LREC-COLING 2024

The First Workshop on Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @LREC-COLING-2024 (NeusymBridge 2024)

Workshop Proceedings

Editors
Tiansi Dong, Erhard Hinrichs, Zhen Han, Kang Liu, Yangqiu Song, Yixin Cao, Christian F. Hempelmann, Rafet Sifa

21 May, 2024
Torino, Italia

Proceedings of NeusymBridge 2024: The First Workshop on Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ LREC-Coling 2024

ISBN 978-2-493814-12-8
ISSN 2951-2093 (COLING); 2522-2686 (LREC)

Jointly organized by the ELRA Language Resources Association
and the International Committee on Computational Linguistics

Message from the Workshop Organizers

Endowing machines with knowledge has long been regarded as one of the important goals of AI. Traditionally, symbols and their relations represent knowledge for natural language processing. Obviously, because of the limitations of the classical symbolic-based knowledge representation theory and knowledge acquisition technologies, symbolic knowledge bases have typical weaknesses, such as limited representation capacity, low acquisition efficiency, low coverage of multiple knowledge types, and applicable difficulties in reasoning scenarios. By contrast, large language models (LLMs) follow quite a different paradigm: the tradition of connectionism and neural networks. It employs distributional numerical vectors/matrices to represent the knowledge. This way, almost all knowledge types can be represented and embodied into a unified semantic space.

LLMs can be treated as knowledge base and provide an easier way to acquire and collect knowledge and inject knowledge into the downstream models or applications. However, compared with traditional symbolic knowledge bases, LLMs still have limitations including hallucination, and relying exclusively on fill-in-the-blank close tasks. A recent study showed that LLMs may miss more tail knowledge than head knowledge. LLMs still struggle to acquire negative knowledge. On the other hand, queries on knowledge graphs, in symbolic and/or neural ways, can vastly answer more complex logical queries, such as union, intersection, negation, counting, etc. Various strategies have been explored to improve the interpretability and reasoning performance of LLMs, for example, CoT, CoT-SC, Tree-of-Thoughts, or using external symbolic inference engines. unavailable. However, researchers still argue that LLMs are not good logical reasoners. One of the main reasons is that LLMs' reasoning is mostly non-rigorous — neither the reasoning process nor the result is guaranteed to be correct and complete. Despite these shortcomings, LLMs are becoming fundamental tools and have achieved great success in both academia and industry. They not only unify various NLP-related tasks in the form of text generation, but also have shown remarkable reasoning ability.

A cutting-edge research direction is to move from System I associative thinking to System II rational thinking – in the sense of D. Kahneman. Researchers are targeting novel machine learning systems for "slow, logical, sequential, conscious, linguistic, algorithmic, planning, and reasoning" problems. Knowledge graphs provide a natural way of connecting the dots across texts. Building an inherent linkage module for LLMs can provide a better global view of the world.

Moving from System I thinking to System II thinking demands traditional deep-learning to go beyond the statistical learning framework, and make qualitative extensions. A variety of new learning biases has been proposed to narrow the gap between higher-level cognition and traditional deep-learning. Language is embodied and schematizes space. The next generation of neural language system shall be a brain- and AI-inspired understanding system that explicitly represents situations, which roots in qualitative spatial representation, then extending to spatio-temporal and event representation, moving on to causality and emotion. Recent research proposes tensors as a unified representation for perception and memory, proposes spheres to explicitly unify symbolic structure with neural embedding for deterministic reasoning, neurosymbolic unification, and for humour understanding.

This workshop invited renowned scholars to give keynotes and active researchers to introduce their pioneering works in the fields, topics covering both academic researches and industrial applications. The state-of-the-art in deep learning for NLP and beyond shows that there are many open research questions to be addressed at the interface of symbolic and neural approaches, and that bridging neurons and symbols may break the glass ceiling of deep learning for NLP.

The NeusymBridge 2024 Organizers

Organizing Committee

- Tiansi Dong – Fraunhofer IAIS – Neurosymbolic Representation Learning Team
- Erhard Hinrichs – University of Tübingen – Department of Computational Linguistics
- Zhen Han – Amazon Inc.
- Kang Liu – Chinese Academy of Sciences – Research Group of Speech and Language Technology
- Yangqiu Song – The Hong Kong University of Science and Technology – Department of Computer Science and Engineering
- Yixin Cao – Singapore Management University – Department of Computer Science
- Christian F. Hempelmann – Texas A&M-Commerce – the Semantic Applied Linguistics and Creative Laboratory
- Rafet Sifa – University of Bonn – the Applied Machine Learning (AML) Lab

Keynotes

Table of Contents of Accepted Papers

Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ LREC-Coling 2024

Tuesday, May 21, 2024

09:00–09:30	**Welcome**
09:30–10:30	**Keynote 1: Pascale Fung – Human Value Representation in Large Language Models - Bridging the Neural and the Symbolic** Chair: Yangqiu Song
10:30–11:00	**Morning coffee break**
11:00–12:00	**Keynote 2: Juanzi Li – Neural-symbolic Programming for Explainable Knowledge-intensive Question Answering** Chair: Yixin Cao
12:00–13:00	**Paper Presentation** Chair: Tiansi Dong *Probing Large Language Models from a Human Behavioral Perspective* Xintong Wang, Xiaoyu Li, Xingshan Li and Chris Biemann *The Semantic Relations in LLMs: An Information-theoretic Compression Approach* Yu-Hsiang Tseng, Pin-Er Chen, Da-Chen Lian and Shu-Kai Hsieh
13:00–14:00	**Lunch Break**

Tuesday, May 21, 2024 (continued)

14:00–15:00 **Keynote 3: Alessandro Lenci – The Semantic Gap in LLMs and How to Bridge It**
Chair: Erhard Hinrichs

15:00–16:00 **Paper Presentation**
Chair: Yixin Cao

Word Sense Disambiguation as a Game of Neurosymbolic Darts
Tiansi Dong and Rafet Sifa

Open Event Causality Extraction by the Assistance of LLM in Task Annotation, Dataset, and Method
Kun Luo, Tong Zhou, Yubo Chen, Jun Zhao and Kang Liu

16:00–16:30 **Afternoon coffee break**

16:30–17:30 **Keynote 4: Volker Tresp – The Tensor Brain: A Unified Theory of Perception, Memory and Semantic Decoding**
Chair: Han Zhen

17:30–18:00 **Paper Presentation**
Chair: Erhard Hinrichs

The Need for Grounding in LLM-based Dialogue Systems
Kristiina Jokinen

18:00–18:30 **Conclusion**
Chair: Rafet Sifa

19:30–21:30 **Workshop Dinner (Keynote speakers, paper presenters, workshop organisers)**

Keynote@NeusymBridge 2024

Human Value Representation in Large Language Models - Bridging the Neural and the Symbolic

Pascale Fung
Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology, Hong Kong
pascale@ece.ust.hk

Abstract

The widespread application of Large Language Models (LLMs) in many different areas and fields has necessitated their explicit alignment to human values and preferences. LLMs have learned human values from their pre-training data, through Reinforcement Learning with Human Feedback (RLHF) and through other forms of value fine-tuning. Nevertheless we lack a systematic way of analyzing the scope and distribution of such human values embedded in LLMs. One can use surveys of value-relevant questions to prompt LLMs for analysis and comparison. But surveys are a form of sparse sampling. In this talk, I will present UniVar, a high dimension representation of human values trained from a value taxonomy and 8 different language models in 6 different languages representing a sampling of the world's culture. We then show the UniVar representation distributions of 4 LLMs, namely ChatGPT, Llama 2, Sola, and Yi, in English, Chinese, Japanese, Indonesian, Arabic and French, which clearly demonstrate the proximity of cultures that share similar values, such as Chinese and Japanese, or Indonesian and Arabic. This is the first time where a high dimensional neural representation has been shown to be effective in generalizing the survey based symbolic representation of human values.

Keynote@NeusymBridge 2024

Neural-symbolic Programming for Explainable Knowledge-intensive Question Answering

Juanzi Li
Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, China
lijuanzi@tsinghua.edu.cn

Abstract

Explainable knowledge-intensive QA aims at returning not only the accurate knowledge answer but also the explicit reasoning process, which can enhance the interpretability and reliability of QA systems. However, state-of-the-art large language models suffer from the notorious hallucination problem, and knowledge graph based methods, such as question semantic parsing, face generalization issues. In this talk, I will present our neural-symbolic framework for explainable knowledge-intensive QA. Specifically, I will introduce our experiences in knowledge-oriented programming, automatic program induction, and probabilistic tree-of-thought reasoning by integrating the parametric knowledge of LLMs and retrieved textual knowledge.

Keynote@NeusymBridge 2024

The Semantic Gap in LLMs and How to Bridge It

Alessandro Lenci
Dipartimento di Filologia, Letteratura e Linguistica
Università di Pisa, Italy
alessandro.lenci@unipi.it

Abstract

The unprecedented success of LLMs in carrying out linguistic interactions disguises the fact that, at closer inspection, their knowledge of meaning and inference abilities are still quite limited and different from human ones. They generate human-analogue texts, but still fall short of fully understanding them. I will refer to this as the “semantic gap” of LLMs. Some claim that this gap depends on the lack of grounding of text-only LLMs. I instead argue that the problem lies in the very type of representations these models acquire. They learn highly complex association spaces that on the other hand correspond only partially to truly semantic and inferential ones. This prompts the need to investigate the missing links to bridge the gap between LLMs as sophisticated statistical engines and full-fledged semantic agents.

Keynote@NeusymBridge 2024

The Tensor Brain: A Unified Theory of Perception, Memory and Semantic Decoding

Volker Tresp
Lehrstuhl für Datenbanksysteme und Data Mining
Ludwig-Maximilians-Universität München
80538 München Germany
volker.tresp@lmu.de

Abstract

We discuss a unified computational theory of an agent's perception and memory. In our model, both perception and memory are realized by different operational modes of the oscillating interactions between a symbolic index layer and a subsymbolic representation layer. The symbolic index layer contains indices for concepts, predicates, and episodic instances known to the agent. The index layer labels the activation pattern in the representation layer and then feeds back the embedding of that label to the representation layer. The embedding vectors are implemented as connection weights linking both layers. An index is a focal point of activity and competes with other indices. Embeddings have an integrative character: the embedding vector for a concept index integrates all that is known about that concept, and the embedding vector for an episodic index represents the world state at that instance. The subsymbolic representation layer is the main communication platform. In cognitive neuroscience, it would correspond to, what authors call, the "mental canvas" or the "global workspace" and reflects the cognitive brain state. In bottom-up mode, scene inputs activate the representation layer, which then activates the index layer. In top-down mode, an index activates the representation layer, which might subsequently activate even earlier processing layers. This last process is called the embodiment of a concept.

Probing Large Language Models from A Human Behavioral Perspective

Xintong Wang[♠], Xiaoyu Li[♡], Xingshan Li[♢], Chris Biemann[♠]
[♠]Department of Informatics, Universität Hamburg
[♡]School of Computer Science and Technology, Beijing Institute of Technology
[♢]Institute of Psychology, Chinese Academy of Sciences
{xintong.wang, chris.biemann}@uni-hamburg.de,
demo.xyli@gmail.com, lixs@psych.ac.cn

Abstract

Large Language Models (LLMs) have emerged as dominant foundational models in modern NLP. However, the understanding of their prediction processes and internal mechanisms, such as feed-forward networks (FFN) and multi-head self-attention (MHSA), remains largely unexplored. In this work, we probe LLMs from a human behavioral perspective, correlating values from LLMs with eye-tracking measures, which are widely recognized as meaningful indicators of human reading patterns. Our findings reveal that LLMs exhibit a similar prediction pattern with humans but distinct from that of Shallow Language Models (SLMs). Moreover, with the escalation of LLM layers from the middle layers, the correlation coefficients also increase in FFN and MHSA, indicating that the logits within FFN increasingly encapsulate word semantics suitable for predicting tokens from the vocabulary.

Keywords: Large Language Models, Interpretation and Understanding, Eye-Tracking, Human Behavioral

1. Introduction

Recent advancements in Large Language Models (LLMs) (Devlin et al., 2018; Radford et al., 2019; Touvron et al., 2023a,b) have showcased their superior capabilities in language understanding, generation as well as zero-shot transferring. Despite their remarkable successes, issues such as the generation of hallucinated (Rawte et al., 2023) and toxic outputs (Leong et al., 2023) have arisen, underscoring the importance of understanding the internal mechanisms and predictive behaviors of LLMs to develop models that are both powerful and reliable.

Research on LLM interpretation has emerged (Zhao et al., 2023; Wang et al., 2023), focusing on dissecting the components of *Feed-Forward Layers (FFN)* and *Multi-Head Self-Attention (MHSA)*. (Geva et al., 2022) highlighted the role of FFN in LLMs, demonstrating how tokens are promoted by utilizing logits in the late layers for word prediction from a vocabulary. (Bills et al., 2023) explored the activation of self-attention heads under varying prompts. Concurrently, cognition and psycholinguistic studies have documented various measures during human reading activities (Hollenstein et al., 2018, 2019; Cop et al., 2017; Luke and Christianson, 2018), closely paralleling the processes observed in language models (Hofmann et al., 2022). As depicted in Figure 1, the juxtaposition of *human reading patterns* and a *transformer block* illustrates the similarity in attention allocation—eye-tracking measurements for humans and FFN/MHSA values for LLMs—motivating our approach to probe LLMs from a human behavioral perspective.

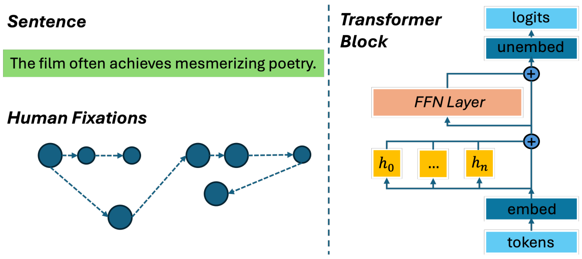


Figure 1: **Comparison of Human reading pattern and transformer block.** The left part shows the *fixation patterns* of a human reader over a given sentence, while the right part demonstrates a transformer block including *FFN layers and multi-head self-attention*. The blue dots mark fixations on the corresponding words above; a wider diameter represents a longer fixation duration.

Specifically, we investigate the **internal workings of FFN and MHSA** in LLMs, such as the GPT-2 model (Radford et al., 2019), by *correlating eye-tracking fixations with LLM values*. Our findings reveal that LLMs, particularly in their ***middle layers***, increasingly mirror human attention patterns, focusing more on essential words. However, in contrast to humans who prioritize *crucial content*, the ***upper layers*** of LLMs refine context understanding, indicating a divergence in focus on *less critical aspects*. This suggests that the outputs of FFN in the ***upper layers*** can facilitate predictions beyond just the final layers, encouraging methods for efficient semantic editing (Wang et al., 2023).

Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ LREC-Coling 2024, pages 1–7
21 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

Furthermore, our comparison of **prediction behaviors** between LLMs and Shallow Language Models (SLMs) reveals that *LLMs more closely resemble human predictive patterns*, where greater emphasis on significant words enhances the certainty of word predictions.

Our **contributions** are as follows:

- We conduct a detailed analysis of the internal mechanisms of FFN and MHSA in LLMs from a human behavioral perspective.
- We juxtapose the word prediction processes of LLMs and SLMs, reinforcing the evidence that LLMs more closely align with human attention patterns, focusing on crucial words to enhance prediction certainty.

2. Related Work

Human Behavior Measures: Studies in cognition and psycholinguistics have deployed simultaneous eye-tracking and electroencephalography during natural and task-specific reading to comprehend human reading processes. Noteworthy datasets in this context include ZuCo 1.0 (Hollenstein et al., 2018), ZuCo 2.0 (Hollenstein et al., 2019), GECO (Cop et al., 2017), and Provo (Luke and Christianson, 2018). However, to the best of our knowledge, there is a paucity of work utilizing these datasets to probe LLMs and their internal mechanisms.

Eye-movement Prediction: A shared task at ACL 2021 (Hollenstein et al., 2021) involved using language models for predicting eye-movement measures. In this shared task, models, including Boosting, MLP, and RoBERTa, displayed significant performance in this task. Besides, linguistic features proved crucial for achieving superior results (Bestgen, 2021). In this paper, we focus on employing eye-movement data for probing LLMs.

3. Preliminary

Large language models (LLMs) predominantly rely on the Transformer architecture (Vaswani et al., 2017), composed of Transformer blocks acting as layers denoted by $l = 1, 2..., L$. As shown in Figure 1, each Transformer block primarily consists of **multi-heads self-attention** and a **feed-forward network**. The motivation for the multi-head self-attention mechanism lies in its ability *to extract various aspects of the sequence, with its capacity deepening with the increase of layers*. Concurrently, the FFN serves to *output for the current layers and makes prediction over a vocabulary*.

More specifically, in layer l, the currently processed representation is denoted by X_i^l, and the output for FFN is computed as:

$$o_i^l = FFN^l\left(X_i^l\right), \tag{1}$$

where o_i^l denotes the output for the current FFN.

An updated representation $\tilde{x}_i^l$, is then achieved by adding X_i^l and o_i^l. The updated representation, $\tilde{x}_i^l$, subsequently undergoes a self-attention process. Given the presence of multi-head self-attention in each layer, all the representations in each self-attention head are concatenated to serve as the input for the subsequent FFN layer, as illustrated below:

$$X_i^{l+1} = \mathrm{concatenate}\Big(\text{Attention}^l\left(\tilde{x}_i^l\right)\Big), \tag{2}$$

In this work, we present empirical evidence understanding the function of multi-head self-attention and FFN layers by correlating their values with human behavioral data, eye-tracking measurements.

4. Eye-tracking Measurements

Human behavioral signals, such as **eye-tracking, fMRI, and EEG**, have been widely utilized in cognition and psycholinguistic studies. Among these signals, eye-tracking offers millisecond-precise recordings of gaze direction, illuminating the *focus of attention during reading and comprehension*. This process bears resemblance to the operations within a **transformer block**, as depicted in Figure 1. Thus, we employ ***eye-tracking data*** to uncover the internal mechanics of the transformer architecture.

Eye-movement Measures	**Abbrev.**	**Definition**
Gaze duration	GD	The sum of all fixations on the current word in the first-pass reading before the eye moves out of the word
Total reading time	TRT	The sum of all fixation durations on the current word, including regressions
First fixation duration	FFD	The duration of the first fixation on the prevailing word
Single fixation duration	SFD	The duration of the first and only fixation on the current word
Go-past time	GPT	The sum of all fixations prior to progressing to the right of the current word, including regressions to previous words that originated from the current word

Table 1: **Definition of Five Eye-tracking Measures**: Gaze Duration (GD), Total Reading Time (TRT), First Fixation Duration (FFD), Single Fixation Duration (SFD), and Go-Past Time (GPT).

In our study, we establish correlations between metrics derived from *multi-head self-attention (MHSA), feed-forward neural (FFN) layers*, and **five specific eye-tracking measurements**: *Gaze Duration (GD), Total Reading Time (TRT), First Fixation Duration (FFD), Single Fixation Duration (SFD), and Go-Past Time (GPT)*. Each of these metrics offers unique insights into the human reading process. For instance, Gaze Duration (GD) refers to the cumulative duration of all fixations on a given word during initial reading before moving to the next word, with *longer durations indicating the word's significance*. Similarly, Total Reading Time (TRT) encompasses all fixation durations on a word, including regressions, indicating that *readers may revisit a word multiple times to refine their understanding*. The detailed meanings of these eye-tracking measures can be found in Table 1.

By leveraging these interpretable eye-tracking metrics, we aim to probe LLMs by correlating their values with those observed in multi-head attention and FFN layers.

5. Experiments

5.1. Experimental Settings

Language Models: For our investigation, we utilized a pre-trained GPT-2 model (*base*) from HuggingFace, focusing on analyzing the internal mechanisms of FFN and multi-head self-attention mechanisms due to its **simplicity and general applicability**. We posit that our probing method is adaptable and can be extended to other, more advanced open-source LLMs such as LLaMA (Touvron et al., 2023a) and Qwen (Bai et al., 2023), among others. Additionally, we broaden our analysis to include **Shallow Language Models (SLMs)** like N-Gram language models (Pauls and Klein, 2011), Recurrent Neural Networks (RNNs), Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) networks (Sherstinsky, 2020), and a recently enhanced RNN variant, the RWKV-V4 model (Peng et al., 2023), to conduct a comprehensive comparison of prediction probabilities. For the training of SLMs, we employ the WikiText-103 dataset.

Eye-tracking Data: For human behavioral data, we utilize the ZuCo 2.0 dataset (Hollenstein et al., 2019), which contains concurrent eye-tracking records captured during two types of reading activities: *natural reading (NR) and task-specific reading (TSR)*. This dataset is notably comprehensive, comprising 730 English sentences, split into 349 sentences read under normal conditions and 390 sentences read under a task-specific paradigm. Eye-tracking data from 18 participants were recorded during both NR and TSR activities. We conducted word prediction experiments using various language models on sentences from the ZuCo 2.0 dataset to then analyze the correlation patterns between human reading behaviors and language model predictions.

Correlation Metrics and Evaluation: Following previous studies (Eberle et al., 2022) on analyzing the prediction behavior of LLMs, we also employ three prevalent correlation metrics: Pearson (Freedman et al., 2007), Spearman (Caruso and Cliff, 1997), and Kendall (Abdi, 2007), to investigate the relationship between values derived from LLMs and human behavioral measures. Despite minor differences, we find these correlation metrics yield similar results. Among them, Spearman exhibits superior robustness when compared to Pearson and Kendall. Unless stated otherwise, experimental results are reported using Spearman analysis. *Given that larger fixations, as indicated by various eye-tracking measures, signify the importance of the current word,* ***a stronger correlation implies that LLMs also allocate more attention to the processed word.***

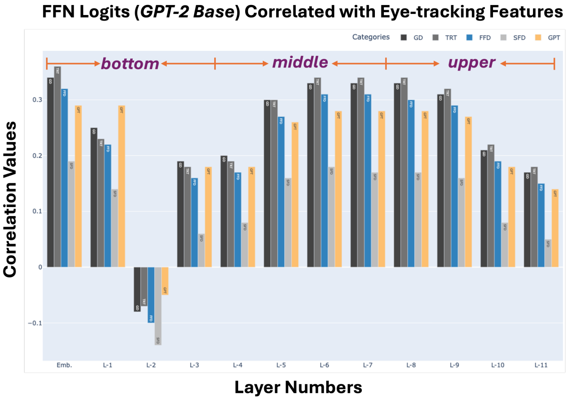


Figure 2: **FFN Correlation Values.** FFN values through layers in GPT-2 *base* Correlated with five different eye-tracking features in three groups: bottom, middle, and upper. (Significant at $p < 0.05$)

5.2. FFN Correlation Analysis

We examine the functions of the FFN within GPT-2. To elucidate our findings, we categorize the 12 layers of GPT-2 (*base*) into three groups: **bottom ($l_1 \rightarrow l_4$), middle ($l_5 \rightarrow l_8$), and upper ($l_9 \rightarrow l_{12}$).** As illustrated in Figure 2, the bottom most layers show a direct correlation between the embedding of input tokens and human reading fixations. This suggests that humans require more time to comprehend critical tokens that are also reflected in the embeddings of LLMs. This correlation diminishes as we ascend through the layers, with the topmost layer of the bottom group (Layer 3) indicating a divergence in processing tokens from human behavior; the FFN at this level begins to process tokens yet in a manner distinct from human reading patterns.

Progressing to the middle layers, the correlation coefficients initially increase and then stabilize, peaking at Layer 6. This pattern suggests that the FFN in these middle layers starts to show similar human fixation behaviors, indicating that the logits within FFN increasingly encapsulate word semantics suitable for predicting tokens from the vocabulary.

Intriguingly, in the upper layers, we observe a decline in correlation values. We hypothesize that at this stage, the LLM begins to incorporate less critical words within sentences into its considera-

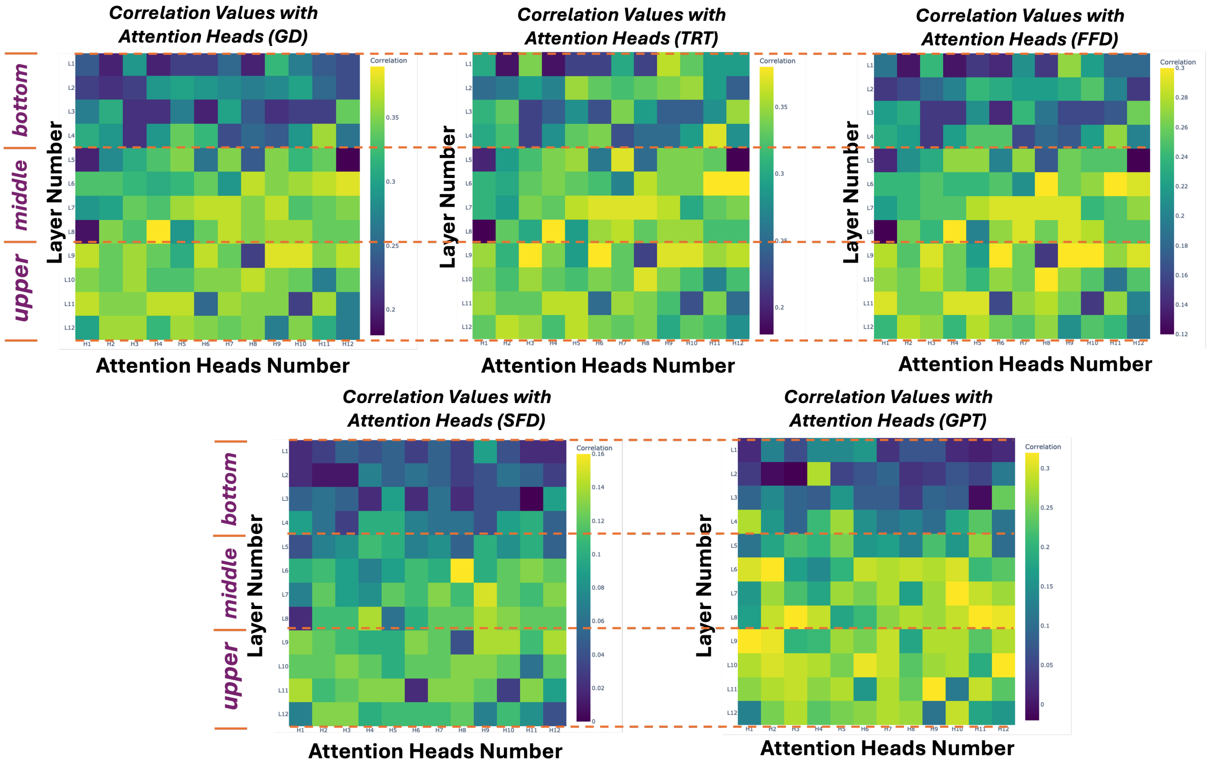


Figure 3: **Attention Heads Correlated Values with Eye-tracking Measurements through Layers Results.** ***Lighter and larger values*** signify stronger correlations.

tion, diverging from human intuition, which tends to focus on the most crucial aspects of the context and disregard less important information.

5.3. Multi-head Self-attention Correlation

Figure 3 presents heatmaps that illustrate the correlation between the values of 12 self-attention heads across 12 distinct layers and human behavioral data; where lighter and larger values signify stronger correlations. Similar to our FFN analysis, we categorized the 12 layers into three groups: **bottom, middle, and upper**. The bottom group exhibits a weaker correlation with human fixations, indicating that while self-attention mechanisms begin to process tokens at this stage, they do so differently from human behavior.

As we ascend through the middle and upper groups, we observe an increase in correlation across different layers and attention heads with human fixations. This pattern suggests that, in these layers, LLMs begin to align more closely with human patterns, especially in focusing on important contextual tokens. Notably, unlike in the FFN analysis, we did not observe a decrease in multi-head attention correlation values in the upper layers. This difference implies that the comprehension capabilities of LLMs are progressively refined up to the final layer, enabling more diverse and accurate word predictions compared to human reading patterns.

Furthermore, among the five eye-tracking measures analyzed, Gaze Duration (GD), Total Reading Time (TRT), First Fixation Duration (FFD), and Go-Past Time (GPT) demonstrate stronger correlations, whereas Single Fixation Duration (SFD) shows a weaker correlation. Given that SFD represents the first and only fixation on a current word—suggesting lesser importance—while GD, TRT, FFD, and GPT include regressions on significant words, this discrepancy explains why LLMs also prioritize these important words.

5.4. Prediction Probability Correlation

We further analyze word prediction probability behaviorals in LLMs and our investigation into the correlation of word prediction probabilities reveals distinct behaviors between Large Language Models (LLMs) and Shallow Language Models (SLMs). For this analysis, we employed two reading tasks: task-specific reading (TSR) and natural reading (NR). The TSR task encompassed 5335 words for prediction analysis, while the NR task included 5329 words. Our findings, detailed in Table 2, are divided into two parts: the upper section presents the correlation outcomes for the TSR task, and the lower section for the NR task.

Overall, SLMs exhibit a notable and consistent **negative correlation** in both the TSR and NR tasks. This trend suggests that SLMs tend to assign higher prediction probabilities with fewer fixations on critical words, thereby increasing the uncertainty of word predictions. In contrast, LLMs, exemplified by GPT-2, demonstrate a significant and **positive correlation** in both tasks. This positive correlation indicates that LLMs exhibit a prediction pattern akin to human behavior, where increased attention to crucial words leads to more confident predictions.

Though the aforementioned conclusions are consistent for both the TSR and NR tasks, it is noteworthy that the correlation values for the NR task are consistently higher than those for the TSR task. We hypothesize that during task-specific readings, humans are guided by specific clues to identify and concentrate on words that are pertinent to the task at hand. Consequently, our word prediction analysis across different LMs aligns more closely with the process in NR.

Model	**Eye-tracking Measures**				
	GD	TRT	FFD	SFD	GPT
			Task-specific Reading		
N-Gram	-0.26	-0.25	-0.23	-0.15	-0.23
RNN	-0.44	-0.43	-0.41	-0.28	-0.40
GRU	-0.46	-0.45	-0.43	-0.30	-0.43
LSTM	-0.42	-0.41	-0.39	-0.26	-0.39
RWKV	-0.39	-0.40	-0.40	-0.27	-0.33
GPT-2	0.23	0.21	0.20	0.12	0.28
			Natural Reading		
N-Gram	-0.33	-0.33	-0.31	-0.15	-0.29
RNN	-0.52	-0.51	-0.50	-0.26	-0.46
GRU	-0.54	-0.53	-0.52	-0.29	-0.48
LSTM	-0.52	-0.50	-0.49	-0.26	-0.46
RWKV	-0.39	-0.39	-0.38	-0.19	-0.28
GPT-2	0.33	0.30	0.30	0.14	0.37

Table 2: **Prediction Probability Correlation Results** using Spearman correlation metric. The numbers in blue mean the significant negative correlation, while the red represent the positive correlation. (Significant at $p < 0.05$)

6. Conclusion

In this work, we probe LLMs through human behavior, specifically employing eye-tracking measurements to dissect the internal workings of LLMs, including the feed-forward layers and multi-head attention. Our findings reveal a similarity between LLMs and humans on word prediction: both exhibit a tendency where heightened attention to pivotal words results in more confident predictions. Our analysis further delineates that feed-forward networks begin to align with human fixation patterns starting from the middle layers, leveraging upper layers to broaden the contextual understanding. Our probing approach stands out for its interpretability from human reading indicators and paves the way for the development of LLMs that are not only reliable but also imbued with a greater degree of trustworthiness.

7. Acknowledgements

This research was funded by the German Research Foundation DFG Transregio SFB 169: Crossmodal Learning: Adaptivity, Prediction, and Interaction.

8. Bibliographical References

Hervé Abdi. 2007. The kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pages 508–510.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yves Bestgen. 2021. Last at cmcl 2021 shared task: Predicting gaze data during reading with a gradient boosting decision tree approach. *arXiv preprint arXiv:2104.13043*.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. *https://openai.com/research/language-models-can-explain-neurons-in-language-models*.

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.

BSI. 1973a. *Natural Fibre Twines*, 3rd edition. British Standards Institution, London. BS 2570.

BSI. 1973b. Natural fibre twines. BS 2570, British Standards Institution, London. 3rd. edn.

John C Caruso and Norman Cliff. 1997. Empirical size, coverage, and power of confidence intervals for spearman's rho. *Educational and psychological Measurement*, 57(4):637–654.

A. Castor and L. E. Pollux. 1992. The use of user modelling to guide inference and learning. *Applied Intelligence*, 2(1):37–53.

J.L. Chercheur. 1994. *Case-Based Reasoning*, 2nd edition. Morgan Kaufman Publishers, San Mateo, CA.

N. Chomsky. 1973. Conditions on transformations. In *A festschrift for Morris Halle*, New York. Holt, Rinehart & Winston.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting geco: An eye-tracking corpus of monolingual and bilingual sentence reading. *Behavior research methods*, 49:602–615.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Oliver Eberle, Stephanie Brandl, Jonas Pilot, and Anders Søgaard. 2022. Do transformer models show similar attention patterns to task-specific human gaze? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 4295–4309.

Umberto Eco. 1990. *The Limits of Interpretation*. Indian University Press.

David Freedman, Robert Pisani, and Roger Purves. 2007. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45.

Paul Gerhard Hoel. 1971a. *Elementary Statistics*, 3rd edition. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester. ISBN 0 471 40300.

Paul Gerhard Hoel. 1971b. *Elementary Statistics*, 3rd edition, Wiley series in probability and mathematical statistics, pages 19–33. Wiley, New York, Chichester. ISBN 0 471 40300.

Markus J Hofmann, Steffen Remus, Chris Biemann, Ralph Radach, and Lars Kuchinke. 2022. Language models explain word reading times better than empirical predictability. *Frontiers in Artificial Intelligence*, 4:214.

Nora Hollenstein, Emmanuele Chersoni, Cassandra L. Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. 2021. CMCL 2021 shared task on eye-tracking prediction. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 72–78.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. Zuco, a simultaneous eeg and eye-tracking resource for natural sentence reading. *Scientific data*, 5(1):1–13.

Nora Hollenstein, Marius Troendle, Ce Zhang, and Nicolas Langer. 2019. Zuco 2.0: A dataset of physiological recordings during natural reading and annotation. *arXiv preprint arXiv:1912.00903*.

Otto Jespersen. 1922. *Language: Its Nature, Development, and Origin*. Allen and Unwin.

Chak Leong, Yi Cheng, Jiashuo Wang, Jian Wang, and Wenjie Li. 2023. Self-detoxifying language models via toxification reversal. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4433–4449.

Steven G Luke and Kiel Christianson. 2018. The provo corpus: A large eye-tracking corpus with predictability norms. *Behavior research methods*, 50:826–833.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Honzaernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*.

Adam Pauls and Dan Klein. 2011. Faster and smaller n-gram language models. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 258–267.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. 2023. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306.

Charles Joseph Singer, E. J. Holmyard, and A. R. Hall, editors. 1954–58. *A history of technology*. Oxford University Press, London. 5 vol.

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753, Istanbul, Turkey. European Language Resource Association (ELRA).

S. Superman, B. Batman, C. Catwoman, and S. Spiderman. 2000. *Superheroes experiences with books*, 20th edition. The Phantom Editors Associates, Gotham City.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, et al. 2023. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*.

9. Ethical Considerations

The eye-tracking data employed in this study, derived from the ZuCo 2.0 dataset, are publicly accessible and adhere to established ethical protocols.

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.

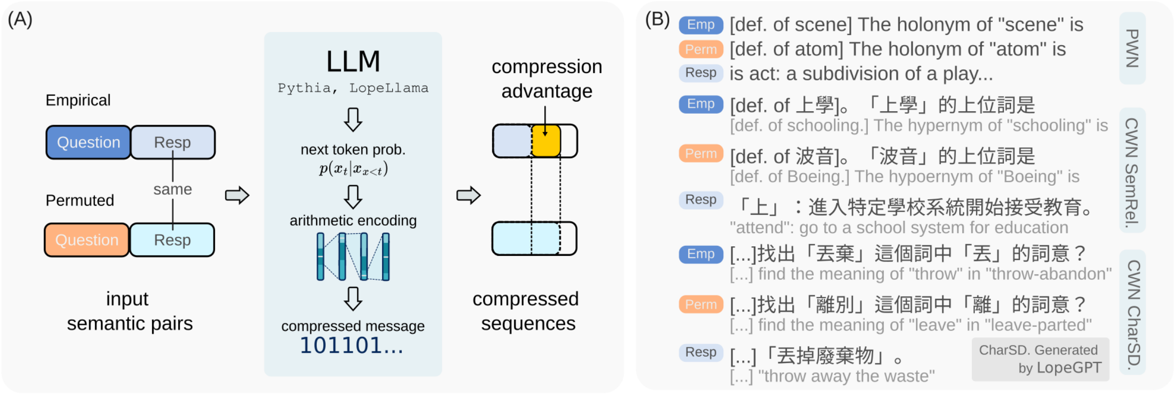

(A)
Empirical
Question
Resp
same
Permuted
Question
Resp
input
semantic pairs
LLM
Pythia, LopeLlama
next token prob.
$p(x_t|x_{<t})$
arithmetic encoding
compressed message
101101...
compression
advantage
compressed
sequences
(B)
Emp [def. of scene] The holonym of "scene" is
Perm [def. of atom] The holonym of "atom" is
Resp is act: a subdivision of a play...
PWN
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
CWN SemRel.
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"
CharSD. Generated by LopeGPT
CWN CharSD.

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/`

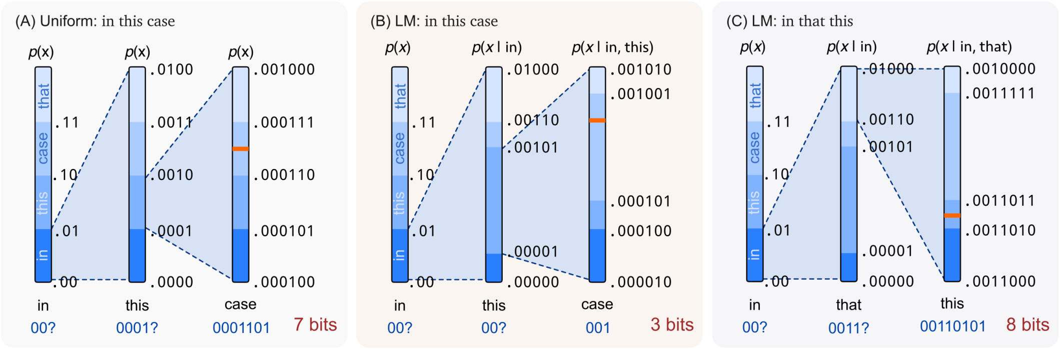


Figure 2: The schematic illustration of arithmetic encoding. Each panel shows the encoder following different probability distributions of a three-word sequence, "*in this case*". The encoder compresses one word in each step, assigning a unique interval to the word based on its probability, and the precision 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.

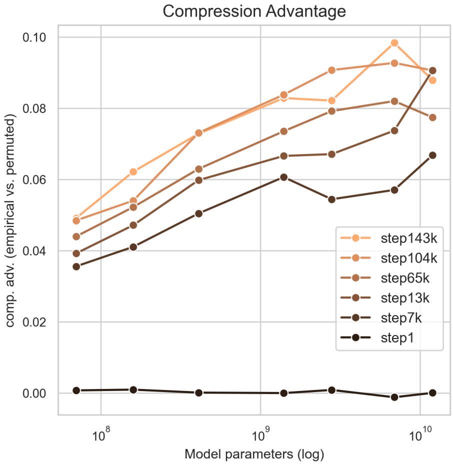


Figure 3: The compression advantage by model size and throughout training. The compression advantages consistently increase as the model increases in size and over the course of training.

	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.

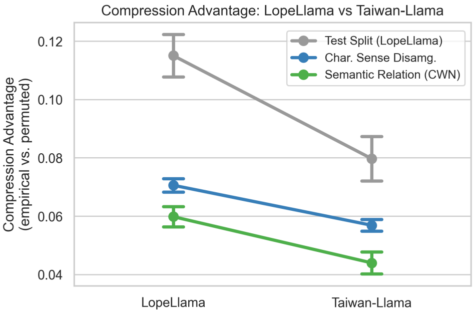


Figure 4: Compression advantages of LopeLlama and Taiwan-LLaMa on three different tasks. LopeLlama shows consistent compression advantages over the base model Taiwan-LLaMa across different datasets. Error bars indicate one standard error.

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

Word Sense Disambiguation as a Game of Neurosymbolic Darts

Tiansi Dong, Rafet Sifa
Media Engineering, Fraunhofer IAIS
Schloss Birlinghoven, 1, 53757 Sankt Augustin, Germany
{tiansi.dong, rafet.sifa}@iais.frauhofer.de

Abstract

Word Sense Disambiguation (WSD) is one of the hardest tasks in natural language understanding and knowledge engineering. The glass ceiling of the 80% F1 score is recently achieved through supervised learning, enriched by knowledge graphs. Here, we propose a novel neurosymbolic methodology that may push the F1 score above 90%. The core of our methodology is a neurosymbolic sense embedding, in terms of a configuration of nested n-dimensional balls. The central point of a ball well preserves pre-trained word embeddings learned from data, which partially fixes the locations of balls. Inclusion relations among balls precisely encode symbolic hypernym relations among senses, and enable simple logic deduction among sense embeddings. We trained a Transformer to learn the mapping from a contextualized word embedding to its sense ball embedding, just like playing the game of darts (a game of shooting darts into a dartboard). A series of experiments are carried out using pretraining n ball embeddings, which cover around 70% training data and 75% testing data in the benchmark WSD corpus. Euclidean distance and cosine similarity functions are used as objective functions, separately, and each reaches $> 95.0\%$ F1 score in the ALL-$n$ball dataset. This substantially breaks the glass ceiling of deep learning methods. Future work is discussed to develop a full-fledged neurosymbolic WSD system that substantially outperforms deep learning approaches.

Keywords: sense disambiguation, neurosymbolic representation, knowledge graph, NLP

1. Introduction

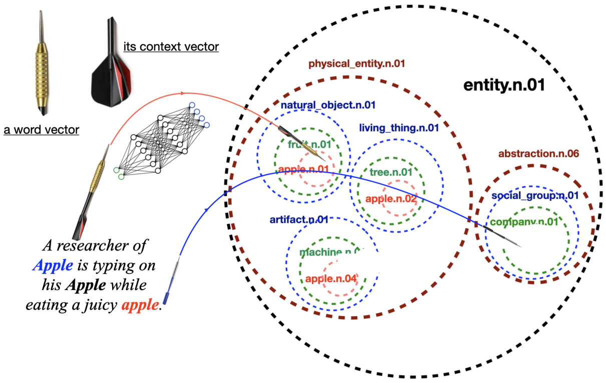


Figure 1: A neurosymbolic approach to Word-Sense Disambiguation works like playing the game of Dart. A deep neural network learns to shoot a contextualized word embedding vector to its sense regions in the Dart board.

Word Sense Disambiguation (WSD) is the task of acquiring the intended meaning of a word within the context where it appears (Navigli, 2009). It is one of the fundamental topics of natural language understanding in Artificial Intelligence (AI) (Weaver, 1949/1955), in part because WSD is hard, and has wide applications, such as information extraction, machine translation, opinion mining, question-answering, sentiment analysis, text understanding. Deep learning approaches have attained estimated human performance, and reached a glass ceiling over 80% (Bevilacqua et al., 2021), yet, they still make simple mistakes that humans would not do (Maru et al., 2022). Technically, classifying a word and its context into a word-sense class is limited to the knowledge that can be acquired from the training data (Bevilacqua et al., 2021), because word-senses are represented as *opaque* classes, and symbolic hypernym relations among senses cannot be used for deduction in the vector space. However, recent researches show ways to represent sense class in probabilistic box lattice (Vilnis et al., 2018) or fuzzy boxes (Dasgupta et al., 2022), or approximated in the hyperbolic space (Nickel and Kiela, 2017). However, it is possible to embed without loss a large symbolic tree-structured taxonomy of word senses as nested spheres with crisp boundaries, while well-preserving pre-trained vector embedding in the sphere centres (Dong et al., 2019a,b; Dong, 2021). In such a neurosymbolic paradigm, a word-sense is no more an *opaque* class; rather, it is explicitly embedded as an n-dimensional region with a crisp boundary. This provides a new way to tackle the tough WSD problem. Here, we vividly describe the new approach as a game of darts as follows: A neurosymbolic WSD is a neural dart player that shoots a contextualized word vector to the place of a configuration of regions, where its sense is located. This configuration of regions precisely encodes the sense inventory and latent features of words, as illustrated in Figure 1.
For example, apple.n.01, orange.n.01, and watermelon.n.01 are members of fruit.n.01. In classic deep-learning approaches, they are embedded as

Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ LREC-Coling 2024, pages 22–32
21 May, 2024.

four vectors. Here, we extend them into regions, so that their membership relations are explicitly represented by inclusion relations among these balls: the ball of fruit.n.01 contains the balls of apple.n.01, orange.n.01, and watermelon.n.01. The advantage for WSD is not only that shooting to a region is much easier than shooting to a point, but also that explicit region representation enables logical deduction among senses: shooting a contextualized vector of the word *apple* to the region of fruit.n.01 is sufficient to determine apple.n.01 as the intended sense; while shooting to the region of abstraction.n.06 is reasonable to hypothesize that it may refer to a company, even *apple* does not have the sense of company (*company.n.01*) in the sense inventory, as shown by the blue shooting path in Figure 1. The contribution in this paper are listed as follows.

1. We propose a novel neurosymbolic methodology for WSD, which seamlessly unifies supervised learning approaches and simple symbolic reasoning among hypernym relations;
2. We implement a simple Transformer to realise the first neurosymbolic WSD system, whose input is pre-trained word embeddings and whose output is a vectorial location in the pre-trained n-ball sense embeddings. The performance of this WSD breaks the ceiling of traditional deep-learning approaches in all 6 benchmark datasets where hypernym structures are available, and outperforms ChatGPT;
3. Our experiments show that using Direct Upper Hypernym (DUH) in testing achieved the best F1 score, while using DUH in training reduces the amount of training senses without weakening the performance;
4. Supported by our preliminary experiments, we envisage a novel neurosymbolic WSD system that may greatly outperform current SOTA systems and list a number of future works.

The rest of the article is structured as follows: we first review the recent WSD methods, and motivate our approach; then, we describe the details of the novel neurosymbolic approach. In experiments, we first set the targets, and report the statistics of training and testing dataset, then report and analyse experiments results, by comparing with performances of SOTA WSD systems and ChatGPT. In the end, we list a number of future work to realise full-fledged neurosymbolic WSD systems.

2. Related Works

2.1. Word Sense Disambiguation

The research on Word Sense Disambiguation (WSD) has a long history, with contributions from many researchers worldwide. A recent survey can be found in (Bevilacqua et al., 2021). The task of WSD is to automatically decide the intended sense in a given context, where senses of words are selected from the fixed word-sense inventory. A WSD system has three components, as follows: (1) a word in a given context, (2) a word sense inventory, e.g., WordNet (Miller et al., 1990; Miller, 1995), BabelNet (Navigli et al., 2021), and (3) an annotated corpus, e.g. SemCor (Miller et al., 1993), where some words have been manually or automatically annotated with intended word senses. The knowledge graph approaches and supervised deep-learning approaches are the main WSD approaches. Their performances are determined by the quality and the size of the knowledge bases (Pilehvar and Navigli, 2014).

Knowledge-based approaches for WSD Knowledge-based approaches leverage part of the graph structure of word-sense inventories, e.g. WordNet, BabelNet, where words connect with all their senses. By injecting the context of a word into the graph will slightly change the graph structure, and affect the probability distribution of senses of the word in the graph, which can be computed by the Personized PageRank algorithm (Agirre et al., 2014). The sense with the highest probability will be selected. This approach can be improved by connecting word-sense inventory with large web texts, e.g., BabelNet (Navigli et al., 2021), a knowledge base that integrates WordNet with Wikipedia (Moro et al., 2014).

From the game theoretical perspective (von Neumann and Morgenstern, 1947), a word can be viewed as a player, and its possible senses as strategies that the player can choose, to maximize a utility function (Tripodi and Navigli, 2019). Precisely, let $W = \{w_1, \ldots, w_n\}$ be the set of the content words in text T, $S_i = \{s_1, \ldots, s_{m_i}\}$ be the set of senses of w_i, $\mathcal{S} = \bigcup S_i$ is the set of all the strategies of the games. The strategy space of a player w_i is represented as a probabilistic distribution $\mathbf{x}_i$. The way how the context determines senses of words is simulated by interactions between two words w_i and w_j through a utility matrix Z. The cell $z_{r,t}$ represents the utility value when w_i chooses the r^{th} strategy and w_j chooses the t^{th} strategy. The value of one sense's strategy is related to its partners, in the following three aspects: word similarity, word-sense similarity, and their sense distributions, and computed in the manner similar to the attention mechanism.

Supervised deep learning for WSD Supervised deep learning approaches frame WSD as a multi-classification task – classifying a word w plus its context C into one of its word-senses s, using an annotated corpus $\mathcal{D}$, in the form of a list of triples $< w, c, s >$, and realized by supervised deep learning (Kågebäck and Salomonsson, 2016; Raganato et al., 2017b; Uslu et al., 2018).

The straightforward way of the supervised deep-learning approach is to compare the similarity between the contextualized embedding of a word w in the testing context c and senses s in the annotated corpus, and choose the most similar one, measured by a loss function $\mathcal{L}(w, c, s)$, either by feed-forward networks (Hadiwinoto et al., 2019), or transformers (Bevilacqua and Navigli, 2019). In these approaches, word senses are treated as discrete class labels. This may cause poor performance on low-frequency senses. To overcome this limitation, (Kumar et al., 2019) explicitly computed word sense embeddings by applying embedding methods for the hypernym structure of the WordNet, then trained an attentive BiLSTM to learn the context embedding of a word to its sense embedding. (Scarlini et al., 2020) computed contextualized sense embeddings by utilizing a variety of resources, such as SemCor, gloss in WordNet, SyntagNet (Maru et al., 2019), UKB (Agirre et al., 2014), and BERT (Devlin et al., 2018). (Loureiro and Jorge, 2019) computed sense embeddings by fully utilizing relations in WordNet, and achieved very competitive performance. Using explicit sense embeddings, (Bevilacqua and Navigli, 2020) successfully reached over 80% F1 score for WSD. (Barba et al., 2021) is able to choose the most important context definition for the target word. Their method inherits the idea of the game-theoretic WSD approach by using a feedback loop to consider the explicit senses of nearby words.

2.2. Neuosymbolic Unification

Both knowledge-based and supervised deep-learning WSD approaches have two assumptions as follows: (1) word senses are opaque classes, (2) a sense inventory has a fixed taxonomy (Bevilacqua et al., 2021). Consequently, in knowledge-based WSD approaches, word senses are represented by probabilistic distributions; in supervised WSD approaches, word senses are represented by latent vector embeddings. However, the two assumptions are somehow incompatible with the existence of a symbolic sense inventory – if a sense inventory has a well-structured and fixed taxonomy, why senses are opaque classes in both approaches? Such incompatibility lies in the discrepancy between the continuous numeric sense representation and the discrete symbolic sense representation – The continuous numeric representation, either as a probabilistic distribution or as a latent vector, cannot explicitly represent the well-defined symbolic taxonomy structure. This incompatibility could be resolved, if word sense embedding can precisely encode the discrete symbolic fixed taxonomy.

A vector sense embedding can be enlarged into an n-dimensional ball, whose radius is geometrically computed to strictly satisfy two conditions as follows: (1) balls of sibling senses are disconnected from each other; (2) balls of child and parent senses are precisely nested – the ball of a child sense is inside the ball of its parent sense. By utilising geometric methods, (Dong et al., 2019a) precisely injected a large tree-structured taxonomy of senses in WordNet-3.0 into pre-trained word embeddings, resulting in a configuration of nested low-dimensional balls. Thus, these nested balls unify numerical vector embeddings and symbolic structures into one representation without loss. Hyperbolic geometric embedding also has the power of neuro-symbolic unification (Tifrea et al., 2019; Chami et al., 2020), so that computational models can inherit good features from both neural computing and symbolic reasoning (Besold et al., 2017; Dong, 2021; Dong et al., 2022; Garcez and Lamb, 2023).

3. *Dart4WSD*: A neurosymbolic Darter

Dart4WSD is a novel supervised neurosymbolic learning methodology for Word Sense Disambiguation, with the novelty that senses are embedded as regions in vector space and that these region embeddings explicitly represent a fixed taxonomy in a sense inventory and well-preserve pre-train vector embeddings. *Dart4WSD* utilises a Transformer to learn the intended sense of a word in a given context, whose general architecture consists of five components; word embedding, a fixed sense inventory, a network that learns contextualized word embedding, a network that transforms the contextualized word embedding to a location in the neurosymbolic region, as illustrated in Figure 2.

3.1. Notations used in *Dart4WSD*

Let w and $\overrightarrow{w}$ be a word and its vector word-embedding, respectively, C represent a context; $\overrightarrow{w}_C$ be a vector embedding of word w in the context C. Let $\overrightarrow{V}_{w_C}$ be the output of our neural network, with the input $\overrightarrow{w}_C$, that is, $\overrightarrow{V}_{w_C} = NN(\overrightarrow{w}_C)$. Let w have k different senses in the inventory $\mathcal{S}_w = \{S_1^w, \ldots, S_k^w\}$, and $\mathcal{O}[S_i^w]$ be the ball embedding of S_i^w, with the central point $\overrightarrow{O}[S_i^w]$ and the radius $r[S_i^w]$.

3.2. The task formulation for *Dart4WSD*

Given an annotated corpus $\mathcal{D}$, we train a neural network NN, with a loss function $\mathcal{L}(NN(\overrightarrow{w}_C), \mathcal{O}[S_i^w])$ that improves the shooting technique of NN so that most of its output vectors are located inside balls of the target senses. In this preliminary work, we compare two objective functions: (1) the Euclidean distance, $\overrightarrow{V}_{w_C} = NN(\overrightarrow{w}_C)$ is inside $\mathcal{O}[S_i^w]$, that is, the distance between $\overrightarrow{V}_{w_C}$ and $\overrightarrow{O}[S_i^w]$ is less than or equal to $r[S_i^w]$. That is, $\mathcal{L}_{dis}(\overrightarrow{V}_{w_C}, \mathcal{O}[S_i^w]) = \max\{0, \|\overrightarrow{V}_{w_C} - \overrightarrow{O}[S_i^w]\| -$

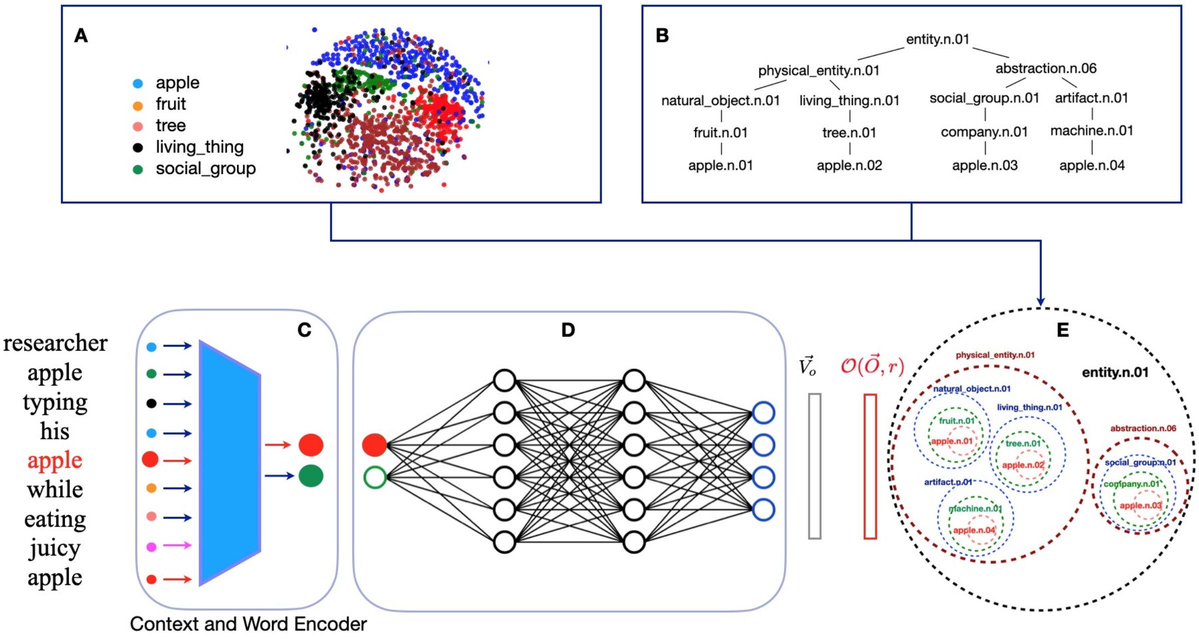


Figure 2: The supervised learning architecture of *Dart4WSD*: (**A**) word embeddings; (**B**) the fixed word sense taxonomy extracted from a sense inventory; (**C**) a neural network that learned contextualized word embeddings; (**D**) a neural network that learns to map a word in a context to its word sense ball embedding; (**E**) the neurosymbolic nested ball embeddings of word senses.

$r[S_i^w]\}$; (2) the well-known cosine similarity, that is, $\mathcal{L}_{cos}(\vec{V}_{w_C}, \mathcal{O}[S_i^w]) \approx \cos(\vec{w}_C, \vec{O}[S_i^w])$. The cosine approximation works well, when balls of sibling senses in the inventory are of the similar size. For example, in Figure 3, the apple.n.01, orange.n.01, and watermelon.n.01, three child senses of fruit.n.01, are embedded as balls with similar sizes; fruit.n.01 and tree.n.01 are siblings at the upper level in the inventory, and also embedded in the similar size. To correctly determine that the word *apple* in the phrase *eating a juicy apple*, the neural network shall map the contextualized word embedding ($\overrightarrow{apple}_{\text{eating a juicy}}$) to a vector inside the ball of the sense fruit.n.01 ($\mathcal{O}[S_1^{fruit}]$). Then, the sense apple.n.01 inside the fruit.n.01 will be chosen as the target sense.

Using upper category information for WSD in the embedding space has been proposed in (Beviá et al., 2006; Vial et al., 2019), we show that using explicit region embedding can fully utilise the upper category information, for at least two reasons as follows: (1) explicit and precise boundaries of regions endow our method the ability to reason with the symbolic hypernym relations in the embedding space; (2) It is reasonable to argue that the context information *eating a juicy…* shall not provide information to direct the word embedding of *apple* exactly to the ball embedding of apple.n.01, as *eating a juicy orange* and *eating a juicy watermelon* are as meaningful as *eating a juicy apple*. We argue that this context information shall direct the word embedding of *apple* towards the sense embedding of its direct upper hypernym, here, fruit.n.01, and deviate from direct upper hypernym balls of its other senses, here, tree.n.01.

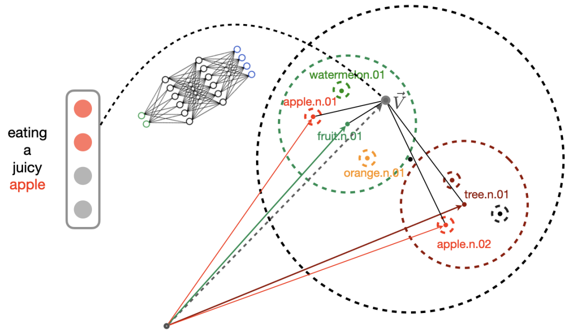


Figure 3: A novel method to choose senses by carrying out reasoning with hypernym relations in the embedding space: as long as the contextualised word embedding $\overrightarrow{apple}_{\text{eating a juicy}}$ is shot within the fruit.n.01 ball, our system will choose apple.n.01 as the target sense.

Let $H_1(S_i^w)$ be the direct upper hypernym of S_i^w in the inventory. We assume that there are no two S_i^w and S_j^w have the same direct upper hypernym, that is, $H_1(S_i^w) \neq H_1(S_j^w)$, if $S_i^w \neq S_j^w$. In the case of using Euclidean distance as the objective function, the sense of w, whose $\mathcal{O}[H_1(S_i^w)]$ (the boundary of the ball of the direct upper hypernym of w) is

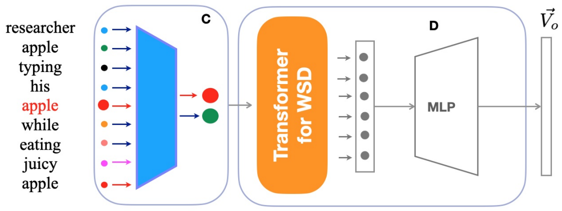


Figure 4: A transformer architecture for *Dart4WSD*.

	#training	#exclude	#n-ball	#no ball
SC	224415	56207	156483	11725
SC+O	1135547	259375	837147	39025

Table 1: The statistics of the numbers of training records. **SC** represents **SemCor**; **SC+O** represents **SemCor+OMSTI**.

senses	#s-class	#s-nball	#s-L1
SC	18953	**15025**	**5799**
SC+O	19253	**15298**	**5852**

Table 2: The statistics of senses in our experiments. #s-class: the total number of senses whose hypernym path is longer than 1; #s-nball: the total number of senses that have ball embedding; #s-L1: the total number of senses that are the direct hypernym of senses in #s-nball.

nearest to $\vec{V}_{wC}$, will be selected as the sense of w in the current context.

$$S^w = \arg\min_{S_i^w \in \mathcal{S}_w} \max\{0, \|\vec{V}_{wC} - \vec{O}[S_i^w]\| - r[S_i^w]\}$$

In the case of using cosine similarity as the objective function, the sense of w, whose $\vec{O}[H_1(S_i^w)]$ (the centre vector of the ball of the direct upper hypernym of w) has the largest cosine value with $\vec{V}_{wC}$, will be selected as the sense of w in the current context.

$$S^w = \arg\max_{S_i^w \in \mathcal{S}_w} \cos(\vec{V}_{wC}, \vec{O}[H_1(S_i^w)])$$

3.3. The neurosymbolic Dartboard for senses

Considering the normal WSD situation that each contextualised word has one target sense, we restrict here the taxonomy of word senses as a tree structure. Accordingly, child-parent senses are precisely encoded as the child ball is inside the parent ball; sibling relation senses are precisely encoded as the disconnectedness relations among sibling sense balls, as illustrated in Figure 3. These features are fulfilled by n-ball embeddings (Dong et al., 2019a,b; Dong, 2021), in which (1) the symbolic taxonomy of word senses is explicitly and precisely encoded by boundary relations among regions, and (2) existing vector sense embedding is preserved by the centre vector of a region, as illustrated in Figure 2. Thus, we use n-balls as the neurosymbolic Dartboard of *Dart4WSD*, for quick prototyping and the proof of concept, and also for the ease of re-production and extension.

3.4. A supervised learning process

The Transformer architecture was originally designed for sequence-to-sequence tasks (Vaswani et al., 2017), and has been applied in a variety of fields (Lin et al., 2021). It can be used as a universal approximation of sequence-to-sequence functions (Yun et al., 2020). We use a Transformer architecture to learn the mapping from the contextualized words to balls of their target senses, as illustrated in Figure 4. Given a sentence s, we transform it into a list of tokens $(t_1, t_2, t_3 ... t_m)$, then, replace each token with contextualised word embedding, $\vec{t}_{1,C_1}, \vec{t}_{2,C_2}, \vec{t}_{3,C_3} \ldots \vec{t}_{m,C_m}$ (Yap et al., 2020). We feed $\vec{t}_{C_i}$ into a Transformer (TF), whose outputs are fed into a two-layered perceptron as follows. Ideally, the output of the perceptron $\vec{V}$ shall be inside the n-ball of the target word sense.

$$\vec{V}_{wC} = Linear(Relu(Linear(TF(\vec{t}_{i,C_i}))))$$

4. Experiments

The target of the experiments is to examine the WSD performance, when the symbolic structure of the sense classes is explicitly and precisely represented in the vector space. We developed *Dart4WSD* as the first such a WSD system, and compared its performance with the SOTA performance, and with the WSD performance of ChatGPT. Our four experiments are designed to answer the questions as follows.

1. How is the WSD performance of LLMs, e.g., ChatGPT?

2. How good is *Dart4WSD* in the task of mapping contextualized word **vector** to sense **vectors**, using Euclidean distance and cosine similarity objective functions, respectively? Which objective function leads to better performance?

3. How is the performance of *Dart4WSD*, if it uses the direct upper hypernym of the target sense (here, n-dimensional balls)? Which objective function leads to better performance?

4. Will the performance be improved in the testing phase, if in the learning phase *Dart4WSD* maps to n-balls of direct upper hypernym senses?

	#test	#exclude	#nball	#no nball
S-2	2275	722	1459	94
S-3	1832	396	1341	95
S-07	449	8	420	21
S-13	1621	0	1435	186
S-15	1013	248	712	53
ALL	7181	1374	5358	449

Table 3: The statistics of testing records. **S-2** represents **Senseval-2**, **S-3** represents **Senseval-3**, **S-07** represents **SemEval-07**, **S-13** represents **SemEval-13**, **S-15** represents **SemEval-15**.

	#S-2/L1	**#S-3/L1**	**#S-07/L1**
#nball	711/522	780//605	327/281
	#S-13/L1	**#S-15/L1**	**#A/L1**
#nball	669/408	350/256	2251/1424

Table 4: The statistics of senses in test records. **#S-2/L1** represents the numbers of different n-balls in **Senseval-2** and the direct upper level hypernyms. Others are interpreted in the same way.

4.1. Datasets

We exclude, from benchmark datasets SemCor (**SC**) and SemCor+OMSTI (**SC+O**), senses that do not have class structures, as our target focuses on the WSD performances, subject to *opaque* or *clear* embedding of sense classes.

SemCor has 224415 training records, among which there are 25845 different senses; senses in 156483 records have n-ball embedding, among which there are 15025 different senses; senses in 11725 records do not yet have n-ball embedding, totalling 3928 different senses. Senses in 56207 records do not have a taxonomy, totalling 6892 different senses.

SemCor+OMSTI has 1135547 training records, among which there are 26265 different senses; senses in 837147 records have n-ball embedding, among which there are 15298 different senses. Senses in 39025 records do not yet have n-ball embedding, totaling 3955 different senses. Senses in 259375 records do not have a taxonomy, totaling 7012 different senses, as listed in Table 1 and Table 2. The n-ball embedding contains 47,634 word senses, covering around 80% senses in the WSD benchmark datasets.

4.1.1. Training data

We create four training datasets, as follows: (1) **SemCor-$n$ball**, (2) **SemCor+OMSTI-$n$ball**, (3) **SemCor-$n$ball-L1**, and (4) **SemCor+OMSTI-$n$ball-L1** in the following way: Firstly, we transform training data into the form as follows: "(sense, a list of word, the index for the word(s) of the sense)". For example, *('aim.n.02', ['have', 'you', 'set', 'specific', 'objectives'], [4])*, which means that the word pointed by the index 4, that is the word 'objectives', should have the sense 'aim.n.02'. The first two datasets **SemCor-$n$ball** and **SemCor+OMSTI-$n$ball** are extracted from SemCor and SemCor+OMSTI with the criteria that target senses have n-ball embeddings. That is, if 'aim.n.02' has an n-ball embedding, this piece of training record will be selected. The other two datasets are created, by setting each target sense in the first two datasets with its direct hypernym. If this hypernym has n-ball embedding, the training record will be selected. For example, 'aim.n.02' has an hypernym path in WordNet-3.0, as follows: ['aim.n.02', 'goal.n.01', 'content.n.05', 'cognition.n.01', …]. Its direct hypernym is 'goal.n.01'. If it has an n-ball embedding, the following training record will be added into the corresponding **-L1** dataset, for example, *('goal.n.01', 'aim.n.02', ['have', 'you', ..., 'objectives'], [4])*.

4.1.2. Testing data

We create $6 \times 2 = 12$ datasets from the six benchmark datasets, namely, **Senseval-2**, **Senseval-3**, **SemEval-07**, **SemEval-13**, **SemEval-15**, **ALL** (Raganato et al., 2017a). From each dataset **E**∈{**Senseval-2**, **Senseval-3**, **SemEval-07**, **SemEval-13**, **SemEval-15**, **ALL**}, we derive 2 testing datasets as follws: **E-$n$ball** and **E-$n$ball-L1**. **E-$n$ball** and **E-$n$ball-L1** are created in the same way as we create training data, as listed in Table 3.

4.1.3. Evaluation

We use the F1 calculation software in the standard WSD corpus, downloaded from `http://lcl.uniroma1.it/wsdeval/home`.

4.2. Setting and running of experiments

Dart4WSD is implemented in PyTorch. We set learning rate to 0.001, 20 epochs, with 4-fold cross validation. Experiments were conducted on MacBook Pro Apple M1 Max (10C CPU/24C GPU), 32 GB memory. Using 50-d Glove word embedding, *Dart4WSD* took less than 10 seconds for one epoch for SemCor-$n$ball training data. *Dart4WSD* converges very fast: the loss of the second epoch is only one tenth of the loss of the first epoch.

4.3. Experiments and Results

4.3.1. Experiment 1

Recent research shows that LLMs, e.g., ChatGPT, can do almost perfect human-like question-answering, and their ability to reason can be improved by using prompt engineering. We created four kinds of prompts to evaluate performances of ChatGPT (gpt-3.5-turbo) on the six benchmark WSD test datasets, as follows: (1) Zero-shot prompt, which gives ChatGPT all senses of a word w, and a sentence containing w, and let ChatGPT choose the right one from the list; (2) few-shot

Obj. func.:	**Senseval-2**		**Senseval-3**		**Senseval-07**		**Senseval-13**		**Senseval-15**	
$\mathcal{L}_{dis}$	L0	L1	L0	L1	L0	L1	L0	L1	L0	L1
SC	34.1%	94.9%	37.5%	94.1%	32.0%	88.2%	32.8%	100.0%	33.9%	91.7%
SC L1	34.1%	**94.9%**	37.5%	**94.1%**	32.0%	**88.2%**	32.8%	**100.0%**	33.9%	**91.7%**
SC+O	34.1%	94.9%	37.5%	94.1%	32.0%	88.2%	32.8%	100.0%	33.9%	91.7%
SC+O L1	34.1%	94.9%	37.5%	94.1%	32.0%	88.2%	32.8%	100.0%	33.9%	91.7%

Table 5: F1 scores of 5×2 datasets by using Euclidean distance as the objective function. The F1 is computed by the standard tool for WSD, which is available in the dataset download from http://lcl.uniroma1.it/wsdeval/home.

Obj. func.:	**Senseval-2**		**Senseval-3**		**Senseval-07**		**Senseval-13**		**Senseval-15**	
$\mathcal{L}_{cos}$	L0	L1	L0	L1	L0	L1	L0	L1	L0	L1
SC	34.6%	95.2%	38.0%	93.3%	33.2%	89.5%	41.3%	100.0%	39.5%	92.9%
SC L1	34.6%	**95.2%**	38.0%	**93.3%**	33.2%	**89.5%**	41.3%	**100.0%**	39.5%	**92.9%**
SC+O	34.6%	95.2%	38.0%	93.3%	33.2%	89.5%	41.3%	100.0%	39.5%	92.9%
SC+O L1	34.6%	95.2%	38.0%	93.3%	33.2%	89.5%	41.3%	100.0%	39.5%	92.9%

Table 6: F1 scores of 5×2 datasets by using cosine similarity as the objective function.

prompt, which adds one example to the zero-shot prompt; (3) CoT prompt, which uses the gloss as a mid-step to connect a sense and the word in a context; (4) few-shot CoT, which adds an example to the Cot prompt. The zero-shot prompt produces the lowest performance, ranging from 30.7% to 37.6%, the few-shot CoT delivers the best performance, ranging from 55.4% to 68.4%, which is below 80% the glass ceiling of the SOTA performance. Other experiments found that LLMs may make correct answers with incorrect explanations (Creswell et al., 2022; Zelikman et al., 2022). Similarly, the case of WSD may provide chances to explore how ChatGPT may correctly understand the meaning of sentences, while misunderstanding the meanings of single words in the sentence.

4.3.2. Experiment2

To answer the second question, we used the **SemCor-$n$ball** dataset to train our *Dart4WSD* neural-network. It learns to map from contextualized word embeddings to centre vectors of sense n-balls. The performances using Euclidean distance range from 32.8% to 37.5% (F1 score); while the performances using cosine similarity range from 33.2% to 39.5%, in all the testing datasets, as illustrated in column L0 of Table 5, Table 6. Compared with the current best result 80% (Bevilacqua and Navigli, 2020), this performance is not good, in part because our inputs are pre-trained glove vectors and the context vector is approximated by averaging the vectors of neighbourhood words with a fixed window size, which limits the Transformer to dynamically select the right contexts, and results in a similar performance as ChatGPT using zero-shot prompt.

	ALL ($\mathcal{L}_{dis}$)		**ALL ($\mathcal{L}_{cos}$)**	
	L0	L1	L0	L1
SC	34.4%	95.2%	37.8%	95.3%
SC L1	34.4%	**95.2%**	37.8%	**95.3%**
SC+O	34.4%	95.2%	37.8%	95.3%
SC+O L1	34.4%	95.2%	37.8%	95.3%

Table 7: F1 scores of the ALL-L0 and ALL-L1 datasets. Using direct hypernyms of target senses (ALL-L1), the performances (with both objective functions) of Dart4WSD break the glass ceiling of deep learning methods.

4.3.3. Experiment3

For the third question, we used the trained model in Experiment 2, and evaluated whether it successfully hit the ball of the direct upper hypernym senses. The F1 scores range from 88.2% to 100% using Euclidean distance, and range from 89.5% to 100% using cosine similarity, as listed in Table 5, Table 6. The F1 score for the ALL-L1 data set reaches 95.0% (Table 7) with each objective function (Euclidean distance and cosine similarity), which greatly outperforms the SOTA performance (80%) (Bevilacqua et al., 2021), and break the performance ceiling (a bit above 90%) of traditional deep-learning approaches (Raganato et al., 2017a).

4.3.4. Experiment4

To answer the third question, we trained *Dart4WSD* by utilising the **SemCor-$n$ball-L1** and **SemCor$+$OMSTI-$n$ball L1** datasets. The target senses in the two training data sets are replaced by their direct hypernyms, so they have less number of senses for learning. There are no drops in the performance, as illustrated in the rows **SC L1** and **SC$+$O L1** of Table 5 – 7. This shows that *Dart4WSD* is less data-hungry, compared with

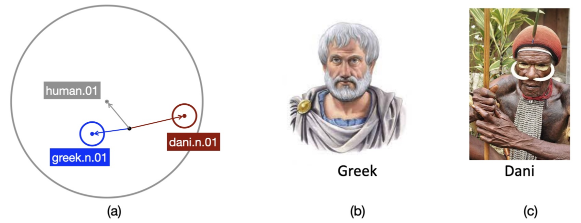


Figure 5: (a) the sphere boundary of human.n.01 includes the dani.n.01 sphere; (b-c) Sample images of Greek people and Dani people.

traditional deep learning systems.

4.3.5. Analysis and Discussions

By using the direct upper-level senses in the testing datasets, Dart4WSD outperforms ChatGPT and the SOTA systems, and even breaks the glass ceiling of deep learning approaches, in the setting of current experiments. We also performed experiments by utilising other pre-trained embeddings, e.g., BERT, and had very similar results. This convergently suggests that the high performance shall be ascribed to the neurosymbolic n-ball embedding that precisely imposes a symbolic sense inventory into the embedding space, while preserving pre-trained word embeddings in the centre points of these n-balls. In this way, the configuration of all n-balls endows *Dart4WSD* with the capability to better represent out-of-distribution data by utilising boundary relations among n-balls. For example, Dani people may have different cultures and histories from many other human races, e.g., Greeks. Their sample images as illustrated in Figure 5(b-c). Descriptions about them may appear in different types of corpus, which may result in different vector embeddings whose cosine similarity is less than 0, as illustrated in Figure 5(a). By utilising n-ball representation, they are represented within the human ball. This may bring the advantage to *Dart4WSD*, easier to make correct decisions, compared with traditional deep learning systems.

5. Conclusions and Outlooks

We prototyped *Dart4WSD*, a novel supervised neurosymbolic method for Word-Sense Disambiguation that dramatically outperforms the traditional deep learning approaches. The core of our method is a configuration of n-dimensional sphere embeddings whose boundary relations explicitly and precisely embed a symbolic sense inventory in the vector space and whose centre hosts latent features learned from data. This neurosymbolic approach is independent of languages and could be especially useful for low-resource languages. To this end, a number of problems shall be solved, listed as follows.

New Datasets for Neuro-symbolic WSD A benchmark dataset for neuro-symbolic WSD shall consist of not only labelled data for traditional supervised learning, but also a symbolic taxonomy of sense inventory. This symbolic part can be a part of a large sense inventory that only describes the taxonomy of senses in the labelled data.

Using a traditional deep-learning system as the backbone Our neurosymbolic method demonstrates its performance only when a well-designed sense inventory is available, which can be unrealistic. It would be promising to build up a neurosymbolic component above a traditional deep-learning WSD system.

More powerful geometric objective functions We used Euclidean distances and cosine similarity as two objective functions. Intuitively, Euclidean distance is more precise to measure relations between spheres, however, its performance in current experiments is a bit less than that of using cosine similarity, which cannot take the boundary information of balls into consideration. There should be powerful geometric objective functions to outperform the cosine similarity measurement.

N-ball for DAG structures The sense inventory in Word-Net 3.0 (Miller, 1995) is not a tree structure, but a Directed Acyclic Graph (DAG). We shall extend the current geometric approach for DAG structures. Creating a new n-ball configuration is not trivial, as the sense taxonomy needs to be precisely embedded (reaching the global loss of zero). This is a very challenging machine-learning task that is worth further research.

Heterogenous structure One assumption of our approach is that senses of word shall have different direct upper hypernyms, so, we can use balls of direct upper hypernyms. This assumption holds for nouns in most of the cases, but, might not hold for verbs. For example, fly.v.01 (*travel through the air; be airborne*) and fly.v.06 (*be dispersed or disseminated*) are both senses the word fly, they share the same direct upper hypernym travel.n.01 (*change location; move, travel, or proceed, also metaphorically*). In this case, using direct upper hypernym is not sufficient to disambiguate between fly.v.01 and fly.v.06. We may need to integrate other knowledge into the sense inventory. We may need to consider Descartes's product of n-balls. For example, one encodes hypernym relations, another encodes part-whole relations.

Towards a new methodology for classification *Dart4WSD* can be generalised for solving any classification problem. In contrast to traditional supervised deep-learning methods, our method will create the dart board before shooting, instead of the other way around (shooting first, then drawing the best-fit target, as described in (Gigerenzer, 2022)).

6. Bibliographical References

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40:57–84.

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2021. Consec: Word sense disambiguation as continuous sense comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1492–1503.

Tarek R. Besold, Artur S. d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro M. Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luís C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *CoRR*, abs/1711.03902.

Michele Bevilacqua and Roberto Navigli. 2019. Quasi bidirectional encoder representations from transformers for word sense disambiguation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 122–131, Varna, Bulgaria. INCOMA Ltd.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. Association for Computational Linguistics.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. pages 4330–4338.

Rubén Beviá, Armando Suárez Cueto, and German Rigau. 2006. Exploring the automatic selection of basic level concepts.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, Online. Association for Computational Linguistics.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning.

Shib Dasgupta, Michael Boratko, Siddhartha Mishra, Shriya Atmakuri, Dhruvesh Patel, Xiang Li, and Andrew McCallum. 2022. Word2Box: Capturing set-theoretic semantics of words using box embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2263–2276, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tiansi Dong. 2021. *A Geometric Approach to the Unification of Symbolic Structures and Neural Networks*, volume 910 of *Studies in Computational Intelligence*. Springer-Nature.

Tiansi Dong, Chrisitan Bauckhage, Hailong Jin, Juanzi Li, Olaf H. Cremers, Daniel Speicher, Armin B. Cremers, and Jörg Zimmermann. 2019a. Imposing Category Trees Onto Word-Embeddings Using A Geometric Construction. In *ICLR-19*, New Orleans, USA. May 6-9.

Tiansi Dong, Achim Rettinger, Jie Tang, Barbara Tversky, and Frank van Harmelen. 2022. Structure and Learning (Dagstuhl Seminar 21362). *Dagstuhl Reports*, 11(8):11–34.

Tiansi Dong, Zhigang Wang, Juanzi Li, Christian Bauckhage, and Armin B. Cremers. 2019b. Triple Classification Using Regions and Fine-Grained Entity Typing. In *AAAI-19*, pages 77–85.

Artur Garcez and Luís Lamb. 2023. Neurosymbolic AI: the 3rd wave. *Artificial Intelligence Review*, pages 1–20.

Gerd Gigerenzer. 2022. *How to Stay Smart in a Smart World: Why Human Intelligence Still Beats Algorithms*. The MIT Press.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. 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 5297–5306, Hong Kong, China. Association for Computational Linguistics.

Mikael Kågebäck and Hans Salomonsson. 2016. Word sense disambiguation using a bidirectional LSTM. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 51–56, Osaka, Japan. The COLING 2016 Organizing Committee.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence, Italy. Association for Computational Linguistics.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2021. A survey of transformers. *arXiv preprint arXiv:2106.04554*.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation. pages 5682–5691.

Marco Maru, Simone Conia, Michele Bevilacqua, and Roberto Navigli. 2022. Nibbling at the hard core of Word Sense Disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, Dublin, Ireland. Association for Computational Linguistics.

Marco Maru, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. Syntagnet: Challenging supervised word sense disambiguation with lexical-semantic combinations. pages 3525–3531.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An On-line Lexical Database*. *International Journal of Lexicography*, 3(4):235–244.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, HLT '93, pages 303–308, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2).

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of babelnet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4559–4567. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Nips*, pages 6338–6347.

Mohammad Taher Pilehvar and Roberto Navigli. 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics*, 40(4):837–881.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167, Copenhagen, Denmark. Association for Computational Linguistics.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3528–3539, Online. Association for Computational Linguistics.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2019. Poincaré glove: Hyperbolic word embeddings. *ICLR-19*.

Rocco Tripodi and Roberto Navigli. 2019. Game theory meets embeddings: a unified framework for word sense disambiguation. pages 88–99.

Tolga Uslu, Alexander Mehler, Daniel Baumartz, and Wahed Hemati. 2018. FastSense: An efficient word sense disambiguation classifier. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation. In *Proceedings of the 10th Global Wordnet Conference*, pages 108–117, Wroclaw, Poland. Global Wordnet Association.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 263–272, Melbourne, Australia. Association for Computational Linguistics.

J. von Neumann and O. Morgenstern. 1947. *Theory of games and economic behavior*. Princeton University Press.

Warren Weaver. 1949/1955. Translation. In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA. Reprinted from a memorandum written by Weaver in 1949.

Boon Yap, Andrew Koh, and Eng Chng. 2020. Adapting bert for word sense disambiguation with gloss selection objective and example sentences. pages 41–46.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2020. Are transformers universal approximators of sequence-to-sequence functions? In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STar: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

7. Language Resource References

Open Event Causality Extraction by the Assistance of LLM in Task Annotation, Dataset, and Method

Kun Luo[1,2], Tong Zhou[1,2], Yubo Chen[1,2], Jun Zhao[1,2] and Kang Liu[1,2,3*]
[1]The Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Shanghai Artificial Intelligence Laboratory
{luokun2024, tong.zhou}@ia.ac.cn
{yubo.chen, jzhao, kliu}@nlpr.ia.ac.cn

Abstract

Event Causality Extraction (ECE) aims to extract explicit causal relations between event pairs from the text. However, the event boundary deviation and the causal event pair mismatching are two crucial challenges that remain unaddressed. To address the above issues, we propose a paradigm to utilize LLM to optimize the task definition, evolve the datasets, and strengthen our proposed customized **C**ontextual **H**ighlighting **E**vent **C**ausality **E**xtraction framework (CHECE). Specifically in CHECE, we propose an Event Highlighter and an Event Concretization Module, guiding the model to represent the event by a higher-level cluster and consider its causal counterpart in event boundary prediction to deal with event boundary deviation. And we propose a Contextual Event Causality Matching mechanism, meanwhile, applying LLM to diversify the content templates to force the model to learn causality from context to targeting on causal event pair mismatching. Experimental results on two ECE datasets demonstrate the effectiveness of our method.

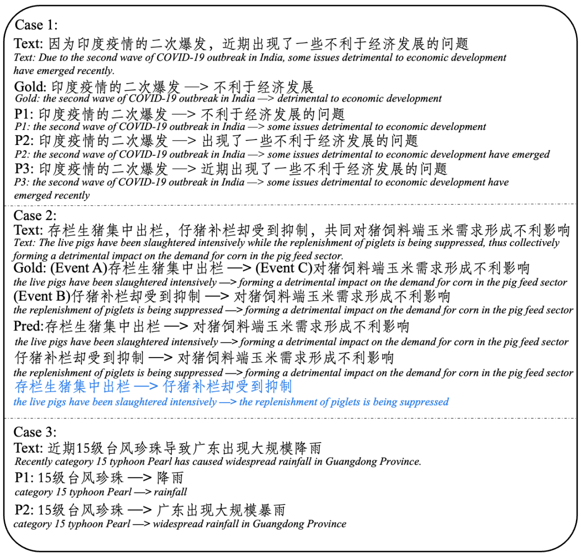
Keywords: Event Extraction, Large Language Model, Knowledge Graph

1. Introduction

Event Causality Extraction (ECE) aims to extract causal relations between event pairs, in which each event is presented as a continuous span within the sentences or documents. Abundant downstream application tasks can be facilitated after extracting event causality from text, including event detection (Weng and Lee, 2011), event prediction (Granroth-Wilding and Clark, 2016) Xu et al. (2020), logical reasoning (Tappin et al., 2020), question answering (Karpukhin et al., 2020), and constructing an event logic graph (Ding et al., 2019) Gao et al. (2022).

Given plain text, an ECE system is responsible for extracting event spans and matching them by causality. Previous works (Yang et al., 2022) (Lyu et al., 2022) (Zhang et al., 2022) Yang et al. (2022) Heindorf et al. (2020) in event causality extraction predominantly employ a two-stage method: event tagging and span-based event causality matching. Much progress (Yang et al., 2022) has been made in this paradigm with the development of pre-trained language models (Devlin et al., 2018). However, two challenges have not caught much attention: Event Boundary Deviation and Event Causality Mismatching.

Event Boundary Deviation: Previous methods struggle to predict causal event boundaries, resulting in redundant or missing words. As shown in Fig 1 Case 1, a typical ECE model makes different predictions P1 to P3 in spans of the effect, but all predictions describe the same event as labeled in GOLD. We explore the origin of the event boundary deviation phenomenon from two perspectives: concluding practical experimental experience and digging deep into the principle of the ECE task.


Case 1:
Text: 因为印度疫情的二次爆发，近期出现了一些不利于经济发展的问题
Text: Due to the second wave of COVID-19 outbreak in India, some issues detrimental to economic development have emerged recently.
Gold: 印度疫情的二次爆发 —> 不利于经济发展
Gold: the second wave of COVID-19 outbreak in India —> detrimental to economic development
P1: 印度疫情的二次爆发 —> 不利于经济发展的问题
P1: the second wave of COVID-19 outbreak in India —> some issues detrimental to economic development
P2: 印度疫情的二次爆发 —> 出现了一些不利于经济发展的问题
P2: the second wave of COVID-19 outbreak in India —> some issues detrimental to economic development have emerged
P3: 印度疫情的二次爆发 —> 近期出现了一些不利于经济发展的问题
P3: the second wave of COVID-19 outbreak in India —> some issues detrimental to economic development have emerged recently

Case 2:
Text: 存栏生猪集中出栏，仔猪补栏却受到抑制，共同对猪饲料端玉米需求形成不利影响
Text: The live pigs have been slaughtered intensively while the replenishment of piglets is being suppressed, thus collectively forming a detrimental impact on the demand for corn in the pig feed sector.
Gold: (Event A)存栏生猪集中出栏 —> (Event C)对猪饲料端玉米需求形成不利影响
the live pigs have been slaughtered intensively —> forming a detrimental impact on the demand for corn in the pig feed sector
(Event B)仔猪补栏却受到抑制 —> 对猪饲料端玉米需求形成不利影响
the replenishment of piglets is being suppressed —> forming a detrimental impact on the demand for corn in the pig feed sector
Pred:存栏生猪集中出栏 —> 对猪饲料端玉米需求形成不利影响
the live pigs have been slaughtered intensively —> forming a detrimental impact on the demand for corn in the pig feed sector
仔猪补栏却受到抑制 —> 对猪饲料端玉米需求形成不利影响
the replenishment of piglets is being suppressed —> forming a detrimental impact on the demand for corn in the pig feed sector
存栏生猪集中出栏 —> 仔猪补栏却受到抑制
the live pigs have been slaughtered intensively —> the replenishment of piglets is being suppressed

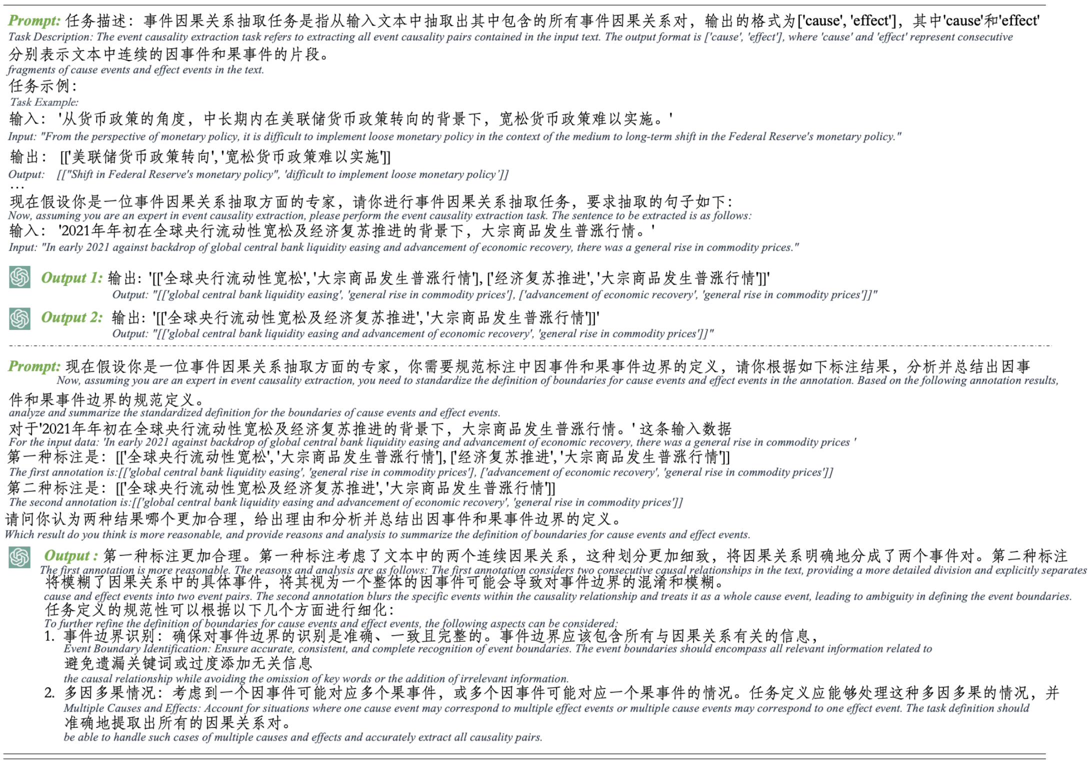
Case 3:
Text: 近期15级台风珍珠导致广东出现大规模降雨
Recently category 15 typhoon Pearl has caused widespread rainfall in Guangdong Province.
P1: 15级台风珍珠 —> 降雨
category 15 typhoon Pearl —> rainfall
P2: 15级台风珍珠 —> 广东出现大规模暴雨
category 15 typhoon Pearl —> widespread rainfall in Guangdong Province

Figure 1: Case study in ECE.

In the process of the case study in preliminary experiments, we observe frequent inconsistent annotations of ECE datasets. As shown in Table

*Corresponding author

Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ LREC-Coling 2024, pages 33–44
21 May, 2024.



Prompt: 任务描述：事件因果关系抽取任务是指从输入文本中抽取出其中包含的所有事件因果关系对，输出的格式为['cause', 'effect']，其中'cause'和'effect'
Task Description: The event causality extraction task refers to extracting all event causality pairs contained in the input text. The output format is ['cause', 'effect'], where 'cause' and 'effect' represent consecutive
分别表示文本中连续的因事件和果事件的片段。
fragments of cause events and effect events in the text.

任务示例：
Task Example:

输入：'从货币政策的角度，中长期内在美联储货币政策转向的背景下，宽松货币政策难以实施。'
Input: "From the perspective of monetary policy, it is difficult to implement loose monetary policy in the context of the medium to long-term shift in the Federal Reserve's monetary policy."

输出： [['美联储货币政策转向','宽松货币政策难以实施']]
Output: [["Shift in Federal Reserve's monetary policy", 'difficult to implement loose monetary policy']]

…

现在假设你是一位事件因果关系抽取方面的专家，请你进行事件因果关系抽取任务，要求抽取的句子如下：
Now, assuming you are an expert in event causality extraction, please perform the event causality extraction task. The sentence to be extracted is as follows:

输入：'2021年年初在全球央行流动性宽松及经济复苏推进的背景下，大宗商品发生普涨行情。'
Input: "In early 2021 against backdrop of global central bank liquidity easing and advancement of economic recovery, there was a general rise in commodity prices."

Output 1: 输出：'[['全球央行流动性宽松','大宗商品发生普涨行情'], ['经济复苏推进','大宗商品发生普涨行情']]'
Output: "[['global central bank liquidity easing', 'general rise in commodity prices'], ['advancement of economic recovery', 'general rise in commodity prices']]"

Output 2: 输出：'[['全球央行流动性宽松及经济复苏推进','大宗商品发生普涨行情']]'
Output: "[['global central bank liquidity easing and advancement of economic recovery', 'general rise in commodity prices']]"

Prompt: 现在假设你是一位事件因果关系抽取方面的专家，你需要规范标注中因事件和果事件边界的定义，请你根据如下标注结果，分析并总结出因事件和果事件边界的规范定义。
Now, assuming you are an expert in event causality extraction, you need to standardize the definition of boundaries for cause events and effect events in the annotation. Based on the following annotation results, analyze and summarize the standardized definition for the boundaries of cause events and effect events.

对于'2021年年初在全球央行流动性宽松及经济复苏推进的背景下，大宗商品发生普涨行情。'这条输入数据
For the input data: 'In early 2021 against backdrop of global central bank liquidity easing and advancement of economic recovery, there was a general rise in commodity prices '

第一种标注是：[['全球央行流动性宽松','大宗商品发生普涨行情'], ['经济复苏推进','大宗商品发生普涨行情']]
The first annotation is:[['global central bank liquidity easing', 'general rise in commodity prices'], ['advancement of economic recovery', 'general rise in commodity prices']]

第二种标注是：[['全球央行流动性宽松及经济复苏推进','大宗商品发生普涨行情']]
The second annotation is:[['global central bank liquidity easing and advancement of economic recovery', 'general rise in commodity prices']]

请问你认为两种结果哪个更加合理，给出理由和分析并总结出因事件和果事件边界的定义。
Which result do you think is more reasonable, and provide reasons and analysis to summarize the definition of boundaries for cause events and effect events.

Output : 第一种标注更加合理。第一种标注考虑了文本中的两个连续因果关系，这种划分更加细致，将因果关系明确地分成了两个事件对。第二种标注将模糊了因果关系中的具体事件，将其视为一个整体的因事件可能会导致对事件边界的混淆和模糊。
The first annotation is more reasonable. The reasons and analysis are as follows: The first annotation considers two consecutive causal relationships in the text, providing a more detailed division and explicitly separates cause and effect events into two event pairs. The second annotation blurs the specific events within the causality relationship and treats it as a whole cause event, leading to ambiguity in defining the event boundaries.

任务定义的规范性可以根据以下几个方面进行细化：
To further refine the definition of boundaries for cause events and effect events, the following aspects can be considered:

1. 事件边界识别：确保对事件边界的识别是准确、一致且完整的。事件边界应该包含所有与因果关系有关的信息，避免遗漏关键词或过度添加无关信息
   Event Boundary Identification: Ensure accurate, consistent, and complete recognition of event boundaries. The event boundaries should encompass all relevant information related to the causal relationship while avoiding the omission of key words or the addition of irrelevant information.
2. 多因多果情况：考虑到一个因事件可能对应多个果事件，或多个因事件可能对应一个果事件的情况。任务定义应能够处理这种多因多果的情况，并准确地提取出所有的因果关系对。
   Multiple Causes and Effects: Account for situations where one cause event may correspond to multiple effect events or multiple cause events may correspond to one effect event. The task definition should be able to handle such cases of multiple causes and effects and accurately extract all causality pairs.

Figure 2: Prompts and responses for task definition generation.

1, there exists a large proportion of labeling inconsistencies in both typical Chinese and English ECE datasets. These inconsistencies confused the model in event boundary predictions trained in these datasets. However, we argue that labeling mistakes are ineluctable. As the model prediction case shown in Fig 1 Case 1, a causal event expressed in span form with consecutive words exists in multiple reasonable variants. In addition, previous research overlooked the explicit definition and annotation guidelines in event causality extraction, hindering the restoration process in these datasets. To this end, clarifying the ECE task definition and fixing inconsistencies in datasets are the primary goals.

On the other hand, we dig into the reasonable span variants of the event. First, each event composed of a continuous span within the input text exhibits multiple literal forms that depict the event with different emphases. Therefore, modeling the event with a specific span fails to capture its overall perspective. However, previous works employ a particular span on behalf of the event, having an inherent shortage of capturing the entirety. Furthermore, building an association between cause and effect events when predicting event span boundaries is essential. As illustrated in Fig 1 Case 3, for the first prediction, "category 15 Typhoon Pearl" cause "rain" constitutes a reasonable but rough causal event pair when independently predicting event span boundaries. But when considering the interdependence between the causal event pair, for the second prediction given the level and name of the typhoon in the cause event, the effect event should include specific rainfall locations and intensity. However, previous works restricted to predicting causal event span boundaries independently, lacking in the consideration of causal associations.

Event Causality Mismatching: After the extraction of potential causal events, the next step is matching cause and effect events with semantics and knowledge. Previous methods will usually face an inevitable challenge, which is mismatching two events by event span. As illustrated in Fig 1 Case 2, a human always estimates causality between event pairs from two perspectives: semantic information and contextual information. From the semantic perspective, based on their common sense and linguistics knowledge, humans can evaluate event causality based on span. However, the final decision cannot be divorced from contextual information aside from causal events, such as conjunctions, background, and correlations. Unfortunately, previous studies focused on modeling the semantic information inside event pairs, neglecting the crucial role of contextual information outside. This flaw in design could lead to confusion for models when tackling complex causal event pair-matching cases. Fig 1 Case 2 illustrates an example where the model incorrectly predicts a causal relationship between events A and B due to their perceived semantic similarity. However, leveraging contextual information, we can determine that there is no causal relationship between events A and B, but rather that events A and B cause event C, simultaneously.

In this paper, we utilize LLM to optimize the task definition, evolve the datasets, and strengthen our proposed customized event causality extraction framework to address the above issues. We introduce a pattern that applies LLM to conclude a task definition and annotation criteria according to the case of labeling inconsistency. And then automatically fix datasets by LLM based on their viewpoint. Apart from the foundation of the task, we construct a Contextual Highlighting Event Causality Extraction framework (CHECE). Specifically, we propose an Event Highlighter to represent an event independent of a specific span, and an Event Concretization Module to predict a single event boundary based on its causal counterpart. Together deal with the event boundary deviation from these three aspects. To deal with the event causality mismatching problem, we propose a Contextual Event Causality Matching mechanism. And to further ensure the model learns from context correlation, we utilize LLM to diversify the context templates.

The contributions of this paper are as follows:

1) We propose a paradigm to utilize LLM to clarify the event causality extraction task annotation and fix existing datasets. And we release the metrics and datasets to promote the relevant research.
2) To handle the event boundary deviation, we propose an Event Highlighter and an Event Concretization Module, guiding the model to represent the event by a higher-level cluster and consider its causal counterpart in event boundary prediction. To tackle the event causality mismatching, we devise a Contextual Event Causality Matching mechanism and apply LLM to diversify the content templates to force the model to learn causality from context.
3) Experiments on both Chinese and English event causality extraction datasets show our method outperforms state-of-the-art methods, especially in our new metrics.

2. The Annotation Clarification and Dataset of Event Causality Extraction

Due to the frequent inconsistent annotations of ECE datasets and their inevitability, which leads to confusion in the final model, we explore clarifying and aligning the dataset annotation with the assistance of LLM, which is well-aligned with the given annotating requirements and requires much less labor compared with human annotators.

Dataset	Manual Label	After Fix
CFC	85%	92%
FinCR	87%	93%

Table 1: Statistics of labeling accuracy before and after dataset evolution.

2.1. Annotation Clarification by LLM

We clarify the annotation criteria with the assistance of the Large Language model(LLM). Taking into account the presence of inconsistent annotated data, we employ the LLM (specifically, text-divinci-003) to generate multiple predictions, preserving each distinct output. Thanks to the rich knowledge that LLM contains and its great ability to follow given instructions, the LLM is able to analyze which of the various outputs it predicts makes the most sense, thereby establishing the essential attribute that should define the event boundary judgment. As shown in Fig 2, the prompts are organized in the format of the chain of thought (Wang et al., 2022b). Through the above process on several sets of inconsistent annotated data, the annotation criteria of event boundary can be concluded, which can be used to create instructions for the LLM to perform dataset repairment following these standards subsequently.

2.2. Measurement

Event Boundary Deviation arises due to the ambiguous task definition and inconsistent dataset annotations, as well as the inherent multivariate nature of events. We propose Easy F1 to measure the model performance more reasonably. In Easy F1, a predicted causal event span is considered correct if its similarity with the gold span surpasses a predefined threshold. The choice of this threshold can be adapted to the data distribution, and we set it at 80 percent. In the English dataset the similarity is measured by tokens, whereas in the Chinese dataset, the similarity is measured by word segmentations.

2.3. Fix Dataset by LLM

We set the concluded task's definitions into prompts and ask the LLM to repair the dataset as required. In the example of concluded event boundary definition *"causal event should be fine-grained, which means the output span cannot contain more than one event and should include all the words describing the same event"*. We set the obtained definition to the requirements, and give three shots

Dataset	Train	Dev	Test	Pairs	Text Length	Causal Distance	Span Length
CFC	2000	250	250	2.17	41	3.4	9.67
FineCR	12541	1583	1557	1.14	69	8.3	13.38

Table 2: Statistics of two ECE datasets.

of manual repair of data according to the requirements. Then we ask the LLM to determine whether the event boundary in the data meets the definition according to the requirements, if not, it needs to be corrected and explain the reason. The prompt examples are shown in the Appendix.

3. Method

In this section, we first formally define the event causality extraction task and then elaborate on each component of our model. The overall architecture of our ECE framework is shown in Fig 3.

3.1. Problem Definition

The input sentence is $X = \{x_1, x_2, ..., x_n\}$ with n tokens. Let $S = \{s_1, s_2, ..., s_n\}$ be all the possible spans in X. The desired outputs are causal event pairs as $T(X) = \{(c, e)|c, e \in S\}$, where c and e are the cause event and effect event presented as continuous spans in the input text.

The problem is decomposed into two parts, first identifying the candidate cause events and effect events and then assessing causality within event pairs formed by combining all candidate cause events with candidate effect events.

3.2. Span Proposal

Given the input sentence X, to obtain the representation of each token, we use a pre-trained language model (PLM) as our sentence encoder. The output is

$$\{h_1, h_2, \dots, h_n \mid h_i \in \mathbb{R}^{d\times 1}\} \tag{1}$$

where d is the embedding dimension, and n is the number of tokens.

Then we judge each s_i in S whether it is a causal event span following the previous span-based method (Su et al., 2022), which uses a global scoring matrix that considers the beginning and the end positions of spans to predict all the candidate cause(effect) spans. It's worth noting that the casual event spans predicted by the Span Proposal Model are not exact events, they may be part of the event lacking some boundary components or they may include the event. In other words, these spans reflect different emphases of the event.

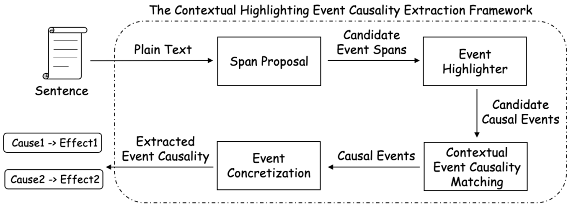


Figure 3: The overall framework of CHECE.

With the obtained sentence representation, using two feedforward layers that rely on the begin and end indices of the span:

$$q_i = W_q h_i + b_q \tag{2}$$

$$k_j = W_k h_j + b_k \tag{3}$$

where $q_i \in \mathbb{R}^d, k_j \in \mathbb{R}^d$ denote the vector representations of the start and end positions. The score $p_{i,j}$ indicating the score of span $s[i : j]$ that starts with i being a cause(effect) span is computed as follows:

$$p_{i,j} = \sigma(q_i^\top k_j) \tag{4}$$

where σ is the sigmoid function. Then we set a threshold μ for the predicted score. We consider the span $s[i : j]$ as a candidate cause(effect) span if $p_{i,j}$ exceeds the threshold value.
Class Imbalance Loss is introduced to the training process $\mathcal{L}_s$:

$$\log(1 + \sum_{(q,k)\in P} e^{-p_{q,k}}) + \log(1 + \sum_{(q,k)\in Q} e^{p_{q,k}}) \tag{5}$$

where q, k represent the start and tail indexes of a span, P represents a collection of spans that are considered candidate cause(effect) spans, Q represents a collection of spans that are not candidate cause(effect) spans.

3.3. Event Highlighter

The Event Highlighter aims to build better representations for events. Due to the inherent uncertainty and multivariate nature of events expressed with natural language, employing a single specific span to model the target event yields multiple candidates with varying boundaries for a given event, thereby significantly occupying the search space and inducing model confusion during the matching of event causality. To this end, we propose a cluster-based event highlighter model to catch the overall perspective and significance of events, exploring to model the event at event-level instead of span-level.

After obtaining all the candidate cause spans and effect spans $s[i : j]$ together with their scores $p_{i,j}$, the event highlighter captures the complete picture and emphasis of the target event E. Specifically, there may be multiple candidate spans that have a slight boundary deviation from the target span (e.g., a different adjective), all describing the same target event E but with different emphases. We use the clustering method to gather the spans that describe the same event. First, we evaluate the token similarity between each span and the target event centered on the target event E. If the similarity exceeds the threshold λ, the span is considered to describe the target event and is clustered with the target event. Each target event corresponds to an event cluster $C = \{s_1, s_2, ..., s_k\}$. Where λ is an adjustable hyperparameter and k is the number of all spans describing the same target event obtained after span clustering.

The representation of a single span is acquired as follows.

$$\boldsymbol{h}_{s_i} = Avgpool(\boldsymbol{h}_{START(i):END(i)}) \tag{6}$$

where $Avgpool$ is the average pooling operation(Lin et al., 2013), $START(i)$ and $END(i)$ denote the start and end indices of the candidate span s_i.

Then the Event Highlighter combines all the candidate spans describing the same target event, weighted by their score in the span proposal model to find the most important tokens of the event and see the full event description covering the longest boundary:

$$\boldsymbol{h_E} = \sum_{1}^{k} p_k \boldsymbol{h}_{s_k} \tag{7}$$

where k is the number of all spans describing the same target event obtained after span clustering.

3.4. Contextual Event Causality Matching

The Event Causality matching model aims to take a pair of cause event E_c and effect event E_e as input and predict whether there is a causal relationship. Previous works concatenate the representations of event pairs and put them into the feedforward layer for causality judgment, which only considers the semantic representation. We argue that the explicit use of contextual information plays an important role in causal judgment. As illustrated in Fig 1, only relying on the semantic representation of events, it is easy to mistakenly judge that there is a causal relationship between events A and B due to their semantic similarity. But in fact, combining the context structure information, we can judge that there is no causal relationship between A and B, and it is A and B that cause C together.

To this end, we propose utilizing the semantic information and context information jointly to evaluate the causality of input event pairs. First, the semantic representations of input event pairs are obtained as:

$$\psi_{sem}(E_c, E_e) = W_{sem}[\boldsymbol{h}_{E_c}; \boldsymbol{h}_{E_e}] + b_{sem} \tag{8}$$

where $\boldsymbol{h}_{E_c}$ and $\boldsymbol{h}_{E_e}$ are the event representation obtained in the Event Highlighter model, W_{sem} and b_{sem} are trainable parameters, and $[A; B]$ denotes the concatenation operation.

Next, we turn to obtain the explicit contextual representation. The mask token is used to replace the position of the event's original tokens and sent to the BERT encoder to let the model pay attention to the context information other than the semantics of the specific token. Then the output mask token is used as the contextual representation, rich in context structure information:

$$\psi_{con}(E_c, E_e) = W_{con}[\boldsymbol{m}_{E_c}; \boldsymbol{m}_{E_e}] + b_{con} \tag{9}$$

where $\boldsymbol{m}_{E_c}, \boldsymbol{m}_{E_e}$ are the contextual representation obtained from the mask token, W_{con} and b_{con} are trainable parameters.

With semantic representation and contextual representation, we model the judgment of event causality jointly by combing the two parts of information using a hyperparameter θ.

$$\psi(E_c, E_e) = \theta\psi_{sem}(E_c, E_e) + (1 - \theta)\psi_{con}(E_c, E_e) \tag{10}$$

where $\psi(E_c, E_e)$ is the score for cause event E_c and effect event E_e to be a pair of causal events. Then we set a threshold υ for the predicted score. We consider cause event E_c and effect event E_e to be a pair of causal events if $p_{i,j}$ exceeds the threshold value.

During the training process, the loss is considered as follows:

$$\mathcal{L}_e = -\sum_{E_i \in E_c, E_j \in E_e} \log P\left(R^*_{i,j} \mid E_i, E_j\right) \tag{11}$$

where $R^*_{i,j}$ represents the gold relation type of event pair.

To better utilize the contextual information, we also augment the training data by constructing template training data. Specifically, we replace each causal event pair of the data in the training set with $[cause]$ and $[effect]$ to form a contextual template that preserves only structural information. We store all the templates in a file. Furthermore, we employ ChatGPT[1] to generate causal event pairs and causal templates in order to harvest rich domain knowledge and diverse causal contextual information from LLM. The detailed prompt is shown in Appendix A.1. With the obtained causal event pairs and causal templates, we synthesize extra data during the training process by replacing the $[cause]$ and $[effect]$ in the chosen template from

[1] https://chat.openai.com/chat

the stored file with a random pair of causal events in the training set or the causal event pairs generated by ChatGPT to enhance the model's ability capacity in capturing contextual information and injecting domain knowledge into the model.

3.5. Event Concretization Module

The Event Concretization Module aims to reify the event pairs judged to have causal relations in the last step from the abstract event representation to the concrete event span. In other words, given a pair of input causal event clusters and their representations, the Event Concretization Module needs to output the most suitable cause span and effect span that best represents the target causal event pair as the final extraction result.

Previous works consider the spans that score higher than the preset threshold in the Span Proposal Module as predicted events. There are two possible disadvantages to this practice: First, multiple spans with slight boundary differences are referred to as the same event, thereby disentangling the information inside the event and inevitably introducing subsequent matching errors. Furthermore, this practice cut off the connections between causal event pairs. As illustrated in Figure 2, judging the boundaries of cause or effect events separately ignores the overall connection of causal events and is error-prone.

Given a pair of input cause and effect event cluster $C_c = \{s_1, s_2, ..., s_m\}$ and $C_e = \{s_1, s_2, ..., s_n\}$ with their representations. First, iterate over each span s_i in the cause event cluster C_c and build its connection with the effect event cluster C_e by concatenating their representation and put into a feedforward network:

$$P_{s_i} = \sigma(W_{concre}[\boldsymbol{h}_{s_i}; \boldsymbol{h}_{E_e}] + b_{concre}) \tag{12}$$

where σ is the sigmoid function, $\boldsymbol{h}_{s_i}$ and $\boldsymbol{h}_{E_e}$ are the span representation of s_i and event representation of E_e. We choose the span with the highest score P_{s_i} in the cause event cluster as the final output cause event. During the training process, the loss is considered as follows:

$$\mathcal{L}_c = -\sum_{s_i \in C_c} \log P\left(r^*_{i,j} \mid s_i\right) \tag{13}$$

where $r^*_{i,j}$ represents the gold type of span s_i which means whether the span s_i is the gold span to represent the cause event. Event Concretization for the effect event cluster is conducted in a symmetric way.

Dataset	Method	Dev		Test	
		Easy-F1	Hard-F1	Easy-F1	Hard-F1
CFC	BERT-CRF	54.94	42.81	53.29	38.54
	GlobalPointer	59.49	51.18	61.96	53.84
	TP-Linker	62.81	53.39	62.28	53.97
	PL-Marker	63.89	53.06	63.71	54.65
	ChatGPT	-	-	31.39	12.56
	Ours	64.40	53.86	63.81	55.24
	Ours+LLM	**64.88**	**55.09**	**65.63**	**55.73**
FineCR	BERT-CRF	55.12	35.60	54.92	35.58
	GlobalPointer	55.72	39.89	54.76	38.97
	TP-Linker	56.21	40.05	56.60	39.39
	PL-Marker	57.99	40.14	58.75	39.90
	ChatGPT	-	-	17.68	7.62
	Ours	58.85	40.37	59.77	40.21
	Ours+LLM	**59.91**	**40.47**	**60.61**	**40.55**

Table 3: Comparison of our model and other baselines on two event causality extraction datasets. We test ChatGPT with 3-shot task examples and task descriptions. *"Ours+LLM"* means our full model with ChatGPT data augmentation.

3.6. Training Strategy

We adopt a joint training approach, wherein we optimize the combined objective function throughout the training process while sharing the parameters of the BERT encoder. The total loss is the sum of these three parts:

$$\mathcal{L}_{total} = \omega_1\mathcal{L}_s + \omega_2\mathcal{L}_e + \omega_3\mathcal{L}_c \tag{14}$$

Performance might be better by carefully tuning the weight of each sub-loss, but we just assign equal weights for simplicity (i.e., ω_1 = ω_1 = ω_1 = 1).

4. Experiments

4.1. Datasets and Preprocessing

We conduct experiments on FineCR and CFC (Yang et al., 2022) proposed in Section 2 to verify the effectiveness of our method. FineCR is a widely used dataset in English. The experiments and analysis on it could be regarded as fair comparisons with previous works. CFC is a more challenging dataset with more ambiguous causal event spans and multiple complicated causalities in a single sentence. The detailed statistical information and split information are shown in Table 2.

4.2. Metrics and Parameter Settings

For automatic evaluation, we utilize easy F1 and hard F1 introduced in Section 2. Since previous methods in full tagging paradigm apply token-wise tag F1 score to report the performance, to fairly compare our performance with baselines, we reproduce these methods and report our metrics.

We use bert-base-uncased (Devlin et al., 2018) and chinese-roberta-wwm-ext (Cui et al., 2021) as the base encoders for the English dataset FineCR and the Chinses dataset CFC. The learning rate is set as 3e-5 in the backbone of BERT. We set the max length of the input sentence to 200/75 for

CFC and FineCR. The batch size is set as 16. We train the model for at most 30 epochs and choose the model with the best performance on the dev set to output results on the test set.

4.3. Baselines

We compare our method with the following baselines:

BERT-CRF(Yang et al., 2022): BERT-CRF is a powerful model that combines BERT's contextual understanding with CRF's sequential tagging for accurate squeue tagging.

GlobalPointer(Su et al., 2022): GlobaoPointer is a span-based method using a global scoring matrix that considers the beginning and the end positions of spans with a global view.

TP-Linker(Wang et al., 2020): TP-Linker is a one-stage joint entity and relation extraction model. We use it to extract event spans that have larger granularity than entities thus bringing great challenges to the model.

PL-Marker(Ye et al., 2021): PL-Marker proposed a novel span representation approach to consider the interrelation between the spans (pairs) by strategically packing the markers in the encoder and achieving SOTA performance in the entity and relation extraction task.

ChatGPT: ChatGPT is a large language model developed by OpenAI which has strong zero-shot and few-shot learning abilities. However, it struggles in the difficult task such as causal relation extraction that requires more comprehensive commonsense knowledge and higher logical reasoning ability. We test the model with a task description and three-shot task examples.

4.4. Compared with State-of-the-art Methods

Table 3 shows the results of our method on two event causality extraction datasets. Overall, our method achieves the best performance from these baselines. Indicating our method's effectiveness and advancement. Specifically, compared among full sequence tagging methods, whatever the tagging schema setting, PLMs help them achieve better performances. However, comparing ChatGPT-Gen with other baselines, we can draw a conclusion that the performance of LLM in this task is inferior to supervised training models. It could be the reason that the complex and specific demands in the ECE task hinder the release of LLM's extensive capacity. Transferring methods in joint extraction of entities and relations to ECE, TP-Linker(Wang et al., 2020) and PL-Marker(Ye et al., 2021) achieve higher f1 than full tagging methods. Prove they can model the span representation and span relation better than plain tagging. Our method obtained better performance in easy and hard f1 than TP-Linker and PL-Marker, which struggle to extract events that have larger granularity than entities. It demonstrated the proposed Event Highlighter and Contextual Causal Event Matching is more customized in this task and could deal with Event Boundary Deviation and Event Causality Mismatching.

Method	CFC		FineCR	
	Easy-F1	Hard-F1	Easy-F1	Hard-F1
Ours	**65.63**	**55.73**	**60.61**	**40.55**
w/o event highlighter	60.11	53.37	52.15	38.12
w/o causal event matching	63.91	53.93	58.61	39.27
w/o LLM template	63.81	55.24	59.75	40.21
with 500 Augment Causality	64.08	53.74	-	-
with 1000 Augment Causality	65.63	55.73	-	-
with 1500 Augment Causality	63.43	55.12	-	-

Table 4: Ablation results on the CFC and FineCR test set.

4.5. Ablation Experiments

To investigate the effectiveness of our proposed components in the method, we also perform ablation experiments on the CFC and FineCR datasets. The ablation results are shown in Table 4, indicating that none of these models can achieve a comparable result with our full version. Demonstrate that all those factors contribute a certain improvement to our model.

Specifically, when we discard the whole event highlighter part, and represent an event with a specific span (Ours w/o event highlighter), the performance drops demonstrate the effectiveness of the event highlighter. In Ours w/o causal event matching, we calculate causal pair score only referring to event representation and ignore the contextual information. The suboptimal performance demonstrates the effectiveness of contextual causal event matching.

We employ ChatGPT to generate causal event pairs and incorporate them into the constructed template to synthesize new training data as introduced in Section 3.4, injecting knowledge and contextual information into the model simultaneously. To further explore the effectiveness of event pair augment from LLM, we attempted varying template count N utilized during training. The bottom of Table 4 shows, when n is zero, the easy f1 drop X from the full model, indicates the effectiveness of the event pair augments from LLM. In addition, the model performance could not improve with the increase of N after N is larger than 1000, manifesting that the templates generated by LLM is varies considerably in quality.

5. Related Work

Event Causality Extraction. Previous works use feature-based methods for event causality extraction. (Ittoo and Bouma, 2011) proposes a method for extracting causal pairs by leveraging part-of-speech analysis, syntactic analysis, and causality templates. (Hashimoto et al., 2014) uses semantic relation (between nouns), context, and association features to extract event causalities from the web. In recent years, deep learning techniques employed in causality extraction. (Li et al., 2021) uses the BiLSTM-CRF model as the backbone to extract cause and effect directly, formulating the task in the causality tagging scheme. (Wang et al., 2022a) proposed a model that aims to transform event causality extraction into causal argument extraction, by incorporating both sentence-level and document-level contextual information. Recently, much progress has been made in this task with the strong language modeling capabilities and rich world knowledge of pre-trained language models (PLMs) (Devlin et al., 2018). (Fajcik et al., 2022) used T5 to identify all cause-effect-signal span triplets. (Yang et al., 2022) and (Lyu et al., 2022) use BERT-CRF model in Fine-grained Event Causality Extraction and FinCausal 2022 tasks, resulting in significant advancements.

LLMs Assist Tasks. The capability of Large Language Models (LLMs) like ChatGPT to comprehend user intent and provide reasonable responses has made them extremely popular lately. Recent studies show that the latest LLMs have the ability to do Information Extraction tasks such as Named Entity Recognization(NER), Relation Extraction(RE), and Event Extraction(EE). (Xu et al., 2023) proposed task-related instructions and schema-constrained data generation to enhance LLM's few-shot relation extraction performance. (Tang et al., 2023) used LLM's rich domain knowledge to induce new event schemas. Some works utilize LLM to improve the performance of downstream tasks. (Dai et al., 2023) and (Ubani et al., 2023) leveraged ChatGPT for text data augmentation and synthetic training data generating to induce extensive knowledge.

6. Conclusion

This paper proposes to utilize LLM to generate the definition of event causality extraction tasks and automatically evolve the datasets. Lay the foundation for further research and improvement. We propose a framework called CHECE to deal with two unaddressed problems. Specifically, the Event Highlighter and an Event Concretization Module, guide the model to represent the event by a higher-level cluster and consider its causal counterpart in event boundary prediction to deal with event boundary deviation. And the Contextual Event Causality Matching mechanism forces the model to predict causality from context information to overcome the causal event pair mismatching issue. Meanwhile, we apply LLM to diversify the content templates to enhance this side. Experimental results on two ECE datasets demonstrate the effectiveness of the method.

7. Ackonwledgements

This work is supported by the National Key Research and Development Program of China (No. 2022ZD0160503), the National Natural Science Foundation of China (No. 62176257). This work is also supported by Yunnan Provincial Major Science and Technology Special Plan Projects (No.202202AD080004).

Limitations

Our work is not without limitations. From the LLM side, on the one hand, the best prompt or the chain of thought for the conclusion of task definition by LLM is under-explored. We believe there exists a better way for LLM to generate the definition and further utilize it to evolve the datasets. On the other hand, this paradigm could produce more labeled data from news or documents from the web. Release a larger dataset remains in our future work. From the framework side, although effective our method is slightly complicated. How to address the above two challenges more concisely is a worth exploring topic.

8. Bibliographical References

Wajid Ali, Wanli Zuo, Wang Ying, Rahman Ali, Gohar Rahman, and Inam Ullah. 2023. Causality extraction: A comprehensive survey and new perspective. *Journal of King Saud University-Computer and Information Sciences*, page 101593.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, et al. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xiao Ding, Zhongyang Li, Ting Liu, and Kuo Liao. 2019. Elg: an event logic graph. *arXiv preprint arXiv:1907.08015*.

Martin Fajcik, Muskaan Singh, Juan Zuluaga-Gomez, Esaú Villatoro-Tello, Sergio Burdisso, Petr Motlicek, and Pavel Smrz. 2022. Idiapers@ causal news corpus 2022: Extracting cause-effect-signal triplets via pre-trained autoregressive language model. *arXiv preprint arXiv:2209.03891*.

Jianqi Gao, Hang Yu, and Shuang Zhang. 2022. Joint event causality extraction using dual-channel enhanced neural network. *Knowledge-Based Systems*, 258:109935.

Jinglong Gao, Xiao Ding, Bing Qin, and Ting Liu. 2023. Is chatgpt a good causal reasoner? a comprehensive evaluation. *arXiv preprint arXiv:2305.07375*.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, Motoki Sano, István Varga, Jong-Hoon Oh, and Yutaka Kidawara. 2014. Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 987–997.

Christopher Hidey and Kathleen McKeown. 2016. Identifying causal relations using parallel wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433.

Yan Hu, Iqra Ameer, Xu Zuo, Xueqing Peng, Yujia Zhou, Zehan Li, Yiming Li, Jianfu Li, Xiaoqian Jiang, and Hua Xu. 2023. Zero-shot clinical entity recognition using chatgpt. *arXiv preprint arXiv:2303.16416*.

Ashwin Ittoo and Gosse Bouma. 2011. Extracting explicit and implicit causal relations from sparse, domain-specific texts. In *Natural Language Processing and Information Systems: 16th International Conference on Applications of Natural Language to Information Systems, NLDB 2011, Alicante, Spain, June 28-30, 2011. Proceedings 16*, pages 52–63. Springer.

Xianxian Jin, Xinzhi Wang, Xiangfeng Luo, Subin Huang, and Shengwei Gu. 2020. Inter-sentence and implicit causality extraction from chinese corpus. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I 24*, pages 739–751. Springer.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Christopher SG Khoo, Syin Chan, and Yun Niu. 2000. Extracting causal knowledge from a medical database using graphical patterns. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 336–343.

Zhaoning Li, Qi Li, Xiaotian Zou, and Jiangtao Ren. 2021. Causality extraction based on self-attentive bilstm-crf with transferred embeddings. *Neurocomputing*, 423:207–219.

Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

Chenyang Lyu, Tianbo Ji, Quanwei Sun, and Liting Zhou. 2022. Dcu-lorcan at fincausal 2022: Span-based causality extraction from financial documents using pre-trained language models. In *Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022*, pages 116–120.

Chaveevan Pechsiri and Rapepun Piriyakul. 2010. Explanation knowledge graph construction through causality extraction from texts. *Journal of computer science and technology*, 25(5):1055–1070.

Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. 2022. Global pointer: Novel efficient span-based approach for named entity recognition. *arXiv preprint arXiv:2208.03054*.

Jialong Tang, Hongyu Lin, Zhuoqun Li, Yaojie Lu, Xianpei Han, and Le Sun. 2023. Harvesting event schemas from large language models. *arXiv preprint arXiv:2305.07280*.

Ben M Tappin, Gordon Pennycook, and David G Rand. 2020. Thinking clearly about causal inferences of politically motivated reasoning: Why paradigmatic study designs often undermine causal inference. *Current Opinion in Behavioral Sciences*, 34:81–87.

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *arXiv preprint arXiv:2304.14334*.

Longbao Wang, Li Chen, Zeyu Zhang, Yingchi Mao, Chong Long, and Yican Shen. 2022a. Event causality extraction based on fusion attention. In *2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, pages 1–5. IEEE.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. Tplinker: Single-stage joint extraction of entities and relations through token pair linking. *arXiv preprint arXiv:2010.13415*.

Jianshu Weng and Bu-Sung Lee. 2011. Event detection in twitter. In *Proceedings of the international aaai conference on web and social media*, volume 5, pages 401–408.

Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023. How to unleash the power of large language models for few-shot relation extraction? *arXiv preprint arXiv:2305.01555*.

Linyi Yang, Zhen Wang, Yuxiang Wu, Jie Yang, and Yue Zhang. 2022. Towards fine-grained causal reasoning and qa. *arXiv preprint arXiv:2204.07408*.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2021. Packed levitated marker for entity and relation extraction. *arXiv preprint arXiv:2109.06067*.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot temporal relation extraction with chatgpt. *arXiv preprint arXiv:2304.05454*.

Yujie Zhang, Rujiang Bai, Ling Kong, and Xiaoyue Wang. 2022. 2sce-4sl: A 2-stage causality extraction framework for scientific literature.

Sendong Zhao, Ting Liu, Sicheng Zhao, Yiheng Chen, and Jian-Yun Nie. 2016. Event causality extraction based on connectives analysis. *Neurocomputing*, 173:1943–1950.

9. Language Resource References

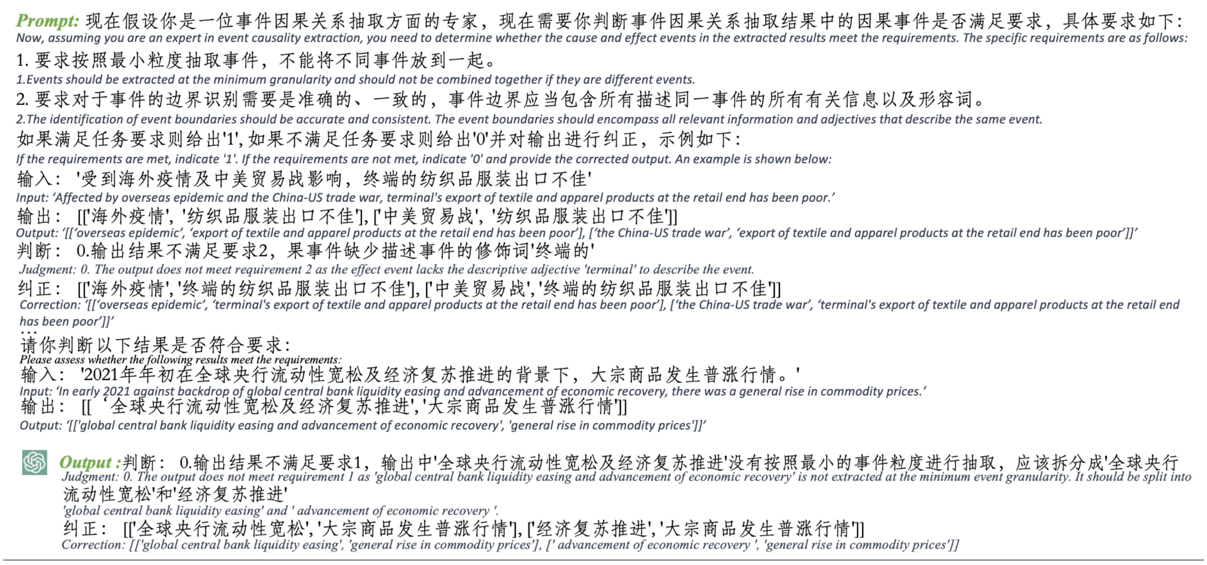
Gao, Jianqi and Luo, Xiangfeng and Wang, Hao. 2022. *Chinese causal event extraction using causality-associated graph neural network*. Wiley Online Library.

Heindorf, Stefan and Scholten, Yan and Wachsmuth, Henning and Ngonga Ngomo, Axel-Cyrille and Potthast, Martin. 2020. *Causenet: Towards a causality graph extracted from the web*.

Xu, Jinghang and Zuo, Wanli and Liang, Shining and Zuo, Xianglin. 2020. *A review of dataset and labeling methods for causality extraction*.

Yang, Linyi and Wang, Zhen and Wu, Yuxiang and Yang, Jie and Zhang, Yue. 2022. *Towards fine-grained causal reasoning and qa*.

A. Appendix



Prompt: 现在假设你是一位事件因果关系抽取方面的专家，现在需要你判断事件因果关系抽取结果中的因果事件是否满足要求，具体要求如下：
Now, assuming you are an expert in event causality extraction, you need to determine whether the cause and effect events in the extracted results meet the requirements. The specific requirements are as follows:

1. 要求按照最小粒度抽取事件，不能将不同事件放到一起。
1.Events should be extracted at the minimum granularity and should not be combined together if they are different events.

2. 要求对于事件的边界识别需要是准确的、一致的，事件边界应当包含所有描述同一事件的所有有关信息以及形容词。
2.The identification of event boundaries should be accurate and consistent. The event boundaries should encompass all relevant information and adjectives that describe the same event.

如果满足任务要求则给出'1',如果不满足任务要求则给出'0'并对输出进行纠正，示例如下：
If the requirements are met, indicate '1'. If the requirements are not met, indicate '0' and provide the corrected output. An example is shown below:

输入： '受到海外疫情及中美贸易战影响，终端的纺织品服装出口不佳'
Input: 'Affected by overseas epidemic and the China-US trade war, terminal's export of textile and apparel products at the retail end has been poor.'

输出： [['海外疫情', '纺织品服装出口不佳'],['中美贸易战', '纺织品服装出口不佳']]
Output: '[['overseas epidemic', 'export of textile and apparel products at the retail end has been poor'], ['the China-US trade war', 'export of textile and apparel products at the retail end has been poor']]'

判断： 0.输出结果不满足要求2，果事件缺少描述事件的修饰词'终端的'
Judgment: 0. The output does not meet requirement 2 as the effect event lacks the descriptive adjective 'terminal' to describe the event.

纠正： [['海外疫情','终端的纺织品服装出口不佳'],['中美贸易战','终端的纺织品服装出口不佳']]
Correction: '[['overseas epidemic', 'terminal's export of textile and apparel products at the retail end has been poor'], ['the China-US trade war', 'terminal's export of textile and apparel products at the retail end has been poor']]'

…

请你判断以下结果是否符合要求：
Please assess whether the following results meet the requirements:

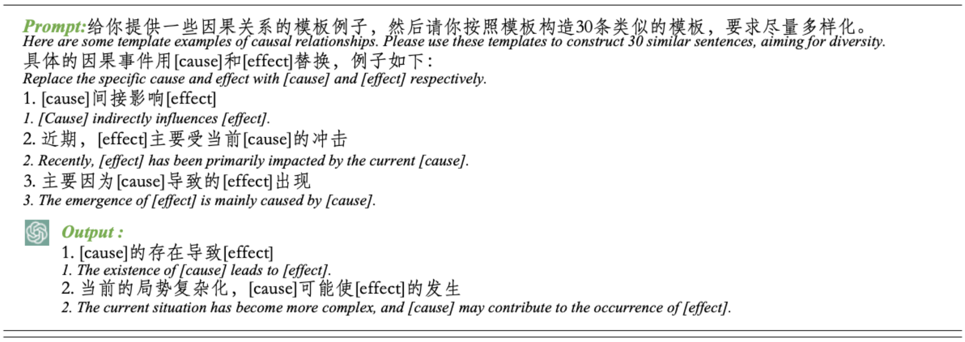
输入： '2021年年初在全球央行流动性宽松及经济复苏推进的背景下，大宗商品发生普涨行情。'
Input: 'In early 2021 against backdrop of global central bank liquidity easing and advancement of economic recovery, there was a general rise in commodity prices.'

输出： [['全球央行流动性宽松及经济复苏推进','大宗商品发生普涨行情']]
Output: '[['global central bank liquidity easing and advancement of economic recovery', 'general rise in commodity prices']]'

Output :判断： 0.输出结果不满足要求1，输出中'全球央行流动性宽松及经济复苏推进'没有按照最小的事件粒度进行抽取，应该拆分成'全球央行流动性宽松'和'经济复苏推进'
Judgment: 0. The output does not meet requirement 1 as 'global central bank liquidity easing and advancement of economic recovery' is not extracted at the minimum event granularity. It should be split into 'global central bank liquidity easing' and ' advancement of economic recovery '.

纠正： [['全球央行流动性宽松','大宗商品发生普涨行情'],['经济复苏推进','大宗商品发生普涨行情']]
Correction: [['global central bank liquidity easing', 'general rise in commodity prices'], [' advancement of economic recovery ', 'general rise in commodity prices']]

Figure 4: Prompts and responses for dataset evolution.



Prompt:给你提供一些因果关系的模板例子，然后请你按照模板构造30条类似的模板，要求尽量多样化。
Here are some template examples of causal relationships. Please use these templates to construct 30 similar sentences, aiming for diversity.

具体的因果事件用[cause]和[effect]替换，例子如下：
Replace the specific cause and effect with [cause] and [effect] respectively.

1. [cause]间接影响[effect]
1. [Cause] indirectly influences [effect].

2. 近期，[effect]主要受当前[cause]的冲击
2. Recently, [effect] has been primarily impacted by the current [cause].

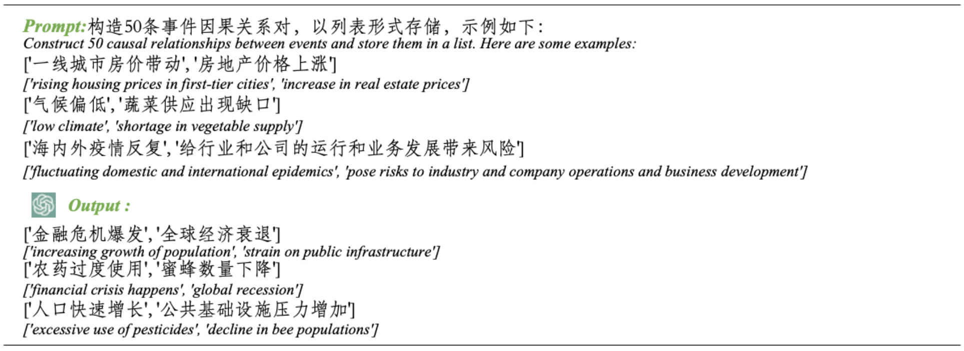
3. 主要因为[cause]导致的[effect]出现
3. The emergence of [effect] is mainly caused by [cause].

Output :

1. [cause]的存在导致[effect]
1. The existence of [cause] leads to [effect].

2. 当前的局势复杂化，[cause]可能使[effect]的发生
2. The current situation has become more complex, and [cause] may contribute to the occurrence of [effect].

Figure 5: Prompts and responses for contextual template generation.



Prompt:构造50条事件因果关系对，以列表形式存储，示例如下：
Construct 50 causal relationships between events and store them in a list. Here are some examples:

['一线城市房价带动','房地产价格上涨']
['rising housing prices in first-tier cities', 'increase in real estate prices']

['气候偏低','蔬菜供应出现缺口']
['low climate', 'shortage in vegetable supply']

['海内外疫情反复','给行业和公司的运行和业务发展带来风险']
['fluctuating domestic and international epidemics', 'pose risks to industry and company operations and business development']

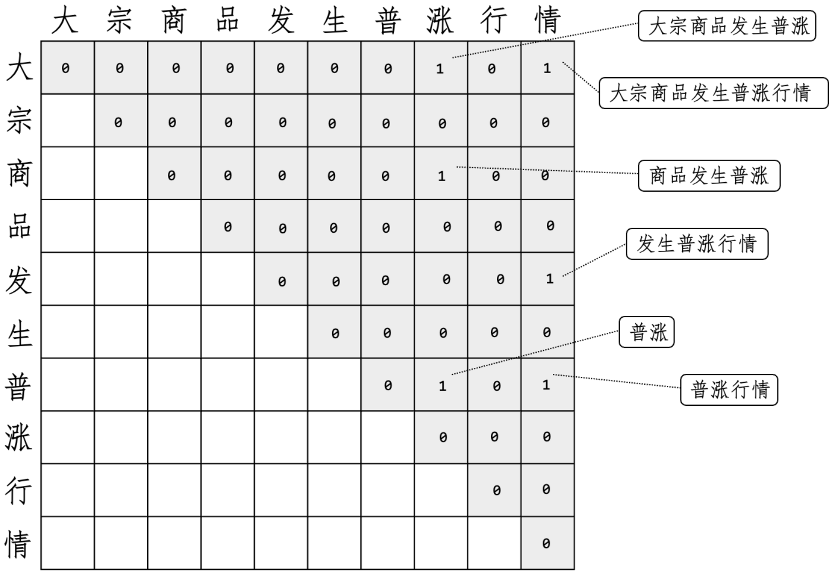
Output :

['金融危机爆发','全球经济衰退']
['increasing growth of population', 'strain on public infrastructure']

['农药过度使用','蜜蜂数量下降']
['financial crisis happens', 'global recession']

['人口快速增长','公共基础设施压力增加']
['excessive use of pesticides', 'decline in bee populations']

Figure 6: Prompts and responses for event span pair generation.



	大	宗	商	品	发	生	普	涨	行	情
大	0	0	0	0	0	0	0	1	0	1
宗		0	0	0	0	0	0	0	0	0
商			0	0	0	0	0	1	0	0
品				0	0	0	0	0	0	0
发					0	0	0	0	0	1
生						0	0	0	0	0
普							0	1	0	1
涨								0	0	0
行									0	0
情										0

Figure 7: Tagging schema in Global Pointer.

The Need for Grounding in LLM-based Dialogue Systems

Kristiina Jokinen
AI Research Center, National Institute of Advanced Industrial Science and Engineering
Tokyo, Japan
kristiina.jokinen@aist.go.jp

Abstract

Grounding is a pertinent part of the design of LLM-based dialogue systems. Although research on grounding has a long tradition, the paradigm shift caused by LLMs has brought the concept onto the foreground, in particular in the context of cognitive robotics. To avoid generation of irrelevant or false information, the system needs to ground its utterances into real-world events, and to avoid the statistical parrot effect, the system needs to construct shared understanding of the dialogue context and of the partner's intents. Grounding and construction of the shared context enables cooperation between the participants, and thus supports trustworthy interaction. This paper discusses grounding using neural LLM technology. It aims to bridge neural and symbolic computing on the cognitive architecture level, so as to contribute to a better understanding of how conversational reasoning and collaboration can be linked to LLM implementations to support trustworthy and flexible interaction.

Keywords: grounding, spoken dialogue systems, large language models, Theory of Mind, conversational AI, knowledge graphs, language-capable robots

1. Introduction

One of the main challenges in cognitive robotics is language-based communication which should be natural as well as grounded in the context in which the dialogue takes place. As pointed out by Wilcock and Jokinen (2023), among others, the main problem of ChatGPT-type interaction is that the models have no understanding of the real world: sentences are generated as strings of words, but they are not grounded in real world experience and they do not convey feelings or a genuine intention to communicate. A robot may assist humans to manipulate objects or navigate in the environment, so the meaning of the utterances must be linked to a true representation of relevant events, objects and actions. Also the lack of trustworthy information and tendency to hallucinate undermine the reliability of LLMs for applications especially in the health and eldercare domains, because of the model's outdated information and unknown data sources, as well as the "long-tail" problem, i.e., problems learning low-frequency facts (Kandpal et al., 2022). Recently also semantic inconsistency of ChatGPT has been studied (Jang and Lukasiewicz, 2023) with the conclusion that inconsistency issues undermine its reliability and cannot simply be resolved by prompt design and data augmentation.

The contributions of this paper deal with research areas of cognitive robotics and conversational AI. We study the linking of neural and symbolic processing from the point of view of conversational AI and support the view that grounding (actually more than one type of grounding) is needed in LLM-based dialogue systems which aim to be of value for human users by providing cognitively plausible dialogue behaviour. We draft a model that uses conversational AI and knowledge graphs for the purpose of building shared understanding of the dialogue situation, combining neural technologies for symbol-level interaction and creating common ground, and also discuss how grounding can be used to leverage both reliable information exchange and smooth interaction for robot dialogues.

The paper is structured as follows. Section 2 summarizes previous and related work. We discuss the grounding models in Section 3. The knowledge graph technologies used in our models are briefly presented in Section 4. We conclude with discussion on future directions in Section 5.

2. Previous and Related Work

We give an overview of our general framework of Constructive Dialogue Model in Subsection 2.1, and summarize related work in grounding in Subsection 2.2

2.1. Constructive Dialogue Model

Context-aware dialogue research (Jokinen, 2018) emphasizes that an intelligent agent needs to be aware of its context in order to support natural and attentive dialogues. An important characteristic of the agent is the ability to communicate in a manner which is well-timed concerning the partner's attention and appropriately formulated concerning the partner's intentions. Such behavior creates common ground to achieve goals, seek information, and create social bonds, i.e. dialogue partners construct conversation together in their conversational interaction.

Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ LREC-Coling 2024, pages 45–52
21 May, 2024.

In cognitive robotics (Cangelosi and Asada, 2022), robots should communicate with humans in a socially correct way, and their ability to recognize the user's spoken and multimodal utterances must be combined with their own speech, gesturing and multimodal behaviour. Consequently, human-robot interactions resemble interactive situations between two agents. However, our claim is not that the robot agent is conscious about its acts or that it understands the meaning of linguistic symbols in the same way as humans; rather, we put forward the view that human-robot interactions are perceived as natural and intentional, if the robot agent's operation and interaction are based on similar capabilities (affordances) as those used in human-human interactions.

The Constructive Dialogue Model (CDM) is a conceptual and operational framework which regards conversational interactions as cooperative activities through which the participants build common ground (for more information see (Jokinen, 1996, 2009)). The CDM architecture takes into account the multidimensional and intertwined nature of human-agent interaction from a dynamic systems theory perspective. Dynamic systems theory perceives human development as a connectionist process of self-organization and emergence: systems can generate novelty through their own activity, which consists of many decentralized and local interactions that occur in real time. In systemic approaches, communication is understood as the emergent product of multiple activities in the participants' cognitive neuroarchitectures, and it can be viewed as a constant but regulated change within a complex dynamic system, formed by the intertwined activities.

In CDM, participants aim to achieve their communicative goals by conveying information about their intentions and tasks. They are engaged in the exchange of new information which includes feedback about their understanding, attitude, emotions, and willingness to interact. Their individual acts create a new (cognitive) state and together the participants generate conversation as a joint action. The dynamic development of conversation enables the participants to construct mutual understanding (although not necessarily agreement about the tasks and intentions), whereas various enablements of communication constrain and regulate the interaction, such as the need to be in contact, to perceive various partner actions as communicative signals, to be able to understand the partner's message, and to be able to produce one's own reaction. Reaction encodes new information which changes the system state and causes the agents to organise their reasoning with respect to the new state.

One of the main challenges for CDM is how to update one's knowledge in order to align with the partner to construct shared context and react appropriately. The process of grounding is used to establish links between new and old information, and to determine optimal communicative action for the construction of shared knowledge. Grounding is manifested by the signals that indicate the agents' cooperation and their attention to the partner's needs: verbal acknowledgement and relevant continuation of the conversation is accompanied by non-verbal feedback. Several studies deal with multimodal feedback-giving processes, expressed by eye-gaze, facial expressions, head nods, hand gestures, body movement, and a wide range of vocalisations such as laughter etc. For instance, Mori et al. (2022) studied nods in human conversations, and proposed a model which includes a component for updating the partner's internal cognitive state (such as knowledge, understanding and emotional stance), on the basis of which the agent can decide on the appropriate feedback. The model focuses on the type of nod, but takes into account also a whole repertoire of possible feedback expression (verbal, gesturing, body posture). Models for such expressive interaction are important in many cognitive robotics applications, where task completion is not enough but more comprehensive and affective interaction is desired.

2.2. Related Work

Rather than focusing solely on task completion as the basis of the efficient communication, linguistic grounding research has focussed on naturalness of interactions and measuring user engagement through the participants' multimodal activity. Such approaches concern cooperative dialogue management (Clark and Wilkes-Gibbs, 1986; Clark and Brennan, 1991; Allwood et al., 1992; Traum and Allen, 1992; Traum and Heeman, 1996), and more recently (Kawano et al., 2021; Udagawa and Aizawa, 2021). Some recent research on linguistic grounding has also been conducted by Axelsson and Skantze (2023a,b) using the Furhat robot as an interface robot. In this work, general knowledge graph entities and links are marked by temporary labels, and the memory space has to be updated every time time a dialogue session starts.

Jokinen et al. (2024) explore how to predict grounding and shared knowledge in dialogues, given the listener feedback and a suitable prompt design with examples. In particular, they investigate if LLMs can be used to construct shared knowledge in an interactive context of an information seeker and an information provider conversing over a particular topic. The information provider's knowledge is structured in a table format, while the seeker queries the information until understood and satisfied with the result. Three types of conversational grounding are assumed: explicit (expressed by explicit feedback signals that show the partner's understanding), implicit (expressed by moving on in the dialogue to other topics without explicit linguistic signals to show the partner's understanding), and clarification (expressed by a clarification question to ask further information). The results are positive and demonstrate the LLMs capability to dynamically build structured knowledge, but further studies are needed to distinguish between the implicit and clarification types of grounding, to include multimodal feedback, and to fine-grain the analysis of situations where misunderstandings occur.

In a series of papers (Wilcock and Jokinen, 2022a,c; Jokinen and Wilcock, 2024), Jokinen and Wilcock have extensively studied cooperative and uncooperative robot behaviour, also using Furhat robots. They propose a solution with knowledge graphs for grounding to control LLMs in dialogue modelling and to alleviate the LLM's tendency to produce false information.

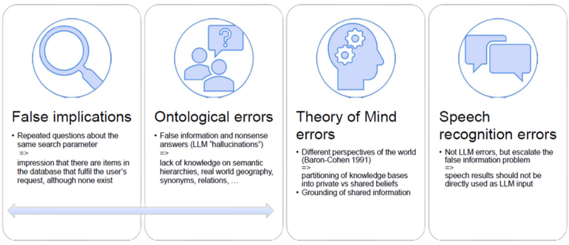

False implications
• Repeated questions about the same search parameter
=>
impression that there are items in the database that fulfil the user's request, although none exist
Ontological errors
• False information and nonsense answers (LLM "hallucinations")
=>
lack of knowledge on semantic hierarchies, real world geography, synonyms, relations, …
Theory of Mind errors
• Different perspectives of the world (Baron-Cohen 1991)
=>
partitioning of knowledge bases into private vs shared beliefs
• Grounding of shared information
Speech recognition errors
• Not LLM errors, but escalate the false information problem
=>
speech results should not be directly used as LLM input

Figure 1: Different errors in LLM-based dialogues.

Their recent work (Wilcock and Jokinen, 2023) compares LLM-based dialogue systems with knowledge graph-based systems from the point of view of errors that occur in testing. They distinguish false implications, ontological errors, and Theory of Mind errors, as shown in Figure 1. The figure also includes speech recognition errors but these are not discussed here as their solutions are not directly included in the knowledge-base reasoning.

False implications are errors where the user is led into making assumptions that are not true, while ontological errors result from a lack of knowledge of the semantics and structure of the world. They can be remedied by adding semantic metadata such as taxonomies and geographical locations to the knowledge graphs, and by using more flexible searches. Theory of Mind errors occur when participants have different perspectives of the situation, and are caused by lack of grounding.

3. Grounding Models

We distinguish between Theory of Mind grounding (in Subsection 3.1) and knowledge grounding (in Subsection 3.2).

3.1. Theory of Mind and Grounding

As mentioned, Wilcock and Jokinen (2023) point out that while the other interaction errors may be resolved by the RAG approach and its developments, Theory of Mind (ToM) errors occur when the participants have different knowledge of the situation and its solution requires modelling of the partner's mental state.

According to Theory of Mind (ToM) (Baron-Cohen, 1991), the development of human cognition requires the understanding of other minds having different content than one's own: another person's mind is related to their perspective of the world which is not necessarily the same as one's own. In cognitive robotics, ToM is used as a basis for the studies to construct a shared knowledge and mutual understanding of the context of the physical world, which are also the main issues in cooperative dialogue modelling.

LLM-based interactions lack shared understanding of the partner's worldview, experience, emotions, and environment. Figure 2 exemplifies a common situation in human-robot interactions. In the user's mind the referent of *the last one* is the recently listed item *To the Herbs*, whereas the system regards the phrase *the last one* to refer to *Pesche Doro*, the last one in its list of database items. The mismatch leads to confusion but can be resolved by a clarification question. The error is not seen as a mistake but as a lack of relevant knowledge, and its recovery thus becomes a matter of constructing appropriate shared knowledge. We call this *conversational grounding*, as it is based on the conversational context of what the partners have been talking about and how they interpret the language referents in the current context.

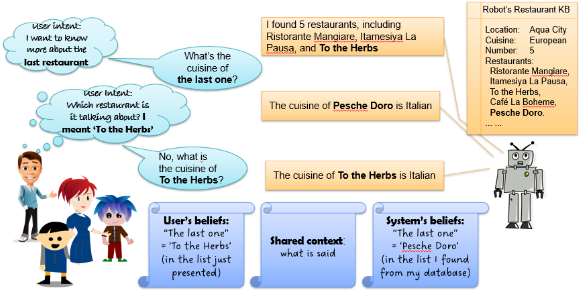

User intent: I want to know more about the last restaurant
What's the cuisine of the last one?
User intent: Which restaurant is it talking about? I meant 'To the Herbs'
No, what is the cuisine of To the Herbs?
I found 5 restaurants, including Ristorante Mangiare, Itamesiya La Pausa, and To the Herbs
The cuisine of Pesche Doro is Italian
The cuisine of To the Herbs is Italian
Robot's Restaurant KB
Location: Aqua City
Cuisine: European
Number: 5
Restaurants: Ristorante Mangiare, Itamesiya La Pausa, To the Herbs, Café La Boheme, Pesche Doro. … …
User's beliefs: "The last one" = 'To the Herbs' (in the list just presented)
Shared context: what is said
System's beliefs: "The last one" = 'Pesche Doro' (in the list I found from my database)

Figure 2: Conversational grounding (ToM error).

Another type of grounding is exemplified in Figure 3. In order to act in the real world and cooperate with humans, the robot agent must have knowledge of the environment and how language concepts are linked to the entities in the environment where the interaction takes place. For instance, in object manipulation and navigation tasks where the robot collaborates with humans, computer vision technology needs to be combined with LLMs to give the robot a sense of the environment and the skill to talk about it. We call this *visual grounding*, which has been long studied in robotics (cf. (Harnad, 1990)), where it refers to the grounding of utterances into the perceptions of the world. Simultaneous visual and conversational grounding allows the agent to assess the relevance and truth of the partner's utterances with respect to the current environment and to generate an appropriate response within the shared knowledge, e.g. asking a clarification question to recognize the correct referent mentioned by the user.

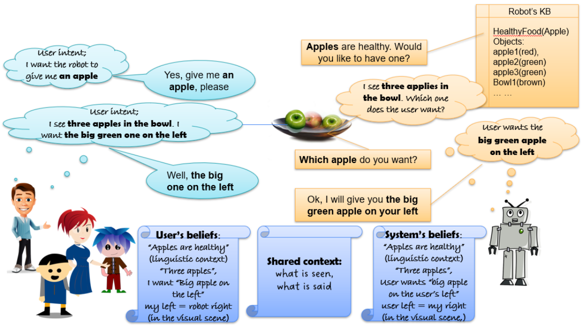

User intent;
I want the robot to give me **an apple**
Yes, give me **an apple**, please
User intent;
I see **three apples in the bowl**. I want **the big green one on the left**
Well, **the big one on the left**
Apples are healthy. Would you like to have one?
Robot's KB
HealthyFood(Apple)
Objects:
apple1(red),
apple2(green)
apple3(green)
Bowl1(brown)
… …
I see **three applies in the bowl**. Which one does the user want?
User wants the **big green apple on the left**
Which apple do you want?
Ok, I will give you **the big green apple on your left**
User's beliefs:
"Apples are healthy"
(linguistic context)
"Three apples",
I want "Big apple on the left"
my left = robot right
(in the visual scene)
Shared context:
what is seen,
what is said
System's beliefs:
"Apples are healthy"
(linguistic context)
"Three apples",
User wants "big apple on the user's left"
user left = my right
(in the visual scene,)

Figure 3: Visual grounding and the real world.

Perspective taking is one of the challenges in current computer vision research (Lemaignan et al., 2011), whereas recent advances in Visual Dialogue Modelling (Wu et al., 2017) combine speech and images in order to allow spoken natural language questions and answers deal with the elements that are recognized in the image.

To address ToM errors, the agent must distinguish private and shared knowledge, and have a goal to build shared knowledge in order to advance the task via communication. We make a distinction between existing static knowledge and dynamic dialogue processing knowledge, but represent both in a knowledge graph. Each user can have a personal knowledgebase which contains their personal information and preferences but can also be extended dynamically in the dialogue, including their view of the dialogue situation. In order to update one's own knowledge and align it with the partner's knowledge, the agent constructs a shared context as part of the knowledge representation. We aim to leverage the knowledgebase approach for updates and reasoning by deploying the typical procedures for searching and updating knowledge graph databases. For instance, communicative actions establish links between the nodes in the graph structure, and these can be dynamically updated as property updates of the entities and the links.

3.2. LLMs, KGs and Grounding

As discussed above, simple application of LLMs enables the robot agents to talk fluently on any topic, but the sentences are basically imitations of what could be said, rather than manifestations of the speaker's intention to convey some information to the partner (giving rise to the phrase "statistical parrot" (Bender et al., 2021)). In the knowledgebase approach, generated sentences are grounded in the knowledgebase, curated by humans to represent true facts of the world. Ontologies and semantic metadata are important tools in providing necessary information about how the world is structured (see Wilcock and Jokinen (2022b)) and we can also use different knowledgebases (document collections, knowledge graphs) which contains relevant information about the domain and dialogue, and can also be said to "represent" the world.

Currently much research is focused on combining Knowledge Graphs (KGs) and LLMs, and a survey of this work is provided by Pan et al. (2023). When KGs are curated by human experts, the data provenance is known and errors of outdated data can be resolved (cf. Wikipedia). For instance, Di Bratto et al. (2021) describe how graph databases can be used as a framework for a understanding the domain during dialogue. They use Internet Movie Database and Wikidata with a reference to personal and common ground concepts. Wilcock and Jokinen (2023) discuss how KGs can be used to provide trustworthy information to the user and how KGs can be augmented with WikiData metadata. Fu et al. (2023) present how KG reasoning and ontologies enable more cooperative responses based on reliable data, and Schneider et al. (2023) describe how to use knowledge graphs and conversational interfaces for exploratory search, bridging the gap between structured and unstructured information retrieval on news articles.

As fluent conversational capability is one of the main advantages of LLMs, current research efforts aim to combine such capability with trustworthy reliable information. The third meaning for "grounding" can hence be found in the LLM and Knowledge Graph literature: it is discussed in the context of knowledgebases providing a reliable starting point for the LLM generation. We call this *knowledge grounding* as it refers to the grounding of linguistic information to the speaker's knowledge and experience of world, stored in knowledgebases and represented in texts, KGs, and cognitive models of the agent's knowledge.

4. Grounding and Knowledge Graph Technologies

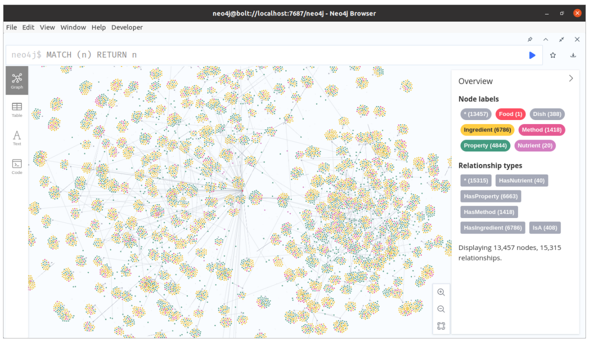
In this section we briefly explore how knowledge grounding can be included in dialogue management, using LLMs, retrieval augmentation, and knowledge graphs.

The RAG (Retrieval Augmented Generation) approach (Lewis et al., 2020) is commonly used as a generation model for reliable knowledge inclusion. It provides a solution to problems with false implications and ontological errors. The processing pipeline is divided into language understanding and response generation. First the user input is analysed to extract important concepts and the user intent. The analysis is then used for making a search query to retrieve relevant information from the knowledgebase. Response generation uses the information retrieved from the knowledgebase together with the user query and dialogue history, as input for the LLM-based generation module which then generates a response.

We use Neo4j graph databases (Robinson et al., 2015) in our work. Most recently LLMs have been used to generate Cypher queries that can search knowledge graphs in Neo4j (Bratanic, 2023). Symbolic representation of knowledge can thus be used as a grounding model for LLMs. Neo4j also includes a vector search capability which supports efficient semantic search of KGs by adding an embedding vector to each node. It can be used with LLMs to make semantic searches based on user queries in natural language that do not require exact lexical matches with node labels. It is interesting that the description of this capability refers to "grounding LLM responses", which in this paper is regarded as an example of knowledge grounding, i.e. representing a way how generative models ground their responses into curated knowledge.

This approach has been demonstated in Wilcock and Jokinen (2023) where knowledge graphs are used with robot dialogues in the CityTalk application to talk about restaurants and hotels (Wilcock, 2019). A similar approach is used in Jokinen and Wilcock (2024) but the interaction deals with the Kyoto cooking database (Kiyomaru et al., 2018) which has been converted into a knowledge graph. The graph is stored in a Neo4j graph database, as shown in Figure 4.

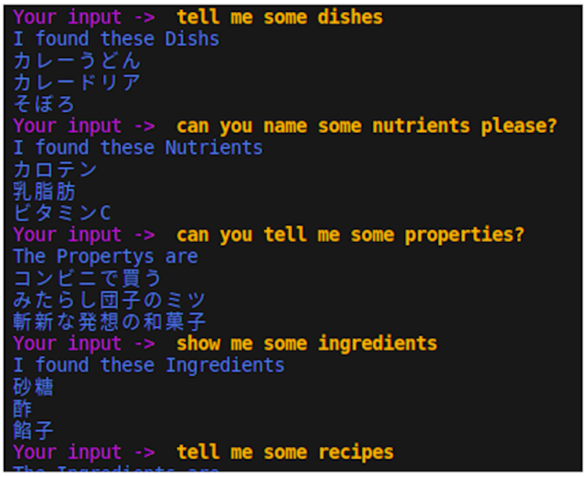
All the nodes in the Kyoto Cooking database are labelled with Japanese names. It is thus possible to have multilingual interaction as the graph can be queried in English or Japanese. An example of these mixed-language queries in Figure 5 is from an earlier version of the system where the names of dishes, ingredients, nutrients, and cooking methods in the responses are in Japanese, and the number of responses is limited to 3.



```
neo4j$ MATCH (n) RETURN n
```

Displaying 13,457 nodes, 15,315 relationships.

Figure 4: Kyoto Cooking knowledge graph in Neo4j.



```
Your input ->  tell me some dishes
I found these Dishs
カレーうどん
カレードリア
そぼろ
Your input ->  can you name some nutrients please?
I found these Nutrients
カロテン
乳脂肪
ビタミンC
Your input ->  can you tell me some properties?
The Propertys are
コンビニで買う
みたらし団子のミツ
斬新な発想の和菓子
Your input ->  show me some ingredients
I found these Ingredients
砂糖
酢
餡子
Your input ->  tell me some recipes
The Ingredients are
```

Figure 5: Currently mixed-language responses.

5. Conclusion and Future Work

The paper describes ongoing research on human-robot dialogues where knowledge graphs are used to make the interaction more natural and trustworthy. The paper supports the view that human interaction with robots is quite unlike interactions with text-based systems or with other types of mobile devices, and that conversational robot agents should enable a grounding process in order to create shared context with the human partner, so as to advance technological readiness of cognitive robot applications.

The shared context is constructed through grounding dynamically in the conversation, and it is represented by knowledge graphs. Structured knowledge modelling concerns relevant information of the application domain and of the world, and ultimately of the speaker's own view-point of the real-world events and entities. The paper aims to show the dynamic nature of grounding and the complexity of the construction of shared knowledge between the dialogue partners.

Three different types of grounding are distinguished: 1) conversational grounding establishes links from language expressions to the shared dialogue context (i.e. beliefs of what knowledge is shared in the context), 2) visual grounding supports grounding of language expressions to suitable elements in the context taking into account the whole visual scene, and 3) knowledge grounding anchors language expressions into the agent's own knowledge (long-term memory in which the agent's knowledge and experience is stored). Each type has an important role in the communication and in the processing of the partner's communicative signals. They also demonstrate how the symbolic representations can be grounded within the same framework of structured knowledge graphs as vectorized documents and LLMs, thus linking symbolic representations of thoughts and intentions to cognitive processing of neural representations. The grounding models also show how the dynamic communication system can be controlled by communicative enablements, and how the problematic issues of false and irrelevant information can be alleviated to harness the conversational power of LLMs for language-capable robots.

Future work concerns user studies to evaluate appropriateness and success of the dialogues, as well as application of the approach to knowledge bases of various sizes and domains. In grounding research, multimodal aspects of dialogue need to be taken into account, as well as better understanding of the grounding process and its cogntive modelling. Main challenges deal with the construction of structured knowledge bases, their maintenance and updating, sustainability of LLMs, and various ethical aspects (Williams et al., 2023) related to language capable agents.

6. Acknowledgements

The author acknowledges the support of Project JPNP20006 commissioned by the New Energy and Industrial Technology Development Organization (NEDO), Japan

7. Bibliographical References

Jens Allwood, Joakim Nivre, and Elizabeth Ahlsén. 1992. On the Semantics and Pragmatics of Linguistic Feedback. *Journal of Semantics*.

Agnes Axelsson and Gabriel Skantze. 2023a. Do you follow? a fully automated system for adaptive robot presenters. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '23, page 102–111, New York, NY, USA. Association for Computing Machinery.

Agnes Axelsson and Gabriel Skantze. 2023b. Using large language models for zero-shot natural language generation from knowledge graphs. ArXiv:2307.07312.

Simon Baron-Cohen. 1991. Precursors to a theory of mind: Understanding attention in others. In A. Whiten, editor, *Natural theories of mind: Evolution, development and simulation of everyday mindreading*, pages 233–251. Basil Blackwell.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, page 610–623. Association for Computing Machinery.

Tomaz Bratanic. 2023. Generating Cypher queries with ChatGPT 4 on any graph schema. https://neo4j.com/developer-blog/generating-cypher-queries-with-chatgpt-4-on-any-graph-schema/.

Angelo Cangelosi and Minoru Asada. 2022. *Cognitive Robotics*. The MIT Press.

Herbert H. Clark and S. A. Brennan. 1991. Grounding in communication. In L.B. Resnick, J.M. Levine, and S.D. Teasley, editors, *Perspectives on socially shared cognition*. APA Books.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Martina Di Bratto, Maria Di Maro, Antonio Origlia, and Francesco Cutugno. 2021. Dialogue analysis with graph databases: Characterising domain items usage for movie recommendations. In *Proceedings of the Eighth Italian Conference on Computational Linguistics CliC-it*, Milan, Italy.

Yahui Fu, Koji Inoue, Chenhui Chu, and Tatsuya Kawahara. 2023. Reasoning before responding: Integrating commonsense-based causality explanation for empathetic response generation. In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL 2023, Prague, Czechia, September 11 - 15, 2023*, pages 645–656. Association for Computational Linguistics.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346.

Myeongjun Erik Jang and Thomas Lukasiewicz. 2023. Consistency analysis of ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing,* *EMNLP 2023, Singapore, December 6-10, 2023*. Association for Computational Linguistics.

Kristiina Jokinen. 1996. Cooperative Response Planning in CDM: Reasoning about Communicative Strategies. In *Twente Workshop Series in Language Technology*.

Kristiina Jokinen. 2009. *Constructive Dialogue Modelling: Speech Interaction and Rational Agents*. John Wiley & Sons.

Kristiina Jokinen. 2018. Dialogue models for socially intelligent robots. In *10th International Conference, ICSR 2018, Qingdao, China, November 28 - 30, 2018, Proceedings*, pages 127–138.

Kristiina Jokinen, Phillip Schneider, and Taiga Mori. 2024. Towards Harnessing Large Language Models for Comprehension of Conversational Grounding. In *14th International Workshop on Spoken Dialogue Systems Technology (IWSDS 2024)*, Sapporo, Japan.

Kristiina Jokinen and Graham Wilcock. 2024. Exploring a Japanese cooking database. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI 2024)*, pages 578–582, Boulder, Colorado, USA. Association for Computing Machinery.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2022. Large language models struggle to learn long-tail knowledge. ArXiv:2211.08411.

Seiya Kawano, Koichiro Yoshino, David Traum, and Satoshi Nakamura. 2021. Dialogue structure parsing on multi-floor dialogue based on multi-task learning. Presented at Robotdial Workshop.

Kanichi Kiyomaru, Sadao Kurohashi, Mitsuru Endo, and Katsuyoshi Yamagami. 2018. Building a basic cooking knowledge base based on cooking recipes and crowdsourcing (in Japanese). In *24th Annual Conference of the Natural Language Processing Society*, Japan.

Séverin Lemaignan, Raquel Ros, Rachid Alami, and Michael Beetz. 2011. What are you talking about? grounding dialogue in a perspective-aware robotic architecture. In *2011 RO-MAN*, pages 107–112.

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. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, pages 9459–9474, Vancouver, Canada.

Taiga Mori, Kristiina Jokinen, and Yasuharu Den. 2022. Cognitive States and Types of Nods. In *Proceedings of the International LREC Workshop on People in Vision, Language and the Mind (P-VLAM)*, pages 17–25, Marseille, France. European Language Resources Association (ELRA).

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying large language models and knowledge graphs: A roadmap.

Ian Robinson, Jim Webber, and Emil Eifrem. 2015. *Graph Databases (2nd edition)*. O'Reilly Media.

Phillip Schneider, Nils Rehtanz, Kristiina Jokinen, and Florian Matthes. 2023. From data to dialogue: Leveraging the structure of knowledge graphs for conversational exploratory search. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation (PACLIC 2023)*, Hong Kong, China.

David R. Traum and James F. Allen. 1992. A Speech Acts Approach to Grounding in Conversation. In *Proceedings of 2nd International Conference on Spoken Language Processing (ICSLP-92)*, pages 137–140.

David R. Traum and Peter Heeman. 1996. Utterance Units and Grounding in Spoken Dialogue. In *Proceedings of International Conference on Spoken Language Processing (ICSLP-96)*.

Takuma Udagawa and Akiko Aizawa. 2021. Maintaining Common Ground in Dynamic Environments. *Transactions of the Association for Computational Linguistics*, 9:995–1011.

Graham Wilcock. 2019. CityTalk: Robots that talk to tourists and can switch domains during the dialogue. In *9th International Workshop on Spoken Dialogue Systems Technology*, pages 411–417. Springer.

Graham Wilcock and Kristiina Jokinen. 2022a. Conversational AI and knowledge graphs for social robot interaction. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI 2022)*, pages 1090–1094, Sapporo, Japan. Association for Computing Machinery.

Graham Wilcock and Kristiina Jokinen. 2022b. Cooperative and uncooperative behaviour in task-oriented dialogues with social robots. In *31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN 2022)*, pages 763–768, Napoli, Italy.

Graham Wilcock and Kristiina Jokinen. 2022c. Should robots indicate the trustworthiness of information from knowledge graphs? In *10th International Conference on Affective Computing and*

Intelligent Interaction (ACII 2022) Workshops and Demos, Nara, Japan. IEEE Computer Society.

Graham Wilcock and Kristiina Jokinen. 2023. To Err Is Robotic; to Earn Trust, Divine: Comparing ChatGPT and Knowledge Graphs for HRI. In *32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN 2023)*, pages 1396–1401, Busan, Korea.

Tom Williams, Cynthia Matuszek, Kristiina Jokinen, Raj Korpan, James Pustejovsky, and Brian Scassellati. 2023. Voice in the Machine: Ethical Considerations for Language-Capable Robots. *Communications of the ACM*, 66(8):20–23.

Qi Wu, Peng Wang, Chunhua Shen, Ian Reid, and Anton van den Hengel. 2017. Are you talking to me? reasoned visual dialog generation through adversarial learning. ArXiv:1711.07613.

Author Index