The semantic relations in LLMs: an information-theoretic compression approach

Yu-Hsiang Tseng [†], Pin-Er Chen[‡], Da-Chen Lian[‡], Shu-Kai Hsieh[‡]
[†]Department of Linguistics, University of Tübingen
[‡]Institute of Linguistics, National Taiwan University
yu-hsiang.tseng@uni-tuebingen.de, cckk2913@gmail.com
{d08944019,shukaihsieh}@ntu.edu.tw

Abstract

Compressibility is closely related to the predictability of the texts from the information theory viewpoint. As large language models (LLMs) are trained to maximize the conditional probabilities of upcoming words, they may capture the subtlety and nuances of the semantic constraints underlying the texts, and texts aligning with the encoded semantic constraints are more compressible than those that do not. This paper systematically tests whether and how LLMs can act as compressors of semantic pairs. Using semantic relations from English and Chinese Wordnet, we empirically demonstrate that texts with correct semantic pairings are more compressible than incorrect ones, measured by the proposed compression advantages index. We also show that, with the Pythia model suite and a fine-tuned model on Chinese Wordnet, compression capacities are modulated by the model's seen data. These findings are consistent with the view that LLMs encode the semantic knowledge as underlying constraints learned from texts and can act as compressors of semantic information or potentially other structured knowledge.

Keywords: compression, arithmetic encoding, lexical resource, Chinese Wordnet, large language model

1. Introduction

The recent achievement of large language models (LLM) has driven explorations of interactions between symbolic, knowledge-driven approaches and subsymbolic, data-driven models (Tiddi and Schlobach, 2022; Colon-Hernandez et al., 2021). The motivation not only stems from the apparent practical values: improving performance on knowledge-intensive tasks and reducing model hallucinations, but also from exploring how such knowledge is learned from the unstructured textual inputs. Indeed, studies have shown such models not only rapidly saturate benchmarks and reach, if not exceed, human baselines (Kiela et al., 2021; Zhong et al., 2022; OpenAI, 2023), but they also learn from the texts substantial structured world or linguistic knowledge, for example, sentential structure (Linzen and Baroni, 2021), factual and commonsense knowledge (Petroni et al., 2019; Luo et al., 2023), and lexical categories (Tenney et al., 2019). This leads to an interesting question: how does the model encode the structured knowledge learned from the unannotated syntagmatic raw texts?

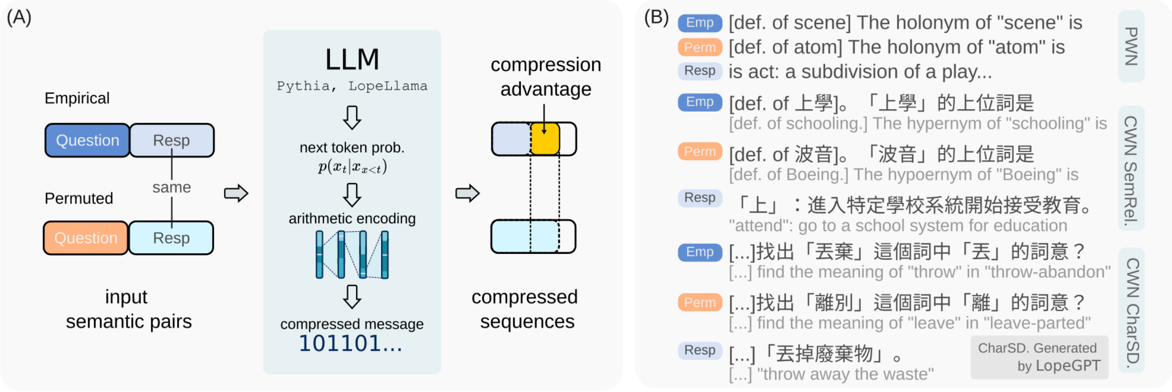
In this paper, we offer an angle and empirical findings of information-theoretic *compression* as a high-level functional view of how a deep learning model encodes structured knowledge during training. The role of compression is best seen in the written form of linguistic communication. For effective communication between a writer and a reader, they must share common backgrounds. One of the backgrounds can be English morphological agreement, which makes some text parts more predictable. For example, seeing an "I am" in the sentence, one will be *less surprised* when seeing a verb with the suffix "-ing" afterward (Juola, 1998).

Morphology, along with syntactical structures, help the writers to build a structured text stream. Texts having structures are more predictable from the previous context, which, in information theory, takes less effort to convey. According to Shannon(1948)'s source code theorem, the more predictable a message is, the less information content it carries, and the more compressible it is. One can study linguistic properties based on their compressibility. For example, researchers study the relationship between linguistic complexity and compressibility of different languages. They manipulated the texts on morphological, syntactical, and pragmatical levels of a given language and studied their impact on the size of the compressed text by a text-based compressor (Juola, 2008; Ehret, 2018).

Structures in texts are not limited to ones signaled with linguistic forms, the world and semantic knowledge is also a shared background among language users. This knowledge acts as a semantic constraint underlying the text, which should also affect the compressibility but might be far more subtle than linguistic forms and may not be fully captured by a text-based compressor. Yet, the current LLMs have achieved remarkable performance in various languages tasks, it is likely they can act as a compressor which is sensitive to the subtlety of semantic knowledge.

The exploration of LLMs acting as a compressor

21 May, 2024.



Emp [def. of scene] The holonym of "scene" is
Perm [def. of atom] The holonym of "atom" is
Resp is act: a subdivision of a play...

Emp [def. of 上學]。「上學」的上位詞是
[def. of schooling.] The hypernym of "schooling" is
Perm [def. of 波音]。「波音」的上位詞是
[def. of Boeing.] The hypoernym of "Boeing" is
Resp 「上」：進入特定學校系統開始接受教育。
"attend": go to a school system for education

Emp [...]找出「丟棄」這個詞中「丟」的詞意？
[...] find the meaning of "throw" in "throw-abandon"
Perm [...]找出「離別」這個詞中「離」的詞意？
[...] find the meaning of "leave" in "leave-parted"
Resp [...]「丟掉廢棄物」。
[...] "throw away the waste"

Figure 1: (A) This study explores the compression advantage of different semantic pairs with different LLMs. The compression advantage is measured with semantic pairs, each of which comprises sequences of correct pairing (empirical) and incorrect ones (permuted). We use arithmetic encoding with the LLM-predicted probability distributions for each token to compress the sequence. The differences in bit length between the compressed empirical and permuted sequences are defined as *compression advantage*. (B) Different types of semantic pairs. PWN are pairs of semantic relations from Princeton WordNet; *CWN SemRel* are semantic relations from CWN; *CWN CharSD* are novel character sense disambiguation sequences not seen by the tested LLMs.

is motivated by both the machine learning and psycholinguistics literature. On the machine learning side, the LLMs are trained to maximize the log probabilities of the following token, which are equivalent to minimizing the bits required for encoding the message (Deletang et al., 2024). That is, the probability distribution produced by LLM may optimally encode the message. (2) From psycholinguistics, implicit from the next-token prediction assumes there is an internal state from which the prediction is derived (Ryskin and Nieuwland, 2023). For autoregressive transformer-based LLMs, these internal states are contextualized and always updated up to the current token, thereby capturing the semantic interdependencies among the texts. Therefore, the LLMs are well-posed as a strong compressor for semantic constraints.

To systematically analyze whether and to what extent the LLMs compress semantic knowledge, we use semantic relations found in English and Chinese Wordnet. We conduct experiments and compute the corresponding *compression advantage*. These experiments use semantic pairs derived from the Princeton WordNet and the Chinese Wordnet (CWN). Each pair includes an empirical sequence, which has a correct semantic pairing, and a permuted one. The underlying rationale is that if the LLMs encode semantic constraints, the empirical sequence should be more compressible, thus increasing the compression advantage. We ask two questions in this paper: (1) whether the LLMs indeed better compress the empirical semantic pairs. (2) how the fine-tuning process affects the model's compression capacities (See Figure 1 for a general overview.)

The rest of the paper is organized as follows. We briefly review the literature on incorporating linguistic knowledge into large language models and how compression offers insights into the model-learned constraints. Next, we describe the proposed compression advantage and the experiments. In Section 4, we introduce LopeLlama[1], which is fine-tuned with the Chinese Wordnet, and compare the compression capacities to the base model on three different datasets.

2. Related Work

In addition to examining the LLMs as a compressor of the semantic pairs, we study how the additional data of semantic relations, either through fine-tuning or retrieval-augmented generation affect the compression advantage. Thus, we briefly review the fine-tuning literature followed by the literature seeing LLMs as compressors.

2.1. Fine-tuning LLMs

Various approaches have been proposed to incorporate linguistic resources or structured knowledge into large language models (Tom Brown et al., 2020; Raffel et al., 2020; Ouyang et al., 2022; Hu et al., 2023). These strategies include the input, architecture, or output injection to a pretrained model or their combinations (Colon-Hernandez et al., 2021; Wang et al., 2021a,b). For instance, the input injection strategy involves converting knowledge

[1] LopeLlama's Huggingface repo will be available after the anonymized review. The code repo: `https://github.com/seantyh/llmcomp/`

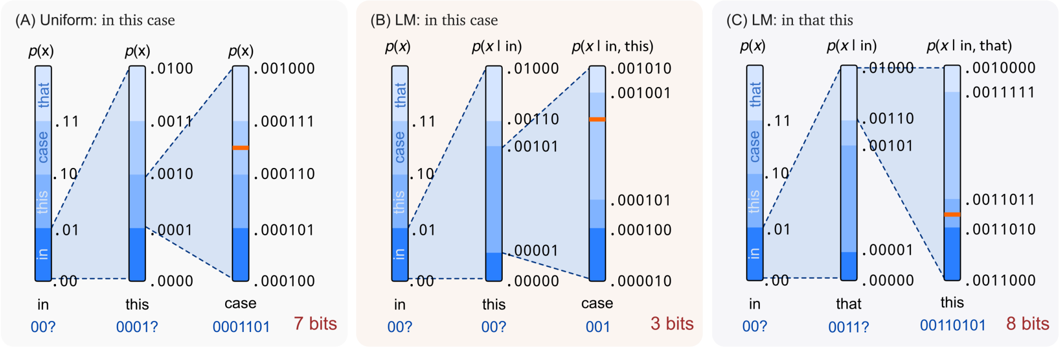


Figure 2: The schematic illustration of arithmetic encoding. Each panel shows the encoder following different probability distributions of a three-word sequence, "*in this case*". The encoder compresses one word in each step, assigning a unique interval to the word based on its probability, and the precision needed to represent the interval determines the length of the compressed message. (A) In the uniform distribution, the compressed message length is 7 bits. (B) When guided by a suitable conditional probability distribution (such as provided by an LLM), the resulting compressed message is shorter. However, (C) when the conditional probability is misspecified, the message becomes longer.

triples into masked sequences with input templates (Bosselut et al., 2019). Along the same lines, one can recast the task into instruction-tuning and write the structured knowledge as an explicit task instruction (Ouyang et al., 2022; Chung et al., 2022; Sanh et al., 2022). To efficiently fine-tune a pre-trained large model, methods such as (low-rank) adaption and quantization can reduce the computation resource requirements for tuning such a model (Hu et al., 2022; Pfeiffer et al., 2020; Dettmers et al., 2022, 2023).

Fine-tuning a model requires access to its base weights. Prompting techniques come into play to improve the model behavior of proprietary, closed-source models. Lately, there has been a surge in studies focused on prompting (Arora et al., 2022; Singh et al., 2023; Wei et al., 2022; Yao et al., 2023a; Fernando et al., 2023); one of the more noticeable methods involves integrating reasoning and actions through external tools (Yao et al., 2023b), such as lexical resources, allowing the model to access external databases. The retrieved data will be added to the prompt and augment the model's generation (retrieval-augmented generation, Lewis et al., 2020). Even without updating model parameters, this in-context learning during prompting resembles implicit gradient descent on the model's parameters (Dai et al., 2023; Von Oswald et al., 2023).

2.2. LLM as a compressor

The strong prediction capability of LLMs positions them to be strong compressors. The relationship between predictors and compressors has long been established, and the underlying mechanisms are described as "two sides of the same coin" (Deletang et al., 2024; MacKay, 2003). The intrinsic connection is best characterized by Shannon (1948)'s source coding theorem, in which the optimal code length of a compressed message is closely related to the entropy of the input data. In this vein, the language model's compression capability stems from the model's ability to identify regularities among input tokens, which allows the model to maximize the predicted likelihood of the next token thereby reducing the entropy of the input sequence.

Viewing an LLM as a compressor goes beyond producing optimal code. Following Ryskin and Nieuwland (2023), underlying this prediction or compression process reflects the internal constraints learned by the model during training, which guide the prediction of the next token. Furthermore, the predicted likelihoods are directly linked to notions of surprisal or cloze probability in psycholinguistics literature (Kutas and Hillyard, 1984; Levy, 2008). The compressed code length thus offers a theoretically driven method to summarise the predicted likelihoods of each token of the input sequence into a simple measure.

3. Compression Advantage

In this section, we show that LLMs indeed act as a compressor for semantic pairs. We first introduce the arithmetic encoder, with which the predicted probabilities from LLMs are encoded into compressed messages. Next, we demonstrate that these compressed messages, after controlling for the sequence length, are always shorter for the correct semantic pairs than the incorrect ones. This pattern remains stable across different sizes of LLMs and is modulated by the model's training iterations over time.

3.1. Arithmetic encoding

The arithmetic encoding is depicted in Figure 2. An arithmetic encoder is composed of two parts: (1) a statistical coder that assigns a bit sequence (a codeword) for individual tokens and (2) a probabilistic model that estimates the token probability at each point of coding (Howard and Vitter, 1994). Arithmetic coding, as a statistical coder, is known to produce code with almost optimal code length given the token distribution $N \cdot H(x_t)$, where N is the sequence length, and $H(x_t)$ is the entropy of the token distribution. Therefore, the encoder assumes a model supplying the token's probability distribution. Figure 2a and 2b show the effects of using different distributions to encode the same word sequence. A uniform distribution has higher entropy and results in a longer code, while the probability estimates from a language model result in a shorter one. However, the probability estimates can be *misspecified* (Figure 2c), which results in a longer code.

The model used by the arithmetic encoder only needs to provide a correct conditional probability estimate rather than reflect the true generation process. In other words, the model may compress the semantic pairs better without having any semantic-related constraints that guide the probability-generating process. Therefore, rather than only inspecting the model's compression capacity based on the produced distributions, evaluating the model's capacity for semantic tasks is also crucial. Ideally, establishing the correlation between the compression advantages and the model's semantic task performance will strengthen the argument that the model's internal constraints guiding the probability distribution are indeed linked to semantic knowledge.

3.2. Semantic relations and compression

In what follows, we first evaluate the models' completion task performance with semantic relation pairs from Princeton WordNet. Next, we use these models and an arithmetic encoder to compress the semantic pairs and compare their compression advantages.

3.2.1. Semantic pair completion

The completion task of semantic relation pairs requires the model, given the gloss, to complete either the hypernym or the holonym of a word in Princeton WordNet. We select the headwords of synsets occurring more than five times in SemCor3.0 as materials. The model is prompted to complete the question, and the textual completions are automatically parsed to extract the predicted words.

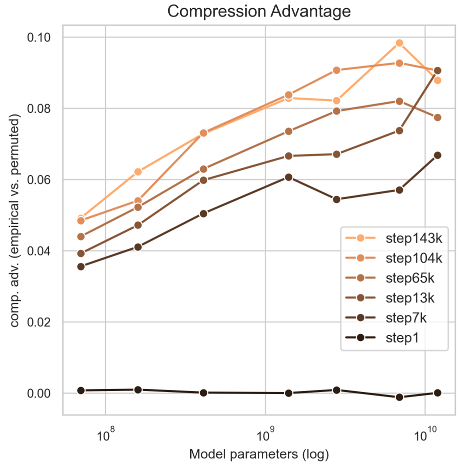


Figure 3: The compression advantage by model size and throughout training. The compression advantages consistently increase as the model increases in size and over the course of training.

	Holonym	Hypernym	
Models	Noun (N=164)	Noun (N=702)	Verb (N=583)
Pythia-12b	.12 (.02)	.51 (.01)	.28 (.01)
Pythia-6.9b	.19 (.02)	.46 (.01)	.32 (.01)
Pythia-2.8b	.15 (.02)	.42 (.01)	.25 (.01)
Pythia-1.4b	.08 (.01)	.31 (.01)	.12 (.01)
Pythia-410m	.10 (.01)	.14 (.01)	.12 (.01)
GPT-3.5	.50 (.03)	.66 (.01)	.50 (.01)
GPT-3.5-inst	.54 (.03)	.51 (.01)	.47 (.01)

Table 1: Model performances on the English semantic relation task. Scores indicate the path similarity score (the higher, the better). Numbers in parentheses are standard errors. The API version of GPT-3.5 is `gpt-3.5-turbo-0613`, GPT-3.5-inst is `gpt-3.5-turbo-instruct-0914`.

Open models, along with the proprietary ones, are selected for the current experiments. We select the Pythia model suite (Biderman et al., 2023) as they provide multiple model size and their checkpoints during the training. The proprietary models, GPT-3.5 and GPT-3.5-instruct are included for comparative purposes. We select these models as they provide both chat-based and text-completion interfaces and allow better comparisons. Nevertheless, we expect other closed-source commercial models will have consistent patterns of results. These closed models do not provide complete logits required for arithmetic encoders but nevertheless provide an idea of how well competitive LLMs can perform in the task.

Table 1 presents the results. Numbers in the tables are path similarity of the predicted and target words in Princeton WordNet. The scores range from 0, indicating no connecting path in WordNet, to 1 (exact match). Instances where the model merely repeats the test words are assigned a zero score. Three observations are noteworthy. First, the model performance generally correlates with the model size, i.e., the larger model is better. Second, the models perform better in hypernym completions than holonyms, and nominal hypernyms are better than verbal ones. The pattern is consistent across model sizes, and it is reasonable because it may reflect the task difficulties between lexical categories but is also consistent with the hierarchical structure differences among different relation types. Thirdly, the proprietary models have consistent patterns, although they do have higher scores across categories.

These findings pave the way to a more detailed analysis of the modes' compression capacities. Although open models are not as competitive as the GPTs, the consistent trend of model sizes shows the extent to which these models capture semantic relations is different. The interesting question is whether the task performances are indeed correlated with the compression advantages of these models. The following experiment explores this hypothesis.

3.2.2. Compression advantages of semantic relations

Having established that the models of different sizes have different performances on semantic completion tasks, we now turn to whether the models' performances consistently reflect on their compression advantages.

We first define the compression ratio (CR) of a given sequence $\mathbf{X}$ of length N as follows,

$$\mathrm{CR} = \frac{\|\mathrm{ArithEnc}(p_{\mathrm{LLM}}(\mathbf{X}))\|}{N \cdot H_{\mathrm{unif}}(x)}$$

where ArithEnc stands for arithmetic encoder used to generate a compressed message, and $\|\cdot\|$ is the message length (in bits). $p_{\mathsf{LLM}}(\mathbf{X})$ indicates the conditional probability distribution of each token, $H_{\mathrm{unif}}(x)$ is the entropy for each token given a uniform distribution. The compression advantage is in turn defined as the difference in CRs between empirical and permuted sequences:

$$\mathrm{CompAdv} = \mathrm{CR}_{\mathrm{perm}} - \mathrm{CR}_{\mathrm{emp}}$$

The empirical sequences have the correct semantic pairing, which includes a question part, the definition of synset and its headword, and a response part, the definition of the hypernym/holonym synset and its headword. The permuted sequence has the same format, only the question part is replaced by another random question part in the dataset (see Figure 1 for an example).

We compare the compression advantage of the response part of each sequence pair. Crucially, the response text in the empirical/permuted pair is the same, only the preceding context is different. This way, any resulting compression advantage of the response text must come from the pairing itself. Therefore, if the model could discriminate the empirical and permuted sequences of semantic pairs, the compression ratio should be different. Specifically, as the empirical one follows the semantic constraints potentially learned from the training text, the model should find it more compressible, resulting in a shorter compressed message. When compared to the permuted sequences, the compression advantage should be larger. Furthermore, this trend of advantage should correspond to the models' semantic task performance: the larger the model, the higher the compression advantages.

Figure 3 shows the results. Consistent with the hypothesis, the compression advantages generally correlate with the model size. The advantage appears to plateau for models larger than 6.9b, which is also observed from Table 1. These results suggest that the model encodes the structured knowledge as a form of internal constraints of what would follow in the text. The larger the model, the learned constraints are more robust, reflecting better semantic task performances and higher compression advantage.

What's more interesting in Figure 3 is compression advantages improve not only with model sizes but also with the training steps. It hints that the data volume the model has seen matters: either mere exposure to a large enough amount of data enables the model to learn the constraints, or, in the training materials, there are structured text patterns that explicitly describe the semantic relations.

In the next section, we explore the factor of the model's seen data. We use another language, i.e., Traditional Chinese, to examine whether the compression advantage would be larger when we explicitly introduce semantic relations to the model. The objectives are twofold: firstly, to replicate the findings of English WordNet in Chinese Wordnet, and secondly, to assess whether direct fine-tuning of a model with texts that explicitly describe semantic relations leads to higher compression advantages for semantic pairs.

4. Lexical Resource and Compression

This section examines whether the introduction of structured knowledge affects the compression advantages. In the previous section, we showed that the more data the model has seen (further into the

	CWN				**MOE Dictionary**			
	BertScore F1		SBERT		BertScore F1		SBERT	
Model	Emp	Perm	Emp	Perm	Emp	Perm	Emp	Perm
LopeLlama	.910	.848	.737	.415	.888	.866	.586	.275
Taiwan-LLaMa	.792	.770	.361	.170	.851	.832	.536	.229
Difference	.118	.078	.376	.245	.037	.034	.050	.046

Table 2: The evaluation of LopeLlama and Taiwan-LLaMa's task performance. We use BERTScore and SBERT to evaluate the output of LopeLlama and Taiwan-LLaMa on 500 CWN and 100 MoeDict unseen instances. Crucially, the differences between LopeLlama and Taiwan-LLaMa in empirical conditions are always higher than the permuted ones.

training process), the larger the compression advantages. The question remains whether explicit introduction of structured knowledge, in a relatively small-amount, also improves the compression, and how the improvement could generalize to different tasks. To investigate this, we fine-tune a new model, `LopeLlama` based on `TaiwanLlama` by explicitly introducing the lexical knowledge from the Chinese Wordnet. We first build and evaluate the fine-tuned model in Section 4.1 and compare the compression advantages of the fine-tuned and based model on three different tasks in Section 4.2.2.

4.1. Fine-tuned model: LopeLlama

4.1.1. Training

We fine-tune LopeLlama on top of Taiwan-LLaMa (Lin and Chen, 2023), which was pre-trained on over 5 billion tokens of Traditional Chinese. The model was further fine-tuned on over 490K multi-turn conversational data to enable instruction-following and context-aware responses.

We train LopeLlama with Chinese Wordnet (CWN), a lexical resource of traditional Chinese. CWN has 29,619 senses, of which 26,657 are used for training, and 2,962 are left for testing. Each sense has a definition or semantic relations. We use these attributes to generate an instruction dataset with the following generation tasks: semantic relation, definition, example sentences, synonyms, hypernyms, and hyponyms (the details of each task are shown in supplementary). For sequences that are too long for the model's context size, we split them into sets of ten. Therefore, a task involving a given sense may be spread across several training examples. After preprocessing, we have 101,483 training examples.

LopeLlama is trained from the base model `TaiwanLlama` [2] with LLaMa Factory (hiyouga, 2023). The fine-tuning is configured to use QLoRA (Hu et al., 2021; Dettmers et al., 2023) of 4-bit quantization and FlashAttention-2(Dao, 2023). The model is trained with 3 epochs, learning rate 4e-4 with cosine scheduling, and the LoRA rank is 16. Complete training parameters can be found in the supplementary materials. The training was completed in about four days on a single RTX A5000.

4.1.2. Performance evaluation

We use automatic evaluation and qualitative case studies to verify that the fine-tuned model has a better performance on the semantic tasks.

To automatically evaluate the output of the fine-tuned `LopeLlama`, we use BERTScore (Zhang et al., 2020) and SBERT (Reimers and Gurevych, 2020)[3], along with the baseline performance of the base model. BERTScore compares the sequence pairs based on token similarity; it calculates the cosine similarities of the most similar token pairs among the reference and candidate sentences. By contrast, SBERT works on the sentence level; it is fine-tuned such that produced sentence embeddings are semantically meaningful and can be compared using cosine-similarity.

Table 2 shows the scores of both fine-tuned LopeLlama and the base model Taiwan-LLaMa. The evaluation results are based on the evaluation split which contains 500 instances. Considering the instruction dataset always follows a predefined template, the differences in BertScore or SBERT may result from the model learning superficial sentential structures. Therefore, we provide a permutation baseline, which permuted the pairing between the instruction prompts and the responses' ground truths in each instance. That is, in permutation sequences, the model's generation will be compared to an incorrect ground truth which nevertheless follows the same response template. This way, the scores in permutation sequences serve as the baseline resulting from superficial sentential structures.

[2] https://huggingface.co/yentinglin/Taiwan-LLaMa-v1.0

[3] The BERTScore model used is distilbert-base-multilingual-cased, and SBERT is paraphrase-multilingual-MiniLM-L12-v2

Word	Generation Samples	
弦 *xián*	CWN	繫在弓上下兩端的線 ‘String tied to the upper and lower ends of the bow.’
	LopeLlama	弓的兩端繫上的線 ‘String tied to both ends of the bow.’
	TwLlama	弦是一種由絲、絹或其他材料製成的細長而敏感的樂器 ‘A stringed instrument made from silk, satin, or other materials, which is slender and sensitive.’
雙向 *shuāng xiàng*	CWN	形容事件進行的方向有兩個 ‘Describing an event with two directions.’
	LopeLlama	形容兩方同時進行 ‘Describing two sides happening simultaneously.’
	TwLlama	雙向：以兩種語言或方向進行溝通 ‘Bilingual: Communicating in two languages or directions.’
獸 *shòu*	MOE	總稱有四足而全身有毛的脊椎動物 ‘General term for vertebrate animals with four legs and fur.’
	LopeLlama	四隻腳、全身有毛的脊椎動物的總稱 ‘Vertebrate animals with four legs and fur, general term’
	TwLlama	與動物有關的事物 ‘Things related to animals.’
岌 *jí*	MOE	山勢高峻 ‘High and steep mountain terrain.’
	LopeLlama	形容山形陡峭，高低落差大的 ‘Describes steep mountain shapes with large elevation changes.’
	TwLlama	岌（yǎn）：形容詞。指小山丘。例句：1.《爾雅・釋山》：「小山岌，大山峘。」2. 如：「岌峨」、「岌嶪」 ‘(yǎn): Adjective. Refers to small hills. Example sentences: 1. "Erya・Explaining Mountains": Small hills are 岌, and big mountains are 峘. 2. Such as "岌峨" and "岌嶪"’

Table 3: LopeLlama and Taiwan-LLaMa sample outputs in definition generation task, given the headword, POS, and example sentences of words from CWN or MOE.

Furthermore, it can be argued that the fine-tuned model only learn the writing style (e.g. lexical choice or collocation patterns) instead of underlying semantics. To address the concerns, we add an additional 100 instances from the MOE dictionary, which have different writing styles in definitions and example sentences, are included. Comparing two models on these instances ensure any sentence similarity cannot be attributed to the surface features. The results show that in all comparisons, LopeLlama always perform better than Taiwan-LLaMA, as seen the empirical differences are always larger than the permuted ones. The differences in MOE Dictionary is indeed smaller, suggesting the fine-tuned model is strongly influenced by the response format. Nevertheless, the findings suggest that the fine-tuned model performs better in the semantic tasks.

In addition to quantitative evaluations, we further manually examine 500 text generation in the test splits, with greedy decoding. Generation samples are shown in Table 3. For instance, in case #1. 弦, LopeLlama accurately describes it with “tied at both ends,” while Taiwan-LLaMa’s response is mixed with definitions of instruments and silk materials. Also, in #2. 雙向, where LopeLlama’s generation is similar to the CWN ground truth, while Taiwan-LLaMa’s generation is more related to ‘bilingual’. Similar cases are observed in MOE dictionary instances, such as #3. 獸. LopeLlama provides relevant features such as “vertebrate animals,” “four legs,” and “fur,” while Taiwan-LLaMa’s only provides a general description.

The automatic and manual evaluations both indicate the fine-tuned model, LopeLlama, has better task performance compared to the base model. We now proceed to examine how the compression capacities of the fine-tuned model, having been fine-tuned on the explicit semantic instruction dataset, differ from those of the original base model.

4.2. Compression advantage in the fine-tuned model

To further study the compression capacities of the fine-tuned LopeLlama model, we compare their compression advantages with three datasets.

The first dataset is the evaluation split of the LopeLlama fine-tuning dataset, which is the exact same dataset used in Table 2. The compression advantages (CAs), as computed in Section 3.2.2, are the difference in the response part’s compression ratio between empirical and permuted sequences. The CA of the fine-tuned LopeLlama is 0.115 ($SE = .0072$), and the one of the base model, TaiwanLlama, is 0.080 ($SE = .0076$). Therefore, consistent with the previous findings, models that perform better in semantic tasks also have larger CAs.

4.2.1. CWN semantic relations

The observed difference in CA might not be surprising for the following reasons. First, these sequences follow the same surface structure as the dataset used to train LopeLlama. A higher CA may result from the model learning to expect surface structures rather than the underlying semantics.

Secondly, different from the English semantic relation dataset used in 3.2.2, the empirical and permuted sequences have the same instruction part but differ in response parts. Although CA automatically controls for different sequence lengths, the sequence difference is nevertheless a confounding variable in the comparison.

To address these concerns, we introduce a second dataset, semantic relation pairs from CWN. The dataset is aimed to serve as the counterpart of the English semantic pair dataset in 3.2.2. There are 626 instances in this dataset, which are 549 hypernymys and 77 holonymys. Each sequence starts with a prompt consisting of a word, its definition, and the intended semantic relation, followed by the response part, which is the target word and its definition. As in the English dataset, the empirical and permuted sequences in a given pair shared the same response (see CWN SemRel. in 1(B)).

The CAs are computed the same way for both models, which is for 0.060 $(SE = 0.004)$ LopeLlama and 0.044 $(SE = 0.004)$ for the TaiwanLlama model. The pattern is the same as observed in the first dataset. The consistent findings suggest that the fine-tuned model captures the superficial sentential structure and learns to encode the semantic relations within the pairs better. More interestingly, the Taiwan-LLaMa is trained on 35B tokens, yet the LopeLlama is fine-tuned with less than 30M tokens. This implies that even a small amount of training data can significantly change compression capacities.

4.2.2. Character sense-disambiguation

The last question about the fine-tuned model's compression capacity is how well it generalizes the learned semantic constraints to unseen tasks. Here, we use the third dataset, which includes task sequences entirely novel for the model: a character sense-disambiguation task. This task exploits the morphological structure in Chinese bisyllabic words. These words have two characters (syllables), most of which could be used as a single-character words and have their own meanings. Thus, these bisyllabic words can also be considered compounds where each constituting single-character words contribute their own meanings, among their multiple senses, to the whole two-character compound. In this character sense-disambiguation dataset, each sequence's question part is to find the meaning of a given character in a bisyllabic word, and the response part is the character's meaning in that word.

There are 469 bisyllabic words in this character sense-disambiguation dataset. These words are selected from CWN, and their constituting characters must also have 5 to 10 senses when used as single-character words. The dataset is automatically generated by an independently developed system that leverages the LangChain framework (Chase) and the GPT-3.5 model (Tom Brown et al., 2020) that has access to CWN database through retrieval-augmented generation (further details of this system, LopeGPT, can be found in Supplementary). It should be noted that identifying the character's meaning in a bisyllabic word is a controversial linguistic topic (Packard, 2000). Therefore, this dataset only serves as a medium to study the compression capacities of the model rather than a normative linguistic analysis of Chinese morphology. The dataset includes empirical and permuted sequence pairs, where the question parts are different, and the response parts are the same in a given sequence pair.

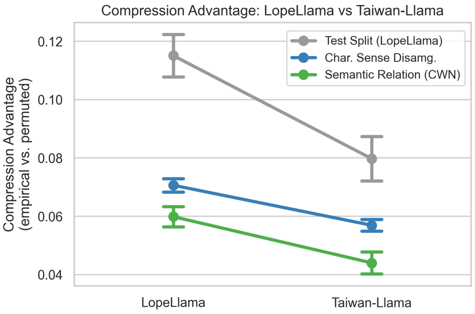


Figure 4: Compression advantages of LopeLlama and Taiwan-LLaMa on three different tasks. LopeLlama shows consistent compression advantages over the base model Taiwan-LLaMa across different datasets. Error bars indicate one standard error.

Interestingly, the same CAs patterns are observed, which are .071 $(SE = .002)$ for LopeLlama and .057 $(SE = .002)$ for TaiwanLLaMa, which indicates the fine-tuned model's compression capacities generalize to the unseen task (CAs results of all three datasets are shown in Figure 4). Crucially, the sequences in this dataset are generated by another model that only has access to CWN through retrieval augmentation. Better CAs in the fine-tuned model than in the base model imply that the fine-tuned model learns abstract semantic constraints underlying CWN. In summary, the findings from the three datasets all indicate that the model's fine-tuning process modulates its semantic compression capacities.

5. Conclusion

This paper offers an angle of seeing LLMs as strong compressors from the information-theoretic compression viewpoint, which is motivated both by the machine learning study on information theory and psycholinguistics theory on prediction mechanism (Juola, 2008; Deletang et al., 2024; Ryskin and Nieuwland, 2023). Along this line, we conduct a series of experiments on the semantic relations from English and Chinese Wordnet, empirically demonstrating that LLMs can indeed compress semantic relations better measured by the proposed compression advantages index, and the compression capacities are consistent with the model's performance on semantic tasks. Moreover, by fine-tuning a new model with a small semantic relation dataset, the compression advantages improve, even in the unseen task. Performance-wise, these results are not surprising given LLMs are competent in natural language processing tasks(Qin et al., 2023); yet, the compression angle shed light on the model performance in a more functional way: as the source coding theorem suggests, predicting and compression are the two sides of the same coin. This paper empirically provides evidence that an LLM can be viewed as a compressor of semantic information or potentially other structured knowledge, where the model learns the text input's underlying constraints, helping it maximize the predictive probabilities.

The compression angle offers a high-level computational viewpoint to LLMs and the semantic relations, yet it does not deal with the algorithmic and representational problem (Marr, 1982): how the model represents the constraints guiding the compression. This question will require further work inspecting the model's states such as contextualized embeddings, circuits, and specific nodes(Prakash et al., 2024; Ghandeharioun et al., 2024; Wang et al., 2023), and how they interact with compression. These studies will help us better understand how LLMs learn and encode structured knowledge.

6. Acknowledgements

The authors gratefully acknowledge the support of ERC Horizon Europe Programme, ERC-2021-ADG, No. 101054902. The work was also partially supported by Chiang Ching-Kuo Foundation for International Scholarly Exchange, Taiwan, No. RG011-D-20.

7. Bibliographical References

Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask me anything: A simple strategy for prompting language models.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

H Chase. Langchain.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Pedro Colon-Hernandez, Catherine Havasi, Jason Alonso, Matthew Huggins, and Cynthia Breazeal. 2021. Combining pre-trained language models and structured knowledge.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

Tri Dao. 2023. FlashAttention-2: Faster attention with better parallelism and work partitioning.

Gregoire Deletang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. 2024. Language modeling is compression. In *The Twelfth International Conference on Learning Representations*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs.

Katharina Ehret. 2018. Kolmogorov complexity as a universal measure of language complexity. *Proceedings of the First Shared Task on Measuring Language Complexity*, pages 8–14.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models.

hiyouga. 2023. Llama factory. `https://github.com/hiyouga/LLaMA-Factory`.

Paul G. Howard and Jeffrey Scott Vitter. 1994. Arithmetic coding for data compression. *Proc. IEEE*, 82:857–865.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank adaptation of large language models.

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 *International Conference on Learning Representations*.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–19.

Patrick Juola. 1998. Measuring linguistic complexity: The morphological tier. *Journal of Quantitative Linguistics*, 5(3):206–213.

Patrick Juola. 2008. Assessing linguistic complexity. *Language Complexity: Typology, Contact, Change. John Benjamins Press, Amsterdam, Netherlands*.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Marta Kutas and Steven A. Hillyard. 1984. Brain potentials during reading reflect word expectancy and semantic association. *Nature*, 307(5947):161–163.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Yen-Ting Lin and Yun-Nung Chen. 2023. Language models for taiwanese culture. Code and models available at https://github.com/MiuLab/Taiwan-LLaMa.

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7(1):195–212.

Linhao Luo, Thuy-Trang Vu, Dinh Phung, and Gholamreza Haffari. 2023. Systematic assessment of factual knowledge in large language models. *arXiv preprint arXiv:2310.11638*.

David JC MacKay. 2003. *Information theory, inference and learning algorithms*. Cambridge university press.

David Marr. 1982. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.

OpenAI. text-embedding-ada-002 [embedding model].

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jerome L Packard. 2000. *The morphology of Chinese: A linguistic and cognitive approach*. Cambridge University Press.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *The Twelfth International Conference on Learning Representations*.

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*, pages 1339–1384, Singapore. 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.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Rachel Ryskin and Mante S. Nieuwland. 2023. Prediction during language comprehension: what is next? *Trends in Cognitive Sciences*, 27(11):1032–1052.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Chandan Singh, John X. Morris, Jyoti Aneja, Alexander M. Rush, and Jianfeng Gao. 2023. Explaining patterns in data with language models via interpretable autoprompting.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ilaria Tiddi and Stefan Schlobach. 2022. Knowledge graphs as tools for explainable machine learning: A survey. *Artificial Intelligence*, 302:103627.

Nick Ryder Melanie Subbiah Tom Brown, Benjamin Mann, 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.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, Honolulu, Hawaii, USA. JMLR.org.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Shuohuan Wang, Yu Sun, Yang Xiang, Zhihua Wu, Siyu Ding, Weibao Gong, Shikun Feng, Junyuan

Shang, Yanbin Zhao, Chao Pang, Jiaxiang Liu, Xuyi Chen, Yuxiang Lu, Weixin Liu, Xi Wang, Yangfan Bai, Qiuliang Chen, Li Zhao, Shiyong Li, Peng Sun, Dianhai Yu, Yanjun Ma, Hao Tian, Hua Wu, Tian Wu, Wei Zeng, Ge Li, Wen Gao, and Haifeng Wang. 2021b. Ernie 3.0 titan: Exploring larger-scale knowledge enhanced pretraining for language understanding and generation.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Qihuang Zhong, Liang Ding, Yibing Zhan, Yu Qiao, Yonggang Wen, Li Shen, Juhua Liu, Baosheng Yu, Bo Du, Yixin Chen, Xinbo Gao, Chunyan Miao, Xiaoou Tang, and Dacheng Tao. 2022. Toward efficient language model pretraining and downstream adaptation via self-evolution: A case study on superglue.

A. LopeLlama: Training Hyperparameters

Table A1 are the hyperparameters of low-rank adaptation when training LopeLlama on the base model.

Hyperparameter	Value
batch_size	4
gradient_accumulation_steps	8
lr_scheduler_type	cosine
learning_rate	4e-4
num_train_epochs	3
fp16	True
quantization_bit	4
lora_rank	16
lora_alpha	16
lora_dropout	0.05
flash_attn	True

Table A1: Training arguments for LopeLlama. All other parameters were set to the default value.

B. Training data for LopeLlama

See Table A2 for the training data and their formats of LopeLlama instruction fine-tuning.

C. Individual scores of LopeLlama on CWN tasks

Table A3 shows the performances of LopeLlama on the individual tasks based on Chinese Wordnet.

D. LopeGPT

LopeGPT is built as a chatbot service leveraging the LangChain framework (Chase) and the GPT-3.5 model (Tom Brown et al., 2020) and integrating language resources to enhance its language understanding and providing more effective, contextually relevant responses. In addition to the character disambiguation tasks used in the current study, LopeGPT offers more functions and helps users accomplish tasks regarding lexical semantics and corpus linguistics. The integrated resources are listed as follows:

CWN. It serves as a language knowledge resource focusing on word senses and semantic relations in Taiwan Mandarin. This wordnet includes over 29,000 senses derived from over 29,000 lemmas, as well as over 12,000 synsets and over 59,000 semantic relations.[4] As we manage to integrate lexical knowledge into LopeGPT, the word sense tagger is also added as an external resource.

Corpus data in Taiwan. The data derives from two resources: (1) Academia Sinica Balanced Corpus of Modern Chinese (ASBC), which includes 19,247 texts, 11M word tokens and 239K word types. (2) Social Media Corpus in Taiwan (SoMe), which collects articles and comments from PTT[5], a BBS (Bulletin Board System) with more than 15 million users in Taiwan. There are 70K posts, along with 3M comments, ranging from 2020 to 2023, extracted from SoMe. The posts have been preprocessed and embedded via the `text-embedding-ada-002` model (OpenAI).

These resources are built into external tools and made available to LopeGPT. Therefore, LopeGPT can capitalize on the aforementioned language resources for lexical semantic tasks. We conducted a series of experiments to assess LopeGPT's capacity for sense identification (for a single character in a bi-syllabic word), semantic relation identification (for a neologism), lemmatizing and POS-tagging sentences, and sense disambiguation (see supplementary for details). The preliminary results demonstrate LopeGPT's proficiency in comprehending word and character meanings in terms of the evaluation tasks. In other words, language resources such as corpora and WordNet significantly enhance LLMs' language comprehension and performance across various natural language processing tasks.

LopeGPT access to the external linguistic resources by the use of tools (as formulated by `langchain`, Chase). These tools are defined as follows:

- *SenseTagTool(text):* Tokenizes and tags text using DistilTagger from the CWN to provide rich contextual information for further processing.
- *QuerySenseFromDefinitionTool(text):* Returns all senses that contain the given text in their definitions. The text can be specified using regular expressions for flexibility.
- *QuerySenseFromLemmaTool(text):* Returns all senses that contain the given text in their lemmas (i.e., the basic form of a word). Like other tools, it also supports regular expression-based text input.
- *QuerySenseFromExampleTool(text):* Returns all senses that contain the given text in their examples. This tool allows for context-based sense querying.
- *QueryAsbcSenseFrequencyTool(sense_id):* Provides the frequency of a particular sense_id in the ASBC, offering insights into the usage and prominence of specific senses.

[4] Each sense includes its definition, example sentences, part-of-speech, and semantic relations.

[5] http://www.ptt.cc/bbs/index.html

Task	Given	Want	#	%
Relations	HW, POS, DEF	REL	28,042	27.6
Definition	HW, POS, SENT	DEF	26,657	26.3
Representative Sentence	HW, POS, DEF	SENT	25,173	24.8
Synonyms	HW, POS, DEF, SENT	SYN	9,863	9.7
PWN Synset	HW, POS, DEF	PWN	7,568	7.5
Hypernyms	HW, POS, DEF	HYPER	3,071	3.0
Hyponyms	HW, POS, DEF	HYPO	1,023	1.0
Supplementary	HW, POS, DEF, SENT	SUPP	86	0.1
Total			**101,483**	**100**

Table A2: Training data for LopeLlama. Several tasks are generated for each sense that represents a specific aspect of that sense. “Given” indicates what information is provided to the model. “Requested” is what the model should generate. **HW**: headword, **POS**: part of speech, **DEF**: definition, **REL**: relation, **SENT**: example sentence, **HYPER**: hypernym, **HYPO**: hyponym, **PWN**: Princeton WordNet Synset, **SUPP**: supplementary

.

Model	Task	#	BS F1 (Perm.)	BS P (Perm.)	BS R (Perm.)	SBERT (Perm.)
LopeLlama	REL	141	0.9456 (0.8624)	0.9551 (0.8711)	0.9368 (0.8553)	0.8858 (0.6203)
Taiwan-LLaMa			0.7718 (0.7339)	0.8232 (0.7695)	0.7287 (0.7038)	0.3967 (0.2326)
LopeLlama	DEF	137	0.9243 (0.8572)	0.9263 (0.8588)	0.9228 (0.8566)	0.7460 (0.2660)
Taiwan-LLaMa			0.8393 (0.8200)	0.8347 (0.8176)	0.8444 (0.8231)	0.4484 (0.1835)
LopeLlama	SENT	119	0.8157 (0.7810)	0.8303 (0.7888)	0.8024 (0.7743)	0.4358 (0.2320)
Taiwan-LLaMa			0.7975 (0.7843)	0.8306 (0.8157)	0.7673 (0.7557)	0.2645 (0.1019)
LopeLlama	SYN	47	0.9506 (0.8536)	0.9520 (0.8574)	0.9493 (0.8507)	0.9071 (0.3950)
Taiwan-LLaMa			0.7594 (0.7335)	0.7899 (0.7542)	0.7329 (0.7164)	0.3689 (0.1552)
LopeLlama	PWN	41	0.9642 (0.9473)	0.9687 (0.9512)	0.9600 (0.9436)	0.8757 (0.7451)
Taiwan-LLaMa			0.7244 (0.7242)	0.7256 (0.7250)	0.7247 (0.7250)	0.2227 (0.1416)
LopeLlama	HYPER	9	0.9516 (0.8993)	0.9455 (0.8957)	0.9579 (0.9035)	0.7996 (0.4902)
Taiwan-LLaMa			0.7935 (0.7840)	0.8243 (0.8097)	0.7657 (0.7609)	0.3251 (0.1463)
LopeLlama	HYPO	6	0.8493 (0.8171)	0.8667 (0.8316)	0.8329 (0.8037)	0.6025 (0.3916)
Taiwan-LLaMa			0.7726 (0.7613)	0.8278 (0.8143)	0.7250 (0.7154)	0.3664 (0.1129)

Table A3: Individual scores for each task on Chinese WordNet. We use BERTScore (**BS**) and **SBERT** to evaluate the output of LopeLlama and Taiwan-LLaMa across Chinese WordNet and MoeDict. Permuted (**Perm.**) means that the reference answer in each prediction is compared against is randomly shuffled within each task (e.g., tasks that generate a definition have references shuffled within that group). BERTScore calculates precision, recall and F1 while SBERT calculates cosine similarity. **#** = Number of samples for task, **P** = Precision, **R** = Recall. **HW**: headword, **POS**: part of speech, **DEF**: definition, **REL**: relation, **SENT**: example sentence, **HYPER**: hypernym, **HYPO**: hyponym, **PWN**: Princeton WordNet Synset, **SUPP**: supplementary

- *QueryRelationsFromSenseIdTool(sense_id):* Returns all relations associated with a given sense_id, enabling exploration of semantic connections and relations.

- *QueryAsbcFullTextTool(text):* Enables searching the ASBC and returns the first 50 lines containing the specified text, facilitating access to relevant textual contexts.

- *QueryPTTSearchTool(text):* Converts the input text into vectors and performs similarity-based retrieval to find the top 10 articles most closely related to the query. This tool aids in retrieving contextually relevant information from online sources.