: chouyatou Women's Casual Long Sleeve Button Down Loose Striped Cotton Maxi Shirt Dress

Using intention as the connecting link is inherently more effective than traditional search queries, such as keywords in product titles and metadata, which are often not represented in script steps. To best align intentions with steps, we create exemplars with similar semantics and grammatical structures to effectively guide GPT-4o-mini in generating steps and intentions with consistent linguistic patterns (e.g., omitting the subject, using the simple present tense, and keeping them short and concise). However, gaps between intentions and steps can still occur. To address this, we generate 10 intentions per product to ensure as much coverage as possible. In industrial applications, even more intentions per product could be generated to enhance coverage and improve alignment further, given the low cost of generating outputs with GPT-4o-mini. In our current dataset construction pipeline, expert evaluations and human annotations confirm that our method is effective and does not significantly impact final performance. However, verifying its efficacy at an industrial scale is left to future work by the e-commerce community.

A.1.3 Step-Intention Alignment

Finally, we prompt the LLM again to determine whether a product purchase is necessary for each step in the script. For steps that are deemed necessary, we ask the LLM to generate a list of keywords to help us narrow down the search scope of products and proceed with our intention alignment strategy. We use the following prompt to assess the purchase necessity of each step:

> Given a plan consisting of ten steps, determine whether any additional item or product can be helpful in each step to make it successful. Note that it can be anything or any product that is helpful in terms of achieving the step. There are steps that definitely do not require additional help from other things or items, such as "inviting a friend", "going to somewhere", "select a time", "search for a specific information". They are usually actions that can be done directly by the person and do not require additional assistance from a product. There are also steps that can be assisted by having other products, such as "prepare food", "clean the house", "write a letter", "make a phone call", "prepare entertainment". They usually involve interactions with some tools, materials, or other things to complete the action, or can be done mor easily or efficiently with the help of them. Given the steps below, first provide a yes or no answer to whether it is helpful to purchase a product to achieve the step. Then, provide a short list of product keywords that represent items that can be helpful in achieving the step. These keywords can be general to represent more items. You are forced to follow the example format in generating the answer, which is first generate a one word answer, either "yes" or "no", then generate a

list of keywords. Generate one line per step. For example:

...

Step 1: Get measuring cups and spoons (Purchase a set with common baking measurements to accurately measure ingredients)

Step 2: Get a food scale (Weigh ingredients like meat and flour for reliable portioning)

Step 3: Use a thermometer (Monitor oil and internal temperatures for frying and roasting)

Step 4: Time activities (Use a timer to track marinating, baking, etc. for consistency)

Step 5: Watch tutorial videos (View cooking demos to learn proper knife skills and techniques)

Step 6: Take an in-person cooking class (Learn from a professional chef for hands-on experience)

Step 7: Practice fundamental recipes (Master basic recipes to handle ingredients and temperatures)

Step 8: Focus on one technique (Work on skills like sautéing, searing, or deglazing)

Step 9: Invest in high-quality cookware (Buy pans that distribute heat evenly for optimal cooking)

Step 10: Follow recipes precisely (Carefully measure and time each step before improvising)

yes (measuring cups, spoons, measurement, ingredients)

yes (food scale, scale, weight)

yes (thermometer, oil, temperature)

yes (timer, activities, marinating)

yes (tutorial videos, cooking demos, knife skills)

no

yes (fundamental recipes, basic recipes, recipes)

no

yes (high-quality cookware, pans, heat)

yes (recipes, measure, time)

...

Step 1: Choose a marathon to participate in (Research and select a marathon that fits your schedule and location preference)

Step 2: Register for the marathon (Complete the registration form and pay any associated fees)

Step 3: Create a training plan (Develop a schedule with incremental mileage increases and rest days)

Step 4: Purchase proper running gear (Buy running shoes, moisture-wicking clothing, and a water bottle)

Step 5: Start your training program (Follow your schedule, gradually increasing your running distance each week)

Step 6: Maintain a balanced diet (Eat a mix of carbohydrates, proteins, and fats to fuel your training)

Step 7: Stay hydrated (Drink plenty of water daily and during your runs)

Step 8: Practice long runs (Include one long run per week to build endurance, following your training plan)

Step 9: Get enough rest (Ensure adequate sleep and recovery time to avoid overtraining and injuries)

Step 10: Plan race day logistics (Prepare your transportation, know the race course, and plan post-race recovery)

For SentenceBERT, we use T5-xxl (11B; Raffel et al., 2020) as the backbone. We begin by calculating the embeddings of all purchase intentions for a product and all steps separately, then compute the semantic similarity between every pair of intention and step using cosine similarity as the metric. Each product's purchase relatedness is determined by the average similarity of all pairs relevant to the product for a specific step. To ensure that only related products are selected, we set a lower bound threshold of $\tau = 0.4$, which filters out approximately 95% of products at each step, improving the data quality.

Analyzing the distribution of embedding similarity, we find that 13% of intentions have a similarity score higher than 0.5 with the user objective. If we were to eliminate cases where intentions closely match objectives, the effectiveness of product retrieval would likely decrease. This is because many product recommendations rely on the semantic alignment between intentions and specific steps in a user's script. Removing these closely matched cases could lead to gaps in relevant product as-

sociations, resulting in less accurate or relevant recommendations. Therefore, we argue that non-script-level intentions—those not directly tied to a specific step—also play a crucial role in improving product retrieval and the overall user experience.

A.2 Evaluation Prompts

To evaluate LLMs on three tasks in ECOMSCRIPTBENCH, we present our evaluation prompts in a zero-shot scenario in Table5. These prompts are consistently used across all model evaluations to ensure a fair comparison.

For few-shot evaluations, examples are added after the task descriptions and before the prompted test entry. The exemplars are randomly sampled for each test entry from a set of 20 expert-annotated examples.

For Chain of Thought (COT) prompting, we specifically instruct LLMs to "think step by step and generate a short rationale to support your reasoning." We then ask them to provide an answer based on the generated rationale. The sampling temperature, τ, is set to 0.1 by default, and 5 COT responses are sampled with τ set to 0.7 in the SC-COT setting. In the SC-COT setting, we also explicitly include another round of conversation to allow the LLM to verify whether the prediction is correct according to its generated rationale.

For self-reflection, we follow previous approaches (Wang et al., 2024c; Koa et al., 2024) and construct similar prompts to evaluate the LLM.

A.3 Evaluation Implementations

To evaluate PTLMs in a zero-shot manner, we adopt the evaluation pipeline used for zero-shot question answering (Ma et al., 2021; Wang et al., 2023b,c, 2024b). Specifically, for each task, we convert the question into two declarative statements, which serve as natural language assertions corresponding to 'yes" or "no" options. For instance, when determining whether a product is necessary for a step, we generate two assertions: "The product <PRODUCT> is helpful to the step <STEP>," and "The product <PRODUCT> is not helpful to the step <STEP>." The models are then tasked with computing the loss of each assertion. The assertion with the lowest loss is considered as the model's prediction. This approach allows any PTLM to be evaluated under classification tasks with an arbitrary number of options or even type classification based on a single assertion. We use the open code library[1] as our code base and follow the default hyperparameter settings. For VERA, we follow the exact same implementation[2] (Liu et al., 2023). The accessed backbone models are `liujch1998/vera-xl` (3B) and `liujch1998/vera` (11B), and all other hyperparameter settings follow the default setting.

For evaluating LLMs in a zero-shot manner, we transform the input for each task into assertions using natural language prompts, as explained in Appendix A.2 and Table 5. The models are then prompted to determine the plausibility of the provided assertions by answering yes or no questions. We parse their responses using pre-defined rules to derive binary predictions. When generating each token, we consider the top 10 tokens with the highest probabilities. Their generation process is limited to 10 tokens for computational efficiency.

For fine-tuning LLMs, we use LoRA for fine-tuning, and the LoRA rank and α are set to 16 and 32, respectively. We adopt the open code library from LlamaFactory[3] (Zheng et al., 2024) for model training and evaluation. We similarly use an Adam (Kingma and Ba, 2015) optimizer with a learning rate of 5e-5 and a batch size of 8. The maximum sequence length for the tokenizer is set at 300. All models are fine-tuned over three epochs and the last checkpoint is evaluated. We use three random seeds and report the average performance for all experiments.

Finally, for evaluating proprietary LLMs, such as GPT-4o and GPT-4o-mini, we similarly prompt them as with open LLMs. Detailed prompts are explained in Appendix A.2.

B Annotation Details

B.1 Worker Selection Protocol

To ensure the high quality of our human annotation, we implement strict quality control measures. Initially, we invite only those workers to participate in our qualification rounds who meet the following criteria: 1) a minimum of 2,000 HITs approved, and 2) an approval rate of at least 90%. We select workers separately for each task and conduct three qualification rounds per task to identify those with satisfactory performance. In each qualification round, we create a qualification test suite that includes both easy and challenging questions, each

[1] https://github.com/Mayer123/HyKAS-CSKG
[2] https://github.com/liujch1998/vera
[3] https://github.com/hiyouga/LLaMA-Factory

Task	Prompt
SV.	You are given an objective **<TEST-ENTRY-OBJECTIVE>** and a script **<TEST-ENTRY-SCRIPT>**. Your task is to assess the plausibility and feasibility of the script in relation to the objective. First, evaluate the plausibility by determining if the script logically aligns with the objective. Next, consider the feasibility by assessing whether the script is realistic and achievable given the constraints of the objective. Based on your evaluation, please output a binary answer "yes" or "no". "yes" indicates that the script is both plausible and feasible. "no" indicates that the script is either implausible or infeasible. Please answer with one word "yes" or "no":
PD.	You are given an objective **<TEST-ENTRY-OBJECTIVE>**, a specific action **<TEST-ENTRY-STEP>**, and a product **<TEST-ENTRY-PRODUCTT>**. Your task is to determine whether the step requires the purchase of a product to assist the user in accomplishing that step. First, assess if the step **<TEST-ENTRY-STEP>** necessitates any product purchase. If it does, evaluate whether purchasing the product **<TEST-ENTRY-PRODUCT>** can effectively help with the step. Based on your evaluation, please output a binary answer "yes" or "no". "yes" indicates that the product is a good match and can contribute to the step. "no" indicates that the step does not require any product purchase or that the product cannot help. Please answer with one word "yes" or "no":
SPV.	You are given an objective **<TEST-ENTRY-OBJECTIVE>**, a script consisting of multiple steps **<TEST-ENTRY-SCRIPT>**, and the products associated with each step **<TEST-ENTRY-PRODUCT-ENRICHED-SCRIPT>**. Your task is to determine the overall feasibility of the product-enriched script. Evaluate whether any internal conflicts exist between the different steps and their associated products. If all products are suitable for their respective steps and can collaborate effectively within the entire script. Based on your evaluation, please output a binary answer "yes" or "no". "yes" indicates that all products are appropriate and can work together seamlessly. "no" indicates that there are internal conflicts among the steps and products. Please answer with one word "yes" or "no":

Table 5: Evaluation prompts used for benchmarking LLMs' performances across three tasks in ECOMSCRIPTBENCH: SV, PD, and SPV refer to: Script Verification, Product Discrimination, and Script-Product Verification.

with a gold label from the authors. Workers are required to complete a minimum of 40 questions. To qualify, they must achieve an accuracy rate of at least 75% on the qualification test. After our selection process, we chose 56 workers from a pool of 300 candidates as our benchmark annotators. On average, our worker selection rate stands at 18.67%. Following the qualification rounds, workers are required to complete another instruction round. This round contains complex questions selected by the authors, and workers are required to briefly explain the answer to each question. The authors will then double-check the explanations provided by the annotators and disqualify those with a poor understanding.

B.2 Annotation Instructions

For each task, we provide workers with comprehensive task explanations in layman's terms to enhance their understanding. We also offer detailed definitions and several examples of each choice to help annotators understand how to make decisions. These definitions largely align with our task definitions, as explained in Section 3.1. Each entry requires the worker to annotate using a four-point Likert scale. Workers are asked to rate each given script using such scale, where 1 signifies strong agreement and 4 indicates strong disagreement. We consider annotations with a value of 1 or 2 as plausible and those with a value of 3 or 4 as implausible.

To ensure comprehension, we require annotators to confirm that they have thoroughly read the instructions by ticking a checkbox before starting the annotation task. We also manually monitor the performance of the annotators throughout the annotation process and provide feedback based on common errors. Spammers or underperforming workers will be disqualified. The overall inter-annotator agreement (IAA) stands at 78% in terms of pairwise agreement, and the Fleiss kappa (Fleiss, 1971) is 0.53. The IAA and Fleiss Kappa scores for the three subtasks are closely aligned, with a difference range of ±0.05. These statistics are generally comparable to or slightly higher than those of other high-quality dataset construction works (Sap et al., 2019; Fang et al., 2021a,b; Hwang et al., 2021; Wang and Song, 2024), which indicates that the annotators are close to achieving a strong internal agreement.

B.3 Expert Verification

Finally, we seek the help of three e-commerce NLP experts, each with extensive experience in NLP research, to validate the annotations. The experts are NLP scientists with extensive experience in e-commerce NLP. They are well trained in con-

ducting NLP research and are familiar with the e-commerce domain. In contrast, AMT crowd-sourced workers are generally considered to have only a basic understanding of AI, NLP, and related fields. Therefore, recruiting experts to verify the annotated labels is critical, as they have a deeper understanding of the tasks and can better assess whether the collected labels align with the task requirements and design. They are given the same instructions as those provided to crowd-sourcing workers and asked to verify a sample of 200 annotations for each task. The high level of consistency between our expert annotators and AMT annotators, as demonstrated in Table 1, suggests that our AMT annotation is of high quality.