Learning to Perform Complex Tasks through Compositional Fine-Tuning of Language Models

Victor S. Bursztyn[1], **David Demeter**[1], **Doug Downey**[1,2], and **Larry Birnbaum**[1]

[1]Department of Computer Science, Northwestern University, Evanston, IL, USA
[2]Allen Institute for Artificial Intelligence, Seattle, WA, USA
{v-bursztyn,ddemeter}@u.northwestern.edu
{d-downey,l-birnbaum}@northwestern.edu

Abstract

How to usefully encode compositional task structure has long been a core challenge in AI. Recent work in chain of thought prompting has shown that for very large neural language models (LMs), explicitly demonstrating the inferential steps involved in a target task may improve performance over end-to-end learning that focuses on the target task alone. However, chain of thought prompting has significant limitations due to its dependency on huge pretrained LMs. In this work, we present compositional fine-tuning (CFT): an approach based on explicitly decomposing a target task into component tasks, and then fine-tuning smaller LMs on a curriculum of such component tasks. We apply CFT to recommendation tasks in two domains, world travel and local dining, as well as a previously studied inferential task (sports understanding). We show that CFT outperforms end-to-end learning even with equal amounts of data, and gets consistently better as more component tasks are modeled via fine-tuning. Compared with chain of thought prompting, CFT performs at least as well using LMs only 7.4% of the size, and is moreover applicable to task domains for which data are not available during pretraining.

1 Introduction

Philosophy, linguistics, and computer science have long debated how and whether to explicitly encode the compositionality of task structure in models of language understanding and generation (Fodor and Pylyshyn, 1988). The prevailing paradigm in today's NLP is *end-to-end learning*, in which the learning of compositional task structure is subsumed by the learning of a complex target task, with the support of increasingly powerful language models (LMs) (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020).

Recent work in compositionality in NLP has been mostly limited to semantic parsing and multi-hop reasoning for the purpose of Q&A (Shaw et al., 2021; Wolfson et al., 2020; Min et al., 2019). However, a series of recent works have proposed generating "chains of thought" as a means to expand an LM's ability to reason beyond a single forward pass (Wei et al., 2022; Zelikman et al., 2022; Nye et al., 2021). The success of chain of thought approaches suggests broader opportunities to study the use of compositional structure as a means to improve the learning of complex tasks, rather than as a byproduct of end-to-end learning.

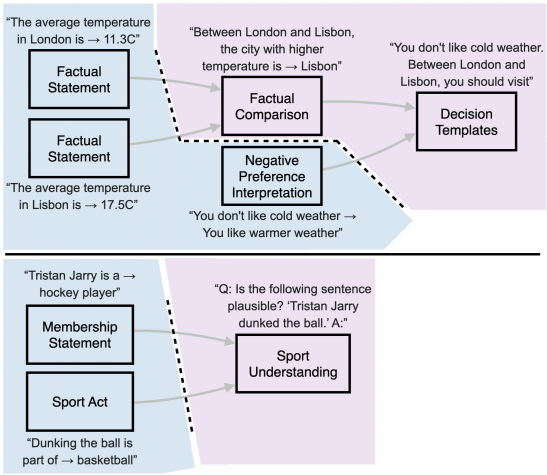

Figure 1: Component tasks involved in a recommendation prompt (above) and in sports understanding (below). In compositional fine-tuning (CFT), component tasks shaded in light blue precede those in light purple.

Breaking down a complex task into sub-tasks is a ubiquitous construct in human problem-solving. In machine learning, it has inspired curriculum learning (CL) (Bengio et al., 2009), which hypothesizes that a model should start learning from easier concepts and progress to harder ones, as humans do. In this work, we explore the idea of CL through the lens of *incremental task complexity*, which is fundamentally different from prior works in NLP centered on incremental example difficulty (e.g., organizing training data by increasing sequence length or decreasing word frequency).

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 1676–1686
December 7-11, 2022 ©2022 Association for Computational Linguistics

We propose compositional fine-tuning (CFT), a fine-tuning strategy in which sub-tasks are organized as components of a curriculum that progressively teaches a target task, as shown visually in Figure 1. CFT is novel in two ways: it is a CL approach in NLP that focuses on incremental task complexity instead of incremental example difficulty; and unlike chain of thought prompting, CFT does not depend on huge, pretrained LMs—it relies on smaller, fine-tuned LMs instead. This is advantageous because the largest LMs are hard to access and expensive, and their pretraining data, while vast, still fail to cover a wide range of domains.

We focus on conversational recommendation, which is especially rich in complex tasks (Bursztyn et al., 2021). As shown in Figure 1, a relatively short recommendation prompt may comprise component tasks as diverse as understanding a user preference—related to pragmatics—and finding an item that correctly matches the semantics of such a preference. Despite this diversity in component tasks, recommendation tasks are still underexplored in the NLP community (Penha and Hauff, 2020; Malkiel et al., 2020; Wang et al., 2021).

We make the following contributions:

- We contribute a new schema for generating recommendation datasets, which we instantiate in two domains: world travel and local dining. By design, LMs are more likely to hold prior knowledge about world cities than about local restaurants, making our released dataset challenging to different degrees.
- We propose **compositional fine-tuning (CFT)**: an approach based on decomposing a target task into component tasks, and then fine-tuning smaller LMs on a curriculum of such component tasks. We instantiate CFT in our recommendation tasks as well as the sports understanding task from (Wei et al., 2022).
- We present experiments[1] showing that CFT consistently outperforms end-to-end learning, with up to 32% gains in the local dining domain given equal amounts of training data. When compared to chain of thought prompting, we further find that CFT performs equally or better while requiring LMs only 7.4% of the size (as seen in Table 1).

[1]Data and code fully available at: https://github.com/vbursztyn/compositional-fine-tuning

Base Model	Method	Score on Decision Templates
DaVinci	8-Shot Prompting	0.83 ± 0.08
DaVinci	8-Shot Chain of Thought	**0.98 ± 0.02**
Curie	8-Shot Chain of Thought	0.50 ± 0.12
Curie	CFT on Factual Statements, Factual Comparisons, and Decision Templates	**0.95 ± 0.01**

Base Model	Method	Score on Decision Templates
DaVinci	8-Shot Prompting	0.54 ± 0.09
DaVinci	8-Shot Chain of Thought	**0.55 ± 0.07**
Curie	8-Shot Chain of Thought	0.50 ± 0.06
Curie	CFT on Factual Statements, Factual Comparisons, and Decision Templates	**0.74 ± 0.05**

Table 1: Comparison to chain of thought in the world travel domain (above) and local dining (below). CFT performs as well as chain of thought prompting for world cities and 35% better for local restaurants, with an LM only 7.4% of the size (13B vs 175B).

2 Related Work

2.1 Chain of Thought Approaches

Chain of thought approaches are the most recent stream of research connected to ours (Wei et al., 2022; Zelikman et al., 2022; Gu et al., 2021; Nye et al., 2021; Talmor et al., 2020; Rajani et al., 2019). Wei et al. (2022) recently proposed chain of thought prompting, the idea that very large LMs can do much better at "system 2 tasks"—tasks that require deeper reasoning skills, such as math problems or symbolic reasoning—if they are given examples in the prompt that explicitly describe the intermediate steps of the task. Although effective in improving accuracy, its dependency on huge, pretrained LMs still limits chain of thought prompting. In contrast, our CFT approach shows similar gains vs end-to-end learning on our tasks, but in a setting with LMs that are more than an order of magnitude smaller.

Among these previous works, we highlight (Talmor et al., 2020) as an attempt to study the effect of factual knowledge injection in LM performance on tasks that involve chaining different facts. In our ablation studies in §5, we cover a configuration that is analogous to theirs and show improvements from having an additional component task.

2.2 Compositionality in Question Answering

Many recent works in the Q&A literature have strived to study compositionality on either a question or system level. At the question level, learning to decompose a question into smaller questions and reasoning over these sub-questions in order to

arrive at a final answer (multi-hop reasoning) has been a common goal (Khot et al., 2020; Min et al., 2019; Yang et al., 2018; Khashabi et al., 2018). At the system level, investigating a system's ability to generalize from question types seen during training (e.g., "Who directed *x*?") to new, unseen instances of the same type (e.g., "Who directed Inception?") has attracted increasing attention (Keysers et al., 2019). Further works have explored both problems—multi-hop reasoning and compositional generalization—through the lens of semantic parsing (Wolfson et al., 2020; Shaw et al., 2021).

In contrast, we focus on a new schema of recommendation tasks, where by design the decomposition required to perform the task is not transparent from the question itself but is known *a priori* across a variety of domains. This schema allows us to evaluate the effectiveness of a novel CFT approach in two domains, and to compare it against the recent chain of thought prompting approach.

2.3 Curriculum Learning (CL)

The seminal work in CL (Bengio et al., 2009) included a language modeling experiment in which training data were ordered from most to least frequent based on corpus statistics. Since then, many works in NLP have explored different measures of example difficulty, as simple as sequence length for NLG (Rajeswar et al., 2017) and as complex as estimates based on model performance (Sachan and Xing, 2016; Xu et al., 2020). However, such a focus on example difficulty has kept these works distant from the "shaping hypothesis" that inspired (Bengio et al., 2009): the idea that a complex task can be taught by breaking it into a sequence of smaller steps of incremental complexity (Krueger and Dayan, 2009). In this work, instead of incremental example difficulty, we explore a different approach to incremental complexity based on organizing training data around component tasks.

To the best of our knowledge, the closest works can be found in the domain of spatial navigation instructions (Dan et al., 2021; Lake and Baroni, 2018), in which an LM starts with simple block-moving instructions and progresses to compositional ones. However, our work differs in the diversity of our component tasks, in the more extensive experimentation that ensues, and in the applicability of CFT to other similarly diverse domains.

3 Problem Definition

The recommendation task depicted in Figure 1 takes as input a set of *items* (set I) and a set of user preferences (set P), such that $Recommend(P, I)$ outputs the item that best matches the user preferences. In its simplest form, we have a pair of items $I = \{i_1, i_2\}$ and a single preference $P = \{p\}$, such that $Recommend(\{p\}, \{i_1, i_2\})$. This form maps naturally to what we call a **"decision template,"** composed of two sentences: one with a preference (e.g., "You don't like cold weather."), and another with a sufficiently different pair of items (e.g., "Between London and Lisbon, you should visit" → Lisbon). We use the term "decision" because $Recommend(P, I)$ can be considered an instance of a decision task where I represents options and P expresses the criteria to be applied.

Breaking down $Recommend(\{p\}, \{i_1, i_2\})$ into component tasks, the first task consists of comparing two items along a given attribute. This can be defined as $Compare(a, o, \{i_1, i_2\})$ that takes as input an attribute a (e.g., temperature), an order o (e.g., higher), and the two items, and then outputs the item that satisfies the comparison. We call this task a **"factual comparison"** (e.g., "Between London and Lisbon, the city with warmer weather is" → Lisbon), which is further decomposed into **"factual statements"** that simply enunciate the attribute value of an item (e.g., "The average temperature in Lisbon is" → 17.5C).

With that, a domain D can be formalized as $D = (I^{full}, A)$ where I^{full} is the full set of items and A the set of attributes. Considering the world travel domain, for example, I^{full} may represent a list of well-known cities and $A = \{temperature, population\}$ the average temperature and total population, respectively. We instantiate this schema in our experiments in §5, but it can be used to generate new recommendation datasets or repurposed for other decision tasks.

3.1 A Challenging Task for Pretrained LMs

Even state-of-the-art LMs such as GPT-3 (Brown et al., 2020) struggle at this recommendation task, as evidenced by experiments fully described in §5. As shown in Table 1, 175B parameter DaVinci in 8-shot mode can accurately recommend 83% of test cases in the world travel domain, but only 55% in the local dining domain, which cannot be improved with chain of thought prompting. As shown in Tables 2 and 3, performance is very low with 13B

parameter Curie in 0-shot mode: only 6% of test cases lead to correct recommendations in the world travel domain, and only 18% in the local dining domain. It is with this challenge in mind that we propose compositional fine-tuning (CFT).

4 Compositional Fine-Tuning (CFT)

CFT consists of three sequential steps: **Decompose**, where we break the complex task into component tasks; **Demonstrate**, where we generate examples for each of these component tasks; and **Fine-Tune**, where we organize the training data according to task-level compositionality.

4.1 Decompose

For the Decompose step, Figure 1 and §3 establish the component tasks behind decision templates. In this work, the decomposition is performed manually in order to evaluate whether using compositional structure during fine-tuning can potentially improve the learning of complex tasks. This assumption is similar in spirit to the step-by-step "exemplars" manually provided in chain of thought prompting (Wei et al., 2022). In line with their findings, we believe that the confirmation of our hypothesis helps to motivate further research in automating this step.

4.2 Demonstrate

Once we have a diagram with component tasks, we need to demonstrate them, preferably with some degree of natural language variation. In our recommendation dataset, we implement a single factual comparison (e.g., comparing London and Lisbon with a = temperature and o = higher) using two different phrasings, and the corresponding decision template using eight different phrasings.

For factual comparisons, the first phrasing *directly* refers to the attribute value (e.g., "Between London and Lisbon, the city with higher average temperature is" → Lisbon), and the second phrasing *indirectly* refers to the same attribute value (e.g., "Between London and Lisbon, the city with warmer weather is" → Lisbon).

For decision templates, following (Bursztyn et al., 2021), each possible a and o combination (i.e., each preference) is phrased in either a positive form (e.g., "You like warmer weather.") or a negative form (e.g., "You don't like cold weather."). Additionally, each of these two phrasings can be rephrased in the first- or third-person ("Someone..."), as well as in a subjunctive form (e.g., "You are looking for a city with warmer weather. If I were you, I would visit").

Completing our setting, as seen in §3, factual statements are represented by a single phrasing that simply enunciates an attribute value. Therefore, with $|A| = 2$, each pair of items yields four factual statements, eight factual comparisons, and 32 decision templates.[2]

4.3 Fine-Tune

Once all these phrasings are populated with item pairs from one of our two domains, we are done generating our training data. For the Fine-Tune step, we organize such data according to component tasks' dependencies. As seen in Figure 1, there are tasks that do not depend on any other (in light blue), while there are tasks that do (light purple).

From left to right, we consider each colored layer a phase in our curriculum: the first phase includes data for factual statements and negative preference interpretations; and the second phase includes factual comparisons and decision templates. As explained in §5.4, negative preference interpretations are a small component task that is useful when decision templates are partially seen during training.

Within each phase, we find empirical benefits in shuffling training data. We put forward two potential explanations for that. First, in earlier phases, shuffling ensures that all component tasks included in a phase are equally learned by the end of it, helping in the next one. Second, in later phases, shuffling should also help training to converge because these later tasks are increasingly similar to the target task.

5 Experiments

Considering our problem and CFT, we pose the following questions:

- **RQ1**: How does CFT compare with end-to-end learning?
- **RQ2**: How does CFT compare with chain of thought prompting?

We conduct four experiments to answer RQs 1 and 2, leveraging data from two domains: world travel, and local dining. World travel represents a less challenging scenario, considering that the LM is more likely to have prior knowledge about world

[2]Fully available at: `https://bit.ly/3xeP8E1`

cities and their various attributes. Local dining, conversely, represents a more challenging scenario, as the LM is less likely to exhibit any prior knowledge. Our experiments are based on GPT-3's Curie model (13B parameters), which was the largest LM available for fine-tuning at that time.[3]

5.1 Data

Each domain comprises two attributes. For world cities, we have $A = \{temperature, population\}$ where average city temperatures are obtained from Wikipedia[4] and city populations from SimpleMaps 2019.[5] After merging items from both sources, we end with 347 well-known cities (>50k inhabitans) from around the globe, such that $D_c = (I_c^{full}, \{temperature, population\})$ and $|I_c^{full}| = 347$. For local restaurants, we randomly sample 240 restaurants from the city with most restaurants in the Yelp dataset[6], Toronto. We have $A = \{price, distance\}$ where restaurant prices are obtained from Yelp and distances to a hypothetical location are randomly generated, thus limiting the LM's access to prior knowledge in this scenario. With that, we have $D_r = (I_r^{full}, \{price, distance\})$ and $|I_r^{full}| = 240$.

In terms of component tasks, we have 694 factual statements for the cities domain and 480 for restaurants, covering two attributes per item. Whenever factual statements are provided in CFT, they always cover I^{full} entirely in order to give the LM full knowledge of the attribute values.

However, for factual comparisons and decision templates, we wish to evaluate the LM's ability to generalize to cities and restaurants not seen in such statements during training. Therefore, we split I^{full} between training and test items before we generate item pairs. We keep only 30% of I^{full} for training, and we sample from the remaining 70% when testing a fine-tuned LM. This way, cities and restaurants used at test time are only seen during training in factual statements, *never* in factual comparisons or decision templates.

When generating examples from these itemsets, we enforce minimum differences in attribute values: for pairs of cities, a 10C difference in temperature and a 2.5M difference in population; and for pairs of restaurants, a 1 dollar-sign difference in price and a 3 mile difference in distance. Factual comparisons and decision templates are only populated with item pairs that exhibit at least these differences in attribute values.

When applying these rules to the training items, we end with roughly 1,970 pairs of cities and 2,320 pairs of restaurants. In combination with the phrasings in §4.2, we have roughly 15.8k factual comparisons for cities and 18.5k for restaurants; and 63k decision templates for cities and 74.2k for restaurants. To make sure that factual comparisons and decision templates are represented by similar amounts of training data, we sample decision templates until the number of tokens match that of factual comparisons.

Lastly, across all factual comparisons and decision templates, we flip the order of the items (e.g., London and Lisbon) with a 50% chance so that the LM cannot use position as a short-cut for the answer. Our data is made fully available to the research community at `https://github.com/vbursztyn/compositional-fine-tuning`

5.2 Evaluation

Once we fine-tune Curie on a given CFT configuration, we may evaluate the model on a task from the second phase—either factual comparisons or decision templates—in a given domain. We generate examples by applying the same rules seen in the generation of training data, but now applied to the held-out test items. When evaluating factual comparisons, we report the average performance on 1.6k test cases (200 examples per phrasing, times eight phrasings); and when evaluating decision templates, we report the average performance on 6.4k test cases (200 times 32 phrasings).

A single test case is evaluated by generating the top 5 predictions with greedy decoding. If the answer is more likely than the wrong candidate, then this test case score is 1; otherwise, it is 0.

5.3 Experiment 1: The Role of Components

In our first experiment, we want to answer RQ1 by examining the role of component tasks in the learning of our complex task. To this end, we focus on the deepest dependencies of decision templates, i.e., factual comparisons and factual statements. We ablate each of these component tasks in different CFT configurations while measuring model

[3] `https://beta.openai.com/docs/engines`
[4] `https://en.wikipedia.org/wiki/List_of_cities_by_average_temperature`
[5] `https://simplemaps.com/data/world-cities`
[6] `https://www.yelp.com/dataset`

Model fine-tuned on			Average score on	
Factual Statements	Factual Comparisons	Decision Templates	Factual Comparisons	Decision Templates
No	No	No	0.16 ± 0.06	0.06 ± 0.07
Yes	No	No	0.11 ± 0.07	0.27 ± 0.15
No	Yes	No	0.90 ± 0.02	0.54 ± 0.17
No	No	Yes	0.74 ± 0.16	0.89 ± 0.04
Yes	Yes	No	0.95 ± 0.02	0.63 ± 0.18
Yes	No	Yes	0.78 ± 0.22	0.92 ± 0.02
No	Yes	Yes	0.89 ± 0.03	0.88 ± 0.03
Yes	Yes	Yes	0.96 ± 0.01	**0.95 ± 0.01**

Table 2: Experiment 1 in the world travel domain. CFT with factual statements or factual comparisons consistently increases performance. The best configuration includes all tasks (row #8, in boldface).

Model fine-tuned on			Average score on	
Factual Statements	Factual Comparisons	Decision Templates	Factual Comparisons	Decision Templates
No	No	No	0.16 ± 0.04	0.18 ± 0.05
Yes	No	No	0.00 ± 0.00	0.13 ± 0.06
No	Yes	No	0.52 ± 0.11	0.51 ± 0.10
No	No	Yes	0.50 ± 0.06	0.52 ± 0.07
Yes	Yes	No	0.66 ± 0.13	0.54 ± 0.10
Yes	No	Yes	0.50 ± 0.05	0.55 ± 0.04
No	Yes	Yes	0.53 ± 0.12	0.53 ± 0.10
Yes	Yes	Yes	0.75 ± 0.05	**0.74 ± 0.05**

Table 3: Experiment 1 in the local dining domain. The best configuration, again with all tasks, outperforms the second best (row #6) by 35%.

performance on decision templates. Although performance on decision templates is our primary endpoint, we secondarily measure performance on factual comparisons. Tables 2 and 3 show the results of each CFT configuration for world cities and local restaurants, respectively.

We can see that factual statements consistently improve performance: on Table 2, they improve performance by 3-17%, including an 8% improvement of row #8 (the best configuration) relative to row #7. On Table 3, they improve performance by 5-40%, with maximum improvement on row #8 (again, the best configuration) over row #7. This component task has a small footprint—694 factual statements for cities and 480 for restaurants—and is the most likely one to be contemplated in end-to-end learning schemes (e.g., when knowledge bases are included during training).

We can also see that factual comparisons monotonically increase performance: on Table 2, although there is no change from row #7 to row #4, they improve row #8 by 3% over row #6. On Table 3, although again there is no change from row #7 to row #4, they improve row #8 by 35% over row #6. Therefore, in the best configuration, the effect of factual comparisons is comparable to that of factual statements (35% vs 40%). Scores on factual comparisons also suggest that the learning of both tasks

Total # of tokens	Model fine-tuned on			Average score on	
	Factual Statements	Factual Comparisons	Decision Templates	Factual Comparisons	Decision Templates
186,413	Yes	No	Yes	0.78 ± 0.22	0.92 ± 0.02
367,144	Yes	No	Yes	0.75 ± 0.26	0.93 ± 0.02
367,157	Yes	Yes	Yes	0.96 ± 0.01	**0.95 ± 0.01**

Table 4: CFT vs end-to-end learning (plus facts) with equal amounts of training data for world cities. The gap practically does not change.

Total # of tokens	Model fine-tuned on			Average score on	
	Factual Statements	Factual Comparisons	Decision Templates	Factual Comparisons	Decision Templates
293,967	Yes	No	Yes	0.50 ± 0.05	0.55 ± 0.04
581,370	Yes	No	Yes	0.50 ± 0.10	0.56 ± 0.04
581,379	Yes	Yes	Yes	0.75 ± 0.05	**0.74 ± 0.05**

Table 5: CFT vs end-to-end learning (plus facts) with equal amounts of training data for local restaurants. Again, the gap practically does not change.

in the second phase of CFT is indeed synergistic.

Interestingly, the second best configuration (row #6) represents an end-to-end learning scheme with access to factual knowledge, which is similar to configurations studied by (Talmor et al., 2020). However, because factual comparisons and decision templates were designed to have the same number of tokens in our CFT configurations, row #8 has access to almost two times as much training data as row #6.

For this reason, we run a follow-up experiment to test if the performance gains are indeed explained by the presence of more components, and not by access to more data. We increase the number of decision templates in row #6 until we have equal amounts of training data.

On Tables 4 and 5, we can see how the quantity of training data does not explain the performance difference. With equal amounts of training data, **our CFT configuration with more component tasks consistently outperforms end-to-end learning** with factual knowledge: by 2% for world cities, and up to 32% for local restaurants. Importantly, CFT yields substantial improvements in the more challenging scenario where the LM has less prior knowledge on items, thus a performance that is further from the upper bound.

5.4 Experiment 2: Attribute Transfer

In our second experiment, we continue to address the question: Are more component tasks better for CFT? To complement Experiment 1, we split the original decision templates data into two folds, one for each attribute, and we ablate these folds in each domain while measuring model performance on the entire set of decision templates.

Model fine-tuned on					Average score on	
Factual Statements	Factual Comparisons	Decision Templates (Weather)	Decision Templates (Population)	Negative Preference Interpretations	Factual Comparisons	Decision Templates
Yes	Yes	No	Yes	No	0.94 ± 0.01	0.84 ± 0.19
				Yes	0.95 ± 0.01	**0.89 ± 0.10**
		Yes	No	No	0.95 ± 0.01	0.88 ± 0.12
				Yes	0.96 ± 0.01	**0.90 ± 0.14**

Table 6: Experiment 2 for world cities. Adding only 12 negative preference interpretations improves performance by 2-6% on the two folds.

Model fine-tuned on					Average score on	
Factual Statements	Factual Comparisons	Decision Templates (Price)	Decision Templates (Distance)	Negative Preference Interpretations	Factual Comparisons	Decision Templates
Yes	Yes	No	Yes	No	0.64 ± 0.13	0.56 ± 0.06
				Yes	0.70 ± 0.08	**0.65 ± 0.05**
		Yes	No	No	0.68 ± 0.13	0.67 ± 0.17
				Yes	0.67 ± 0.17	**0.69 ± 0.16**

Table 7: Experiment 2 for local restaurants. Again, adding only 12 negative preference interpretations improves performance by 3-16% on the two folds.

On Tables 6 and 7, when analyzing the configurations on rows #1 and #3, we notice that learning is partially transferred to the unseen attribute, with performance drops of 7-24% relative to rows #8 of Tables 2 and 3. We also notice that unseen preferences phrased in the negative form (e.g., "You don't like cold weather.") are the biggest source of error. Therefore, we add one extra component task to these CFT configurations: negative preference interpretations.[7]

As seen in Figure 1, these interpretations simply teach the LM to interpret negations (e.g., "You don't like cold weather" → "You like warmer weather"), consisting of only 12 statements for each domain—a tiny footprint. Interestingly, this small component task indeed improves performance across all configurations: 2-6% for world cities, and 3-16% for local restaurants. Analyzing *exclusively* the decision templates containing negations, performance improves by an average of 9% for cities and 15% for restaurants.

5.5 Experiments 3 & 4: Comparison to Chain of Thought Prompting

In our two final experiments, we want to answer RQ2 by comparing CFT with chain of thought prompting. We do this from two perspectives: first, from the viewpoint of the recommendation tasks introduced in this work; and second, from the viewpoint of sports understanding, a commonsense task studied by (Wei et al., 2022).

5.5.1 Recommendation Tasks

We instantiate chain of thought prompting in our two domains as described in (Wei et al., 2022), with $k = 8$ per their code. For each domain, we manually construct 8 "exemplars" using item pairs that only exist in the training set. Each exemplar includes relevant factual statements, how these are used in a factual comparison, and how this is used to answer the overarching decision template.[8] As recommended by (Wei et al., 2022), we leverage pretrained DaVinci (175B parameters), which is the largest LM we have access to; but we also test pretrained Curie (13B parameters), which is the base model for all our CFT runs. Finally, to isolate the effect of chain of thought, we test DaVinci with regular 8-shot prompting. Due to the much higher costs of 8-shot chain of thought prompting with DaVinci, in these runs we reduced the sample size to 100 test cases per phrasing (3.2k test cases in total). Results can be seen on Table 1.

Interestingly, for world cities, chain of thought prompting with pretrained DaVinci can answer almost all test cases (98% of them), which is not too far ahead of CFT using Curie (95%). Pretrained DaVinci with regular 8-shot prompting performs substantially worse (83%), which shows how both chain of thought prompting and CFT are more effective in this scenario. However, chain of thought prompting with pretrained Curie performs as low as random chance (50%). This suggests that **CFT is capable of similar performance while requiring an LM only 7.4% of the size** (13B vs 175B).

For local restaurants, results are even more favor-

[7] Fully available at: https://bit.ly/3O0WIce

[8] Prompts available at: https://bit.ly/3rDiwS6

able for CFT. All approaches based on pretrained LMs struggle in this more challenging domain, performing only slightly above chance on the price attribute (up to 65%). CFT is the only approach capable of answering 74% of all test cases, which shows a fundamental limitation of chain of thought prompting when faced with domains where facts are not as easily accessible.

5.5.2 Sports Understanding

Next, we instantiate CFT in the sports understanding task from (Wei et al., 2022), which consists in determining if a sentence mentioning a certain well-known sport player performing a certain sport act is plausible. Component tasks can be seen in Figure 1. We note that this task is very similar in structure to the "hypernymy" and "meronymy" tasks from (Talmor et al., 2020), thus also representing a large category of inferential tasks.

To gather data for the Demonstrate step, we resort to a similar strategy to (Zelikman et al., 2022): using the generated chains of thought in (Wei et al., 2022), we filter all the explanations that lead to a correct answer. From these 815 examples (originally 980, given their accuracy of 83% on the task), we parse 390 unique membership statements and 182 unique sport acts. Our CFT configuration includes all membership statements and sport acts, but only 50% of question-answer pairs in a 2-fold cross-validation scheme. Performance on the two folds are 95.83% and 95.57%. Therefore, like in §5.5.1, this again suggests that CFT is capable of similar performance to chain of thought promping while requiring an LM only 7.4% of the size.

6 Discussion

Across our experiments, both in our released dataset and in sports understanding, we found consistent evidence that LMs may benefit from compositional structure when learning a complex task. Although we obtain improvements from *three very different* types of component tasks—factual statements, factual comparisons, and negative preference interpretations—standard end-to-end learning schemes tend to overlook the explicit use of compositional structure or focus only on factual knowledge. We hope to encourage further research in other principled, task-agnostic methods for leveraging compositional structure in LM fine-tuning.

Compared to chain of thought prompting, methods based on fine-tuning have at least two advantages. First, 100+B parameter LMs are hard to access and expensive. When using DaVinci with 8-shot chain of thought prompting, each of our examples costs USD 7.5 cents,[9] which is roughly *50 times more expensive* than fine-tuning Curie with CFT. Second, many domains are not within pretraining data (e.g., due to proprietary data), so it is necessary to consider fine-tuning methods that inject custom data *and* preserve the LM's ability to chain thought. This limitation of strictly prompt-based methods has been recently noted by (Zhou et al., 2022), and we emphasize it in light of our results in the local dining domain.

While CFT certainly requires more data than chain of thought prompting, interestingly, we found it to be remarkably more efficient w.r.t model size. Works leading up to (Wei et al., 2022) have hypothesized that generating intermediate steps expands an LM's ability to reason beyond a single forward pass (Nye et al., 2021); however, CFT suggests that we have not yet exhausted what can be done within one forward pass. Considering this optimal use of smaller models, CFT can be potentially used for "distilling" a complex multi-step workflow based on very large LMs—as seen in (Wu et al., 2022)—into one smaller LM.

We believe that our findings motivate research in fully automating the steps behind CFT. For the Decompose step, prior NLP works in decomposition (Dan et al., 2021; Sakaguchi et al., 2021; Perez et al., 2020) could be expanded to this context. Zhou et al. (2022), in particular, point to an interesting direction with "least-to-most prompting." We note that the automation of the Decompose step is also warranted by chain of thought prompting, in which decomposition is also performed manually. For the Demonstrate step, automation would entail a few sub-steps: (i) exploring the lexical space (e.g., the space of possible preferences); (ii) generating paraphrases to increase natural language variation (e.g., our phrasings); and (iii) populating these phrasings with data from a domain of interest (i.e., I^{full}). In contrast, automating the Fine-Tune step is straightforward.

Finally, any models generated with CFT can be viewed as components themselves. For example, if a model is not able to handle larger sets of preferences or items (i.e., $|P| > 1$ or $|I| > 2$) in a decision template without losing performance, then one potential solution is to use an upstream agent to

[9] Two requests required, with 625 prompt tokens each. Querying DaVinci currently costs USD 6 cents per 1k tokens.

break a complex case into smaller ones (i.e., with $|P| = 1$ and $|I| = 2$) and combine their outputs. Khot et al. (2022) propose a framework than can be applied to this end.

7 Conclusion & Future Work

In this work, we proposed CFT as an improvement upon end-to-end learning. To enable research on this topic, we developed a new schema for generating recommendation datasets, which we instantiated in two domains. We showed that CFT indeed *consistently* outperforms end-to-end learning, as much as 32% for local dining. Furthermore, we found evidence suggesting that more component tasks can be beneficial for CFT. Finally, instantiating chain of thought prompting in our dataset and CFT in sports understanding, we found CFT to be as good or better with LMs only 7.4% of the size.

For future work, we plan to apply CFT to tasks with even more depth and breadth as in Figure 1, as well as to conventional spatial navigation datasets (e.g., SCAN from Lake and Baroni (2018)). At the same time, we encourage others to test CFT on the large family of tasks that fit the inference types already covered in Figure 1: those including facts, comparisons, criteria interpretations, and decisions, as seen in the recommendation task; or those including facts and assertions, as seen in sports understanding. We also plan to explore ways for fully automating the Decompose and Demonstrate steps.

8 Limitations

This work focuses on testing if CFT outperforms end-to-end learning and chain of thought prompting in two very different domains. Despite the positive evidence, it remains to be seen: (i) if task decomposition can be fully automated, and (ii) if different decompositions—in the case of tasks that allow for multiple decompositions—yield similar results. Both are second-order research questions that can be pursued once compositionality has been confirmed to improve performance. Importantly, both questions have been left open in the initial chain of thought work as well. We hope that our results will add to theirs in attracting more attention to these questions in the future.

Another limitation of this work is that CFT is not applicable to several decomposition datasets that have been proposed. For example, a dataset focused on compositional generalization may include many different types of questions, each requiring different types of intermediate steps. CFT is not designed for intermediate steps that carry out very heterogeneous logic. Nonetheless, as shown in the recommendation tasks, CFT is still relevant for a substantial family of tasks with real-world applicability.

Lastly, this work is limited by its focus on the English language, and by the use of GPT-3 for its unique range of model sizes. For example, when we discuss that CFT on a 13B parameter model (Curie) is a much cheaper alternative to chain of thought prompting on a 175B parameter model (DaVinci), the finding is limited to this setting. It is important to replicate this work on other languages and models, which we plan to do as these become available.

Acknowledgements

We would like to thank reviewers for their helpful feedback. This work was supported in part by gift funding from Adobe Research and by NSF grant IIS-2006851.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Victor Bursztyn, Jennifer Healey, Nedim Lipka, Eunyee Koh, Doug Downey, and Larry Birnbaum. 2021. "it doesn't look good for a date": Transforming critiques into preferences for conversational recommendation systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1913–1918, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Soham Dan, Xinran Han, and Dan Roth. 2021. Compositional data and task augmentation for instruction following. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2076–2081, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Yuling Gu, Bhavana Dalvi Mishra, and Peter Clark. 2021. Dream: Uncovering mental models behind language models. *arXiv preprint arXiv:2112.08656*.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Tushar Khot, Kyle Richardson, Daniel Khashabi, and Ashish Sabharwal. 2022. Hey AI, can you solve complex tasks by talking to agents? In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1808–1823, Dublin, Ireland. Association for Computational Linguistics.

Kai A Krueger and Peter Dayan. 2009. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Itzik Malkiel, Oren Barkan, Avi Caciularu, Noam Razin, Ori Katz, and Noam Koenigstein. 2020. RecoBERT: A catalog language model for text-based recommendations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1704–1714, Online. Association for Computational Linguistics.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy. Association for Computational Linguistics.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Gustavo Penha and Claudia Hauff. 2020. *What Does BERT Know about Books, Movies and Music? Probing BERT for Conversational Recommendation*, page 388–397. Association for Computing Machinery, New York, NY, USA.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. 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. *Journal of Machine Learning Research*, 21:1–67.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942.

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. 2017. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*.

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 453–463.

Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. 2021. proScript: Partially ordered scripts generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2138–2149, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938, Online. Association for Computational Linguistics.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. *Advances in Neural Information Processing Systems*, 33:20227–20237.

Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Kam-Fai Wong, and Daxin Jiang. 2021. Finetuning large-scale pre-trained language models for conversational recommendation with knowledge graph. *arXiv preprint arXiv:2110.07477*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break It Down: A Question Understanding Benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Eric Zelikman, Yuhuai Wu, and Noah D Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.