



Politecnico di Bari

Repository Istituzionale dei Prodotti della Ricerca del Politecnico di Bari

A journey through the hidden representations of pretrained language models: semantics, factuality and beyond

This is a PhD Thesis

Original Citation:

A journey through the hidden representations of pretrained language models: semantics, factuality and beyond / De Bellis, Alessandro. - ELETTRONICO. - (2026).

Availability:

This version is available at <http://hdl.handle.net/11589/295622> since: 2026-01-15

Published version

DOI:

Publisher: Politecnico di Bari

Terms of use:

(Article begins on next page)

22 April 2026



Politecnico
di Bari



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

Alessandro De Bellis

A Journey through the Hidden Representations of Pretrained Language Models: Semantics, Factuality and Beyond

Thesis submitted for the degree of Philosophiae Doctor

Interuniversity Ph.D. Program in Industry 4.0

Politecnico di Bari

Tutors

Prof. Engr. *Tommaso Di Noia*

Prof. Engr. *Eugenio Di Sciascio*

2025

This page intentionally left blank



Dissertation submitted for the degree of *Philosophiae Doctor*
Interuniversity Ph.D. Program in Industry 4.0

Cycle:

38th (2022, November 1 – 2025, October 31)

Administrative Headquarters:

Politecnico di Bari – Department of Electrical and Information Engineering

Hosting University:

Politecnico di Bari – Department of Electrical and Information Engineering

Title:

A Journey through the Hidden Representations of Pretrained Language Models:
Semantics, Factuality and Beyond

Ph.D Candidate:

Alessandro De Bellis

Academic Discipline:

Information Processing Systems (Italian code: IINF-05/A)

Tutors:

Prof. Engr. Tommaso Di Noia, Politecnico di Bari (Bari, Italy)

Prof. Engr. Eugenio Di Sciascio, Politecnico di Bari (Bari, Italy)

Coordinator:

Prof. Engr. Salvatore Digiesi, Politecnico di Bari (Bari, Italy) *Salvatore Digiesi*

External Reviewers:

Prof. Pierpaolo Basile, University of Bari (Bari, Italy)

Prof. Fabrizio Silvestri, Sapienza University of Rome (Rome, Italy)

Last version:

December 30, 2025

All rights reserved. No part of this publication may be reproduced or transmitted, in any form or by any means, without permission.

This page intentionally left blank



LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il sottoscritto Alessandro De Bellis nato a Castellaneta il 11/07/1995

residente a Acquaviva delle Fonti in via Umberto Giordano, 10 e-mail aledeb95@gmail.com

iscritto al 3° anno di Corso di Dottorato di Ricerca in Industria 4.0 ciclo 38°

ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

A JOURNEY THROUGH THE HIDDEN REPRESENTATIONS OF PRETRAINED LANGUAGE MODELS:
SEMANTICS, FACTUALITY AND BEYOND

DICHIARA

- 1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
- 2) di essere iscritto al Corso di Dottorato di ricerca in Industria 4.0 ciclo 38°, corso attivato ai sensi del "Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari", emanato con D.R. n.286 del 01.07.2013;
- 3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
- 4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archiverà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito <http://www.creativecommons.it/Licenze>), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
- 5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviata/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
- 6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
- 7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Bari, 14/01/2026

Firma _____

Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Bari, 14/01/2026

Firma _____

This page intentionally left blank

Abstract

The rapid emergence of pretrained language models (PLMs) has fundamentally transformed Natural Language Processing (NLP). Despite their success, the standard fine-tuning paradigm introduces significant limitations, including knowledge destabilization, high computational overhead, reproducibility challenges, and limited transparency. These constraints impede both research progress and the scalable deployment of PLMs in industrial contexts.

This dissertation investigates an alternative approach that leverages the frozen internal representations of PLMs without modifying their parameters. By treating PLMs as fixed computational artifacts, the study examines how their latent representational geometry encodes structured, task-relevant information that can be systematically extracted and utilized.

Empirical analyses demonstrate that frozen PLMs inherently encode diverse forms of knowledge. This includes hierarchical semantic relationships useful for fine-grained entity typing and ontology completion, which in turn enhance graph-based relational inference. Their representations also contain signals related to factuality, enabling potential self-assessment of generated content. Beyond semantic information, frozen PLMs are shown to capture affective and evaluative cues relevant to downstream tasks, such as sentiment analysis, and to exhibit zero-shot generalization in recommendation and search tasks.

Overall, the findings illustrate that a broad spectrum of latent capabilities can be accessed from PLMs without weight modification. These contributions advance interpretability by linking internal representations to human-interpretable semantics, factuality, and affective dimensions, and promote sustainability by reducing dependence on repeated, resource-intensive fine-tuning. The work supports a paradigm in which the capabilities of PLMs are harnessed transparently, efficiently, and responsibly for future NLP and AI applications.

This page intentionally left blank

Publications

This chapter lists publications arising from the doctoral work, presented chronologically and formatted as in the bibliography; **boldface** identifies the corresponding author(s).

1. **Alessandro De Bellis** (2023). “Structuring the Unstructured: an LLM-guided Transition”. In: *Proceedings of the Doctoral Consortium at ISWC 2023 co-located with 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 7, 2023*. Ed. by Claudia d’Amato, and Jeff Z. Pan. Vol. 3678. CEUR Workshop Proceedings. CEUR-WS.org. URL: <https://ceur-ws.org/Vol-3678/paper12.pdf>
2. **Alessandro De Bellis, Giovanni Maria Biancofiore, Vito Walter Anelli**, Fedelucio Narducci, Tommaso Di Noia, Azzurra Ragone, and Eugenio Di Sciascio (2023). “Semantic Interpretation of BERT Embeddings with Knowledge Graphs”. In: *Proceedings of the 31st Symposium of Advanced Database Systems, Galzingano Terme, Italy, July 2nd to 5th, 2023*. Ed. by Diego Calvanese, Claudia Diamantini, Guglielmo Faggioli, Nicola Ferro, Stefano Marchesin, Gianmaria Silvello, and Letizia Tanca. Vol. 3478. CEUR Workshop Proceedings. CEUR-WS.org, pp. 181–191. URL: <https://ceur-ws.org/Vol-3478/paper69.pdf>
3. **Alessandro De Bellis, Vito Walter Anelli**, Tommaso Di Noia, and Eugenio Di Sciascio (2024). “PRONTO: Prompt-Based Detection of Semantic Containment Patterns in MLMs”. In: *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part II*. ed. by Gianluca Demartini, Katja Hose, Maribel Acosta, Matteo Palmonari, Gong Cheng, Hala Skaf-Molli, Nicolas Ferranti, Daniel Hernández, and Aidan Hogan. Vol. 15232. Lecture Notes in

- Computer Science. Springer, pp. 227–246. DOI: [10.1007/978-3-031-77850-6_13](https://doi.org/10.1007/978-3-031-77850-6_13). URL: https://doi.org/10.1007/978-3-031-77850-6_13
4. **Giovanni Servedio, Alessandro De Bellis**, Dario Di Palma, Vito Walter Anelli, and Tommaso Di Noia (2025). “Are the Hidden States Hiding Something? Testing the Limits of Factuality-Encoding Capabilities in LLMs”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics, pp. 6089–6104. URL: <https://aclanthology.org/2025.acl-long.304/>
 5. **Dario Di Palma**, Alessandro De Bellis, Giovanni Servedio, Vito Walter Anelli, Fedelucio Narducci, and Tommaso Di Noia (2025b). “LLaMAs Have Feelings Too: Unveiling Sentiment and Emotion Representations in LLaMA Models Through Probing”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics, pp. 6124–6142. URL: <https://aclanthology.org/2025.acl-long.306/>
 6. **Matteo Attimonelli, Alessandro De Bellis, Claudio Pomo**, Dietmar Jannach, Eugenio Di Sciascio, and Tommaso Di Noia (2025). “Do We Really Need Specialization? Evaluating Generalist Text Embeddings for Zero-Shot Recommendation and Search”. In: *Proceedings of the Nineteenth ACM Conference on Recommender Systems. RecSys ’25*. Association for Computing Machinery, pp. 575–580. ISBN: 9798400713644. DOI: [10.1145/3705328.3748040](https://doi.org/10.1145/3705328.3748040). URL: <https://doi.org/10.1145/3705328.3748040>
 7. **Alessandro De Bellis**, Salvatore Bui, Giovanni Servedio, Vito Walter Anelli, Tommaso Di Noia, and Eugenio Di Sciascio (Nov. 2025). “Type-Less yet Type-Aware Inductive Link Prediction with Pretrained Language Models”. In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Ed. by Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng. Suzhou, China: Association for Computational Linguistics, pp. 27181–27197. ISBN: 979-8-89176-332-6. URL: <https://aclanthology.org/2025.emnlp-main.1383/>

Contents

Abstract	i
Publications	iii
Abbreviations	xix
1 Introduction	1
1.1 Thesis Goal and Contributions	4
1.2 Thesis Outline	5
2 Background	7
2.1 Language Models	7
2.1.1 The Rise of the Transformer Architecture	9
2.1.2 Pre-training and Fine-tuning Paradigm	12
2.2 Parametric Knowledge in PLMs	13
2.2.1 Language Models as Knowledge Bases?	14
2.2.2 Hallucinations and Open Challenges	15
2.3 Interpretability of PLMs Hidden Representations	15
2.3.1 Probing of Representations	16
2.3.2 Recent Advances and Causal Probing	17
2.4 Language Models and Knowledge Graphs	18
2.4.1 Knowledge Graphs	18
2.4.2 Knowledge Graphs for PLMs	19
2.4.3 PLMs for Knowledge Graphs	20
3 Explicit Semantics in PLMs	23
3.1 Semantic Containment in MLMs	28
3.1.1 Methodology	29

3.1.2	Experiments	35
3.1.3	Limitations	45
3.1.4	Summary	45
3.2	Type-Less yet Type-Aware Inductive Link Prediction	47
3.2.1	Background and Motivation	48
3.2.2	Methodology	50
3.2.3	Experimental Setup	55
3.2.4	Results	59
3.2.5	Limitations	64
3.2.6	Summary	65
3.3	Related Work	66
3.3.1	Ontological and Type Knowledge in PLMs	67
3.3.2	Entity Typing with PLMs	68
3.3.3	Verbalizers and Prompt Learning	68
3.3.4	PLMs for Inductive Link Prediction	68
3.4	Overview: Choice of Representation	70
4	Factuality in LLMs	73
4.1	The Case for Self-Verification	76
4.2	Limits of LLMs Factuality-Encoding Capabilities	79
4.2.1	Reproduction of Prior Work: Settings	80
4.2.2	Experimental Reproducibility Results	82
4.2.3	Novel Dataset Generation Strategies for Factuality Self-Evaluation	85
4.2.4	Experiments and Discussion	89
4.2.5	Limitations	97
4.2.6	Summary	99
4.3	Related Work	100
4.4	Overview: Choice of Representation	101
5	Moving Beyond: Hidden Capabilities in PLMs	103
5.1	Sentiment and Emotion Representations in LLaMA	106
5.1.1	Methodology	108
5.1.2	Experimental Results	113
5.1.3	SENTRI LLAMA for Efficient Downstream Tasks	117
5.1.4	Discussion	121

5.1.5	Limitations	123
5.1.6	Summary	124
5.2	GTEs for Zero-Shot Recommendation and Search	125
5.2.1	Background and Methodology	127
5.2.2	Experiments and Discussion	128
5.2.3	Summary	134
5.3	Related Work	135
5.3.1	Sentiment Information in PLMs	135
5.3.2	Language Models as Text Embedders for Recommendation and Search	137
5.4	Overview: Choice of Representation	139
6	Conclusion	143
6.1	Summary of Contributions	143
6.2	Current Limitations and Future Challenges	144
6.2.1	On Explicit Semantics and Structured Knowledge	145
6.2.2	On Factuality and Internal Self-Evaluation	146
6.2.3	On Latent Capabilities and Sustainable Applications	147
6.3	Final Perspective	149

This page intentionally left blank

List of Figures

3.1	PRONTO Schematization: given a pair representing a containment in a reference taxonomy, representing a semantic containment relationship ("is-a"), PRONTO predicts the plausibility of the pair with a learned verbalizer.	30
3.2	Comparison of AUC scores for different verbalization strategies on the DBPedia Eval (hard) dataset. The reference model is RoBERTa-Large.	39
3.3	Visualization of learned verbalizers: the wordclouds report the top weighted tokens for the "is-a" relationship in hard templates. The reference model is RoBERTa-Large under the PRONTO-WS setting.	41
3.4	Overview of TyleR. The process begins with ① extracting the enclosing subgraph and ② applying a node labeling strategy. Multifaceted, semantic representations are then derived using a pre-trained language model ③a, ③b. Finally, a graph neural network ④ integrates structural and semantic information to obtain the final prediction.	50
3.5	Link Prediction (Hits@10) evaluation under varying structural sparsity conditions (i.e., the number of edges L in the enclosing subgraph of the target triple, including the target triple) on FB237-V1 (top) and YAGO21K-610 (bottom).	63
3.6	Hits@10 performance across four type sparsity groups for three FB237 variants, computed according to the number of explicit types linked to each entity (details in Section 3.2.4). The groups, from left to right, represent scenarios with an increasing number of explicit types associated with the known entity.	64

3.7	Overview of the representations extraction process adopted in Section 3.1. The last-layer hidden state ① is used for the verbalizer-free approach, while the pre-softmax logits ② are used as initial input for all the other verbalizer-based classifiers. More details are in Section 3.1.1.	71
3.8	Overview of the representation extraction process adopted in Section 3.2. The language model is assumed to be pretrained with a mask-filling objective (e.g., BERT (Devlin et al., 2019)), and representations are extracted from the final transformer layer at the [MASK] token position. For causal language models (LLaMA-3-8B (Dubey et al., 2024b) in our experiments), the input is reformulated without the mask token, which is replaced by a delimiter (":"); in this case, the extracted representation corresponds to the final-layer hidden state of the last input token.	72
4.1	Comparison of average perplexity scores for Llama 2-7b and OPT-6.7b for the Original dataset by Azaria and Mitchell (2023) and our refined version. Lower perplexity indicates that the sentences are more likely to have been generated by the model. PP(T/F) denotes average perplexity of true/false sentences.	91
4.2	Text generation pipeline for TriviaQA (Joshi et al., 2017). The 10 examples are prepended to the actual question, after which the model is prompted to generate an answer. The answers for the examples are manually crafted based on the available ground truth.	92
4.3	Example prompt provided to GPT-4o mini. This prompt implements a 3-shot learning strategy, supplying three examples that establish the expected judgment pattern for assessing answer correctness against a given ground truth. Following these examples, the model is instructed to evaluate a new question–answer pair using the same criteria.	93
4.4	Overview of the representations extraction process adopted in Section 4.2. Hidden states corresponding to the final input token are extracted from multiple transformer layers of a frozen LLM and used as input features for factuality probing.	102
5.1	Mean Pooling Visually Explained	110

5.2	Last Token Pooling Visually Explained	110
5.3	Max and Min Pooling Visually Explained	110
5.4	Attention Pooling Visually Explained	110
5.5	Visualization of the different pooling strategies used in our experiments.	110
5.6	Layer-wise probing accuracy using the Last-Token approach on the SST-2 dataset, for each of the twelve probing classifiers reported in Table 5.1. Figures show the best performing combination of layer and probing classifier.	114
5.7	Example of a zero-shot prompting template used for binary sentiment classification. The system is strictly instructed to return only numeric sentiment labels.	118
5.8	Example of a zero-shot prompting template used for emotion classification. The system is strictly instructed to return only numeric sentiment labels.	119
5.9	Example of a few-shot prompting template used for binary sentiment classification. The system is strictly instructed to return only numeric sentiment labels.	119
5.10	Example of a few-shot prompting template used for emotion classification. The system is strictly instructed to return only numeric sentiment labels.	120
5.11	Example of a chain-of-thought prompting template used for binary sentiment classification. The system is strictly instructed to return only numeric sentiment labels.	120
5.12	Example of a chain-of-thought prompting template used for emotion classification. The system is strictly instructed to return only numeric sentiment labels.	121
5.13	nDCG scores for various models on the two PS datasets (a) ESCI and (b) Amazon-C4 versus the number of components (log-scale). Points shown correspond to component counts yielding 80%, 95%, and 100% explained variance.	133

5.14	Overview of the representations extraction process adopted in Section 5.1. In this work, we perform a comprehensive, layer- and token-wise analysis of how sentiment information is encoded at the intermediate processing layers of a Llama model. Therefore, all intermediate, token-wise hidden states are taken into consideration. Aside from typical last-token pooling, we experiment with various pooling strategies, detailed in Section 5.1.1).	140
5.15	Overview of the representation extraction process adopted in Section 5.2. Although the figure only depicts the process for queries, queries and items are independently encoded using the same frozen Generalist Text Embedding (GTE) model, producing fixed-size dense vectors. Representations are obtained by aggregating final-layer token hidden states via a pooling mechanism, whose specific form depends on the underlying GTE architecture (e.g., NV-Embed-v2 employs a trainable latent attention pooling layer).	141

List of Tables

3.1	Visualization of the different hard/soft prompts used in this study. [s#] denotes a soft token.	34
3.2	Summary of the statistics for the four constructed DBPedia dataset splits.	35
3.3	Examples of negative pair construction. The examples are generated from the DBPedia dataset following the strategy described in Section 3.1.1.	36
3.4	Results over all combinations of Verbalizer-Template and different PLMs on the DBPedia evaluation (hard) dataset. All negative samples in this evaluation split are hard negatives, constructed as detailed in section 3.1.1. "Acc", "P", "R" and "F1" denote, respectively, accuracy, precision, recall and F1-score. In bold , we report the best results for each column.	38
3.5	GPT-3.5 turbo results (zero-shot) on the Eval (hard) dataset. . .	38
3.6	Performance comparison with the reverse negatives-only dataset. All results are in terms of AUC score. The reference model is RoBERTa-Large.	40
3.7	Results over all combinations of Verbalizer-Template on the DBPedia Eval (hard) dataset, in the "with context" setting. The reference model is RoBERTa-Large. "Acc", "P", "R" and "F1" denote, respectively, accuracy, precision, recall and F1-score. ↓ and ↑ denote, respectively, a decrease or an improvement in F1-score with respect to the results in absence of context (Tab. 3.4). In bold , we report the improved F1-scores.	43

3.8	Results over all combinations of Verbalizer-Template on the DB-Pedia Eval (hard) dataset, when 80% of instances present in the evaluation set have zero occurrences in the training set . The reference model is RoBERTa-Large. "Acc", "P", "R" and "F1" denote, respectively, accuracy, precision, recall and F1-score. In bold , we report the best results over all verbalizers.	44
3.9	Entity Typing results (zero-shot) on Few-NERD. The reference model is RoBERTa-Large.	45
3.10	Statistics of the datasets used in our experiments. The YAGO21K-610 (Zhou et al., 2023) dataset includes ontology triples and entity-type links, while the FB237 dataset variants (Teru et al., 2020) are further processed to extract ontology triples, type links and textual labels.	56
3.11	Assertion prompts (p_1 - p_6) used in the semantic enrichment step (Section 3.2.2). These templates, with a placeholder for the entity, are fed to the Pre-trained Language Model to elicit representations capturing different semantic aspects (type, geographic context, membership, equivalence, difference, similarity) of the entity. . . .	58
3.12	Link Prediction (LP) evaluation on multiple FB237 variants and YAGO21K-610. Best and second-best scores are in bold and <u>underlined</u> , respectively. Evaluation uses the strictest tie-breaking policy (Section 3.2.3), assigning the highest (worst) possible rank to the positive triple in case of ties.	58
3.13	Ablation study on the FB237-V1 dataset, evaluating the impact of different Pre-trained Language Models and aggregation functions (Equation 3.9) for semantic embeddings within TyleR. 'TYPE-ONLY' uses only the representation from the p_1 prompt (Table 3.11). . . .	60
3.14	Performance of Llama3-8B when directly evaluating the likelihood of verbalized triples on the YAGO21K-610 dataset by scoring them using the (negated) model perplexity.	60
3.15	Ablation study on the FB237-V1 dataset, evaluating the impact of chosen number of prompts (Equation 3.9) for semantic embeddings within TyleR. For all configurations, the aggregation function was set to SUM.	61

4.1	True-False dataset categories, original sources, and label splits from (Azaria and Mitchell, 2023).	81
4.2	Replicated SAPLMA performance on the True-False dataset across the selected layers. The results labeled as ‘Orig.’ are taken from the original work, while those labeled as ‘Repr.’ are the replicated results from this study.	82
4.3	Replicated baselines performance on the True-False dataset. The results labeled as ‘Orig.’ are taken from the original work, while those labeled as ‘Repr.’ are the replicated results from this study.	83
4.4	Reproduced SAPLMA performance on the OPT-Generated Dataset (Section 4.2.1). The results labeled as ‘Orig.’ are taken from the original work, while those labeled as ‘Repr.’ are the reproduced results.	83
4.5	Examples of false sentences generated with the Llama 2-7b model, using the perplexity-based generation strategy described in Section 4.2.3.	90
4.6	Novel generation of our True-False dataset, following the approach described in Section 4.2.3, including number of sentences and percentage of true samples.	90
4.7	Summary of the dataset obtained by extracting factoid sentences from TruthfulQA (Lin et al., 2022), following the procedure described in Section 4.2.3.	94
4.8	Summary of the dataset obtained by extracting factoid sentences from SQuAD 2.0 (Rajpurkar et al., 2018), following the procedure described in Section 4.2.3.	94
4.9	Summary of the dataset obtained by extracting factoid sentences from TriviaQA (Joshi et al., 2017) Wikipedia validation split, following the procedure described in Section 4.2.3.	95
4.10	Examples of true and false facts generated by Llama 2-7b and OPT-6.7b based on the questions in TriviaQA (Joshi et al., 2017), obtained following the procedure detailed in Section 4.2.3.	96

4.11	Accuracy values obtained training SAPLMA on the original True-False dataset and on our refined version, then tested on our refined version . Results are shown for the Llama2 and OPT models. ‘Orig.’ denotes the ‘original True-False dataset as training data, while ‘Novel’ denotes our version of the True-False dataset as training data. In bold we denote the best combination of layer/training dataset for each combination of model/topic. . . .	97
4.12	Performance of SAPLMA on a fact dataset generated from TriviaQA. The original topic-wise leave-one-out strategy is adopted. Results are shown for the Llama 2-7b and OPT-6.7b models. . . .	98
4.13	Performance of SAPLMA on a fact dataset generated from TruthfulQA. A 50-50 holdout strategy is adopted. Results are shown for the Llama 2-7b and OPT-6.7b models.	98
4.14	Performance of SAPLMA on a fact dataset generated from SQuAD 2.0, restricted to the validation set and the top-5 most popular topics. The original topic-wise leave-one-out strategy is adopted. Results are shown for the Llama 2-7b and OPT-6.7b models. . . .	99
5.1	Five different clusters of probing classifiers used in this study based on associated model type.	112
5.2	Summary of the number of samples and train-test splits for the four sentiment and emotion datasets used in this study.	113
5.3	Probing results across different pooling methods, reporting the top-3 best-performing combinations of layer/pooling method/prober for each of the datasets used in this study.	115
5.4	Comparison of SENTRILLAMA against DeBERTa, RoBERTa, and prompt-based method. The best-performing model is highlighted in bold , while the runner-up is <u>underlined</u>	122
5.5	Dataset-wise comparison of the different SENTRILLAMA sizes resulting from model compression (retaining only the input embedding and the layers up to the most representative layer).	123
5.6	Computational efficiency comparison of different SENTRILLAMA versions with DeBERTa, RoBERTa, and a prompt-based method. The reference dataset is SST-2.	123

5.7	Sequential recommendation performance: Recall@k (R_k) and nDCG@k (N_k). Results are reported as percentages. Best scores are bolded, second-best underlined. ID stands for ID-based, T stands for Text-based.	131
5.8	Comparison of text encoders for product search (nDCG@100) on the two datasets (results are percentages). “All” averages scores for all domains. We also report two domains, “Office” and “Sports”. .	132

This page intentionally left blank

Abbreviations

AI	artificial intelligence
CLM	causal language model
GTE	generalist text embedding
KB	knowledge base
KG	knowledge graph
KGC	knowledge graph completion
KRL	knowledge representation learning
LLM	large language model
LSTM	long short-term memory
MLP	multi-layer perceptron
MRR	mean reciprocal rank
NER	named entity recognition
NLP	natural language processing
PCA	principal component analysis
PLM	pretrained language model
PS	product search
RAG	Retrieval-Augmented Generation
RNN	recurrent neural network

SA sentiment analysis

SR sequential recommendation

Chapter 1

Introduction

The recent advent of pretrained language models (PLMs) and large language models (LLMs) has the potential to push new boundaries in many applicative scenarios. Due to their exposure to large textual corpora, these models exhibit rich understanding of language structure and meaning, and have shown unprecedented levels of performance in various applicative domains, including text summarization (Goyal et al., 2022), question answering (Kim et al., 2022), sentiment analysis (Zhang et al., 2020a), and relation extraction (Papanikolaou and Pierleoni, 2020).

The dominant engineering pattern remains: pre-train a large parametric model on broad, task- and domain-agnostic, textual data in an unsupervised manner, then adapt it to downstream tasks by updating parameters. Fine-tuning is resource-intensive, multiplying computational and energy costs with each new task. The computational cost of training large models is prohibitively high for all but a few institutions and organizations (Bommasani et al., 2021), limiting the ability of many individuals and communities to participate in their development. While modern parameter-efficient fine-tuning techniques (Houlsby et al., 2019) have emerged to lessen this computational cost, the paradigm still introduces other significant scientific and practical trade-offs. Notably, modifying model parameters, even a small subset, risks catastrophic forgetting (Li et al., 2024), eroding prior knowledge and general capabilities, while overfitting to narrow task distributions. Furthermore, fine-tuning for every task creates a fragmented and unsustainable ecosystem where knowledge is wastefully duplicated across countless model derivatives, each with unique training hyperparameters that hinder direct comparison and impede reproducible science.

While fine-tuning, either full or partial, remains the predominant approach

for adapting LLMs to downstream tasks, in-context learning (ICL) provides an alternative that does not require modifying model weights. As a purely behavioral approach, ICL provides no guarantee that a model’s generated output faithfully reflects its internal, latent knowledge. Models are known to produce responses that contradict their encoded understanding, manifesting as factual errors or *hallucinations* (Ravichander et al., 2025). This highlights a critical gap between a model’s behavioral output and its internal representational state.

A further practical constraint is the current industry landscape. Access to the internals of frontier LLMs is increasingly mediated by API endpoints, with weights, training data composition, and update schedules controlled by vendors. This limits inspectability (e.g., inability to analyze intermediate representations or ablate components), reduces experimental control (e.g., undocumented model revisions, safety filters, or rate limits that shift behavior over time), and weakens reproducibility (results may depend on provider-side changes outside the researcher’s control). The "pay-per-token" economic model also shifts costs from up-front training to ongoing usage, which can be prohibitive for large-scale experimentation and hinders open benchmarking.

This dissertation takes a complementary approach: rather than adapting parameters, it investigates what information is already encoded in *frozen* (i.e. non fine-tuned) PLMs and how to elicit explicit semantics from their hidden representations. The central question is whether the representational geometry of PLMs contains signals that can be read out or composed to support downstream behavior without weight updates. Throughout, *no PLMs were fine-tuned in the making of this thesis*.

Concretely, three questions were explored: (i) *explicit semantics*—can hidden representations be decoded into structured statements about real-world entities, types, and relations; (ii) *factuality*—do these representations encode information about the truthfulness of generated sentences; and (iii) do hidden representations encode properties beyond semantics and factual content, including sentiment and task-oriented utility for recommendation and search.

Methodologically, this thesis builds upon probing (Belinkov, 2022a) and elicitation procedures that operate on a model’s internal signals while keeping all weights fixed. By internal signals, we refer to measurable quantities available from a forward pass, e.g., hidden states in the residual stream and output-layer logits at masked positions, that can be inspected or combined without changing

parameters. Learnable predictors (e.g., a linear layer or scoring function) can then be used to translate these signals into labels, relations, or task scores. While these lightweight predictors still require their own training, they are orders of magnitude smaller and cheaper than fine-tuning the base PLM, offering a more sustainable trade-off.

This perspective contributes along two axes: (1) Interpretability: mapping internal signals to explicit semantics clarifies what the model encodes and supports transparent evaluation. (2) Sustainability: by showing the tasks and domains where frozen PLMs are sufficient (*and where they are not*) we can avoid unnecessary and costly fine-tuning.

The need for interpretability is underscored by a growing common concern about the lack of control over what these models learn. This is not merely a technical annoyance; it is a fundamental barrier to trust. The hallucinations, toxic and bias content that emerge from these models are symptoms of this underlying opacity. When a model fails, e.g., by hallucinating or generating harmful text, we must be able to ask why. Interpretability for PLMs remains largely an unsolved problem that asks for immediate attention, as it is the only path toward building models that are not just powerful but also provably safe and scientifically understood.

Sustainability, on the other hand, benefits from leveraging a model’s internal signals without modifying its weights. By using frozen representations (e.g., residual streams, attention head activations, and multi-layer perceptron (MLP) outputs) task-specific predictors can be trained efficiently, avoiding the high computational and financial costs of full fine-tuning. This approach not only reduces energy consumption and storage requirements but also enables rapid prototyping and deployment of AI solutions across multiple tasks.

In practice, sustainable use of LLMs allows industry to reuse pretrained knowledge, scale AI capabilities more efficiently, and minimize operational overhead. By interpreting and harnessing hidden representations, organizations can achieve high performance on downstream tasks while maintaining environmental and economic efficiency, making advanced AI both practical and responsible for industrial applications.

1.1 Thesis Goal and Contributions

Context: A Summary of the Problem. *The predominant "pre-train/fine-tune" paradigm, while effective, is computationally costly, unsustainable, and leads to a fragmented ecosystem of model derivatives. This dissertation argues for a complementary approach: investigating what knowledge is already encoded in frozen PLMs and how to elicit it via probing their internal signals. This method avoids costly parameter updates, providing a foundation for more interpretable and sustainable AI.*

Building on these motivations, this dissertation contributes to the fields of Natural Language Processing and Machine Learning along two primary axes: interpretability and sustainability of pretrained language models (PLMs).

- **Interpretability:** By developing methods that map opaque internal signals, such as hidden states and output-layer logits, to explicit, human-verifiable semantics, this thesis contributes to understanding what PLMs learn from pre-training data. It provides a methodology for evaluating the intrinsic knowledge structure within a model’s parameters, complementing standard end-to-end performance metrics.
- **Sustainability:** By demonstrating the effectiveness of frozen models across a range of tasks, this research helps to delineate where the computationally expensive process of fine-tuning is necessary and, crucially, where it is not. This provides a blueprint for a more sustainable and efficient research and deployment paradigm, one that reduces the computational, financial, and environmental costs associated with the unchecked proliferation of fine-tuned model derivatives.

Specifically, the main contributions can be summarized as follows:

- We show that frozen pretrained language models (PLMs) exhibit an understanding of *semantic containment*—the inclusion of one type or entity within a more abstract one (e.g., “Obama $\rightarrow$ person,” “President $\rightarrow$ person”), which reflects their inner *type knowledge*. We also hint at how this information can be leveraged for practical semantic web applications, such as entity typing and ontology completion (Section 3.1).

- We demonstrate how the latent type knowledge encoded in PLMs can be leveraged to improve *inductive relational inference* on graphs, mitigating common limitations caused by incomplete ontologies and sparse graph structures, overcoming the need for explicit type annotations in knowledge graphs and enabling more accurate and robust predictions (Section 3.2).
- To enhance trust and interpretability of LLMs, we analyze whether hidden representations encode truthfulness information and propose a robust methodology to probe LLMs for *factuality self-evaluation*, i.e., the model’s ability to assess the factuality of its own outputs (Section 4.2).
- We investigate whether and how *sentiment* and *emotion* information is encoded in the hidden layers of frozen LLMs. We also demonstrate that recent LLMs can be efficiently compressed without sacrificing performance on sentiment and emotion recognition tasks, thereby improving computational efficiency (Section 5.1).
- We explore the applicability of *generalist text embedding (GTE)* models for *recommendation* and *search* tasks without task-specific fine-tuning, challenging the prevailing assumption that fine-tuning a PLMs on a task-specific objective is needed to reach optimal performance; we show that GTE models, trained on large, domain-agnostic, embedding-based objectives, outperform fine-tuned alternatives (Section 5.2).

1.2 Thesis Outline

This thesis is organized into thematic chapters, each addressing the foundational concepts, analyses, and proposals relevant to the study of explicit knowledge in pretrained language models (PLMs) hidden representations. The ordering of topics across chapters is not strictly chronological with respect to the papers publication. Instead, the structure is designed to develop a coherent narrative.

Chapter 1 introduces the motivation, objectives, and overall contributions of the thesis, situating the research within the broader context of natural language processing and machine learning.

Chapter 2 provides the necessary background, offering formal definitions and conceptual grounding in PLMs, their recent development, and open challenges.

These foundations are essential for understanding the discussions in the subsequent chapters.

In Chapter 3, we investigate how explicit semantics are encoded and represented within pretrained language models. This includes analyses of semantic knowledge, its interpretability, and its implications for downstream tasks.

Chapter 4 turns to the issue of factuality in large language models. Here, we examine both the strengths and limitations of current approaches, exploring when and why models succeed or fail in generating factually accurate content.

In Chapter 5, we analyze the hidden capabilities of frozen PLMs, such as their ability to perform sentiment analysis, zero-shot recommendation, and search. This chapter highlights emergent behaviors that extend beyond models' explicit training objectives, with implications for both practical applications and theoretical understanding.

Finally, Chapter 6 concludes the thesis by summarizing the key findings and outlining potential directions for future research. Particular emphasis is placed on bridging the identified gaps and advancing the reliable, interpretable, and safe use of PLMs and LLMs.

Chapter 2

Background

This chapter outlines the essential concepts and technologies on which this thesis is built, establishing the theoretical and technical basis for the research that follows.

2.1 Language Models

The problem of *language modeling* consists in assigning a probability to a sequence of words. More generally, the goal is to assign a probability to a sequence of *tokens*, where a token is a basic unit of text such as a word, subword, or character, depending on the chosen representation. Let a sequence of tokens be denoted as

$$S = (t_1, t_2, \dots, t_T).$$

The goal of language modeling is to compute the joint probability

$$p(S) = p(t_1, t_2, \dots, t_T).$$

This probability is typically factorized using the chain rule of probability, decomposing the joint distribution into a product of conditional probabilities:

$$p(S) = \prod_{t=1}^T p(t_t \mid t_1, t_2, \dots, t_{t-1}).$$

This formulation recasts the problem as one of sequential prediction: at each time step t , the model must predict the probability of the next token t_t given the history of all preceding tokens $t_{<t}$. The efficacy with which a model performs this

task is a direct measure of its linguistic competence.

The problem of language modeling was originally addressed through approaches influenced by statistics, such as n-gram, based on approximating the conditional probabilities with frequency counts from given corpora. Unfortunately, the need to define a fixed vocabulary of n-grams, along with the strict dependency from the observed data, make statistical models such as n-gram rather unsuitable for this task. For this reason, statistical models were largely superseded by deep learning models, which estimate the probabilities of words by means of neural networks. Bengio et al. (2003) introduced the concept of *neural probabilistic language models*, marking a foundational step in the evolution of neural language modeling. The authors proposed learning a model f_{Θ} that approximates the true conditional distribution of a word given its preceding context within a window of size N :

$$p(t_t \mid t_{1:t-1}) \approx f_{\Theta}(t_{t-1}, \dots, t_{t-N}).$$

In this formulation, the model f_{Θ} represents each word in the vocabulary using a *distributed representation*—commonly known as *word embedding* (Mikolov et al., 2013a). The network, consisting of a feed-forward architecture and an embedding matrix, is trained end-to-end via backpropagation to jointly optimize all parameters. The key intuition is that the model learns to assign similar feature vectors to semantically related words—for instance, the vector for “King” becomes numerically close to “Queen,” but distant from “Banana”—thereby capturing linguistic regularities in the embedding space. As a result, the learned probability function becomes a *smooth* mapping: small changes in the feature space, such as those induced by synonyms or related terms, produce correspondingly small variations in predicted probabilities.

Neural-based LMs can be roughly broken down into two families: *unidirectional* and *bidirectional* models. While the former takes into account only the preceding word in a sequence as its context, the latter also takes into account the following words. In an initial phase, neural LMs were mostly based on recurrent neural networks (RNNs) and long short-term memory (LSTM) architectures. ELMo (Peters et al., 2018), as an example, was introduced as a bidirectional language model implemented by means of a bidirectional LSTM that learned contextual, task-specific embeddings for words as a linear combination of the intermediate layers of the architecture.

2.1.1 The Rise of the Transformer Architecture

The limitations of recurrent architectures set the stage for a revolutionary development in NLP: the **Transformer** architecture. Introduced by Vaswani et al. (2017) in their seminal paper *Attention Is All You Need*, this architecture departed entirely from sequential recurrence and convolutions, relying instead on *self-attention*.

This shift had two immediate consequences. First, by eliminating sequential dependencies, the Transformer enabled massive *parallelization*: all tokens in a sequence could be processed simultaneously. This reduced training time and allowed scaling to much larger datasets, laying the groundwork for modern large language models. Second, self-attention directly modeled relationships between any two tokens, regardless of distance. Whereas RNNs require a path length of $O(T)$ to connect the first and last tokens of a sequence, the Transformer reduces this to $O(1)$. This structure makes it far more effective at capturing *long-range dependencies*, a persistent weakness of recurrent models.

Architectural Deep Dive: Self-Attention

The core innovation of the Transformer is the **self-attention** mechanism. This mechanism allows the model to understand context by dynamically weighing the importance of all other tokens in a sequence when representing a single token. For example, in the sentence “The animal didn’t cross the street because *it* was too tired,” self-attention allows the model to learn to associate “it” strongly with “animal,” creating a context-aware representation.

This is achieved using an architecture analogous to a retrieval system. An attention function maps a **query** and a set of **key-value** pairs to an output. Intuitively, for a given token i :

- Its **query** (q_i) vector acts as a request, asking, “Who in this sequence is relevant to me?”
- The **key** (k_j) vector of every other token j in the sequence acts as a label, advertising, “This is what I am.”
- The **value** (v_j) vector of token j represents its actual content or meaning.

The model matches the query q_i against all keys k_j to produce attention scores (i.e., compatibility). These scores are then used as weights to compute a weighted

sum of all the values v_j . The resulting output for token i is a new representation that is blended with the information from the most relevant tokens in the sequence.

In self-attention, these queries, keys, and values are not static; they are all derived from the same source: the layer's input embeddings x_i . Three distinct, learned weight matrices (W^Q, W^K, W^V) are used to project the input into these three roles:

$$q_i = x_i W^Q, \quad k_i = x_i W^K, \quad v_i = x_i W^V.$$

The attention score ω_{ij} between tokens i and j is computed using a dot product:

$$\omega_{ij} = q_i \cdot k_j.$$

These scores are scaled by $\sqrt{d_k}$ (the dimension of the key vectors) to stabilize gradients, and then normalized using the softmax function to create the final attention weights α_{ij} :

$$\alpha_{ij} = \text{softmax}\left(\frac{q_i k_j^\top}{\sqrt{d_k}}\right).$$

The output representation z_i for token i is the weighted sum of all values:

$$z_i = \sum_j \alpha_{ij} v_j.$$

In matrix form, this entire operation is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V.$$

The concept of attention had already appeared in the literature, notably in the context of neural machine translation (Bahdanau et al., 2015), where it was introduced alongside RNNs to enable the model to focus on relevant parts of the input sequence when generating each output token. The key innovation of the Transformer architecture was to eliminate recurrence entirely and rely solely on self-attention mechanisms to model contextual dependencies between tokens. The main benefit of this shift was that attention mechanisms are inherently *parallelizable*, allowing all token representations within a sequence to be computed simultaneously rather than sequentially, as in RNNs. This architectural change dramatically improved training efficiency and scalability, enabling the development of much larger models and the foundation of modern PLMs.

In addition, instead of relying on a single attention mechanism, the Transformer introduces **multi-head attention**, which projects the input queries (Q), keys (K), and values (V) into h different subspaces through parallel linear transformations. Each *attention head* operates independently, allowing the model to capture diverse types of relationships, such as syntactic dependencies, semantic similarity, or long-range context within the sequence. The outputs from all heads are then concatenated and linearly combined to form the final representation, preserving information from multiple perspectives.

Because self-attention by itself is *permutation-invariant*, it lacks an inherent notion of sequence order. To address this, **positional embeddings** are added to the input token embeddings, injecting information about word order. In the original Transformer, these positional embeddings were defined using sinusoidal functions of varying frequencies, enabling the model to generalize to sequences longer than those seen during training.

Encoder-Decoder Structure. The original Transformer, designed for sequence-to-sequence tasks like machine translation (Vaswani et al., 2017), employed an **encoder-decoder** architecture. This design consists of two distinct stacks of components: a stack of N identical **encoder layers** and a stack of N identical **decoder layers**.

The **encoder** stack's role is to process the entire input sequence (e.g., a sentence in French) and transform it into a set of continuous, context-rich vector representations. Each encoder layer in the stack has two main sub-layers:

1. **Multi-Head Self-Attention:** this layer allows each token in the input sequence to attend to all other tokens in the *input sequence*, building a representation that incorporates its full context.
2. **Position-wise Feed-Forward Network:** this is a simple, fully connected neural network that is applied to each token's representation independently. It processes the information gathered by the self-attention mechanism and adds non-linear representational capacity.

The **decoder** stack's role is to generate the output sequence (e.g., the translated sentence in English) token by token. This is an *autoregressive* process, meaning that to generate the next token, the decoder receives the output of the encoder and the sequence of tokens it has *already* generated as input. Each decoder layer is slightly more complex, containing three sub-layers:

1. **Masked Multi-Head Self-Attention:** This layer allows each token in the *output sequence* to attend to other tokens, but only those that precede it. This **causal masking** is critical; it ensures that the prediction for a token at position t can only depend on the known outputs at positions less than t , preventing the model from “cheating” by looking at future tokens.
2. **Encoder-Decoder Cross-Attention:** This is the crucial link between the two stacks. This layer’s *queries* come from the output of the masked self-attention sub-layer (representing the output being built), while its *keys* and *values* come from the final output of the encoder stack (representing the entire input). This allows every token being generated by the decoder to attend to all tokens from the input sequence, focusing on the most relevant parts of the input for generating the current output token.
3. **Position-wise Feed-Forward Network:** This sub-layer is identical in function to the one in the encoder.

To enable the training of such a deep architecture, both the encoder and decoder layers employ two key stabilization techniques. Each sub-layer (self-attention, cross-attention, and feed-forward) has a **residual connection** around it, followed by **layer normalization** (Ba et al., 2016). The full operation for a sub-layer is $\text{LayerNorm}(x + \text{SubLayer}(x))$. The residual connections allow gradients to flow unimpeded during backpropagation, mitigating the vanishing gradient problem, while layer normalization stabilizes the activations, accelerating training.

2.1.2 Pre-training and Fine-tuning Paradigm

Alongside the Transformer came a paradigm shift: self-supervised pre-training followed by task-specific fine-tuning. A large model is first trained on massive unlabeled corpora using automatically generated objectives (e.g., predicting missing words). The resulting representations capture syntax, semantics, and factual knowledge. Fine-tuning on smaller labeled datasets then adapts the model to specific tasks, reducing data requirements and enabling broad applicability.

Causal Language Modeling (CLM) autoregressively predicts the next token in a sequence,

$$P(w_t \mid w_{<t}),$$

and is central to models like the GPT series. This approach is highly effective

for text generation, but its main limitation is the lack of bidirectional context, as each prediction depends only on preceding tokens.

Masked Language Modeling (MLM), popularized by BERT (Devlin et al., 2019), predicts randomly masked tokens using full bidirectional context, resulting in strong representations for natural language understanding. MLM, however, introduces a pre-train/fine-tune mismatch because the [MASK] token is absent during downstream tasks, and standard MLM objectives often ignore dependencies among multiple masked tokens within the same sequence.

Building on or extending these paradigms, other objectives have been proposed. XLNet (Yang et al., 2019) uses *Permutation Language Modeling* to combine autoregression with bidirectional context without relying on [MASK] tokens. ELECTRA (Clark et al., 2020) introduces *Replaced Token Detection (RTD)*, training a discriminator to identify tokens replaced by a generator, improving sample efficiency. Higher-level objectives include *span corruption* for T5 (Raffel et al., 2020), which frames tasks as text-to-text reconstruction, and *Sentence Order Prediction (SOP)* (Lan et al., 2020), designed to better capture discourse coherence.

Note on terminology: the expression *large language model (LLM)* broadly refers to a class of PLMs that are trained on vast amounts of textual data, often spanning billions or even trillions of tokens, and characterized by a substantial number of internal parameters, typically in the order of billions. In general, there is no common consensus on when a model starts to become effectively "large". In this thesis, the two terms will be used largely interchangeably, with *LLM* referring to *PLMs* of sufficient scale for emergent behaviors and scaling effects to become apparent.

2.2 *Parametric Knowledge: Storage, Recall, and Manipulation of Facts in PLMs*

The paradigm of knowledge representation in artificial intelligence has historically been dominated by structured, symbolic systems. For decades, structured knowledge bases (KBs) have served as repositories of world knowledge, meticulously curating facts into explicit, machine-readable formats such as subject–predicate–object triples. Projects such as DBpedia (Mendes et al., 2012) and Wikidata (Vrandečić and Krötzsch, 2014) exemplify this tradition.

The advent of large-scale pretrained language models (PLMs), however, has

introduced the concept of *parametric knowledge*. Through self-supervised learning on massive unstructured corpora, PLMs implicitly encode extensive world knowledge directly within their billions of parameters. This section investigates this phenomenon, charting its discovery, underlying mechanisms, limitations, and the frontier of research into controllable, hybrid knowledge systems.

2.2.1 Language Models as Knowledge Bases?

The rise of LLMs has sparked debate on whether these systems will be able to replace traditional structured knowledge sources (Sun et al., 2024). *Factual recall* is operationalized as the ability of a PLM to predict the object of a relational triple when prompted with its subject and relation.

The seminal work of Petroni et al. (2019) demonstrated that PLMs can recall factual knowledge without task-specific supervision. Using cloze-style probes such as “Dante was born in [MASK],” they showed that models like BERT could retrieve relational facts with competitive accuracy. This introduced the LAMA benchmark and reframed PLMs as potential unsupervised, open-domain QA systems. Later benchmarks, e.g., TruthfulQA (Lin et al., 2022), assessed truthfulness and resistance to misconceptions.

How do LLMs store factual information about real world entities? Geva et al. (2021) proposed the *key-value hypothesis*: MLP layers act as associative memories, with the first projection serving as keys and the second as values. Later work framed both MLPs and attention heads as distributed associative memories, showing storage capacity scales linearly with parameters. More recent causal-tracing studies revealed that factual recall unfolds as an *enrich-propagate-extract* process distributed across layers, involving both MLPs and attention heads. The direct localization of knowledge in PLMs is a prerequisite for *knowledge editing*: the targeted modification or removal of previously acquired information. While pioneering methods such as ROME (Meng et al., 2022) have shown promise, a fundamental challenge remains. The editing process invariably introduces some degree of collateral damage, unintentionally altering the model’s factual recall capabilities beyond the intended target.

Studies injecting synthetic facts during pre-training reveal that facts are acquired incrementally and forgotten non-linearly, with forgetting following a power-law decay. This implies a *learnability threshold*: unless a fact reappears frequently enough, it is forgotten before reinforcement. Larger models absorb knowledge

more efficiently per exposure, deduplication fosters robust generalizations, and larger batch sizes improve resistance to forgetting.

2.2.2 Hallucinations and Open Challenges

Despite their impressive capabilities, claims that PLMs could serve as comprehensive knowledge bases, or even replace traditional symbolic systems, remain largely utopian in the near term. In fact, the knowledge that is embedded in LLMs is hard to evaluate extensively, as these models are prone to generating inaccurate information. It remains unclear whether these errors stem from gaps in their stored knowledge or from flaws in their internal mechanisms.

Besides hallucinations, PLMs continue to face persistent challenges (Pan et al., 2023) related to long-tail knowledge and temporal accuracy. Their outputs often lack explicit provenance, their probabilistic nature can foster overconfidence, and their parametric representations are difficult to update without unintended side effects. These issues are not merely engineering limitations but arise from the fundamental objective of modeling statistical regularities in text rather than grounding predictions in objective truth. Although this thesis does not directly address these other limitations, it recognizes them as crucial considerations and promising directions for future research.

2.3 Interpretability of PLMs Hidden Representations

The internal representations ¹ of pretrained language models (PLMs) are high-dimensional vectors that capture rich linguistic and semantic information learned from large-scale text corpora. Despite their expressiveness, these representations remain largely opaque, forming a *black box* that conceals the model’s internal reasoning and knowledge organization. To address this challenge, *probing* techniques have emerged as a prominent approach to interpretability, providing systematic methods to *peer inside the box* and to examine the information encoded within

¹**Note on terminology:** In this thesis, we define as *hidden representations* all the numerical vectors produced by a model during a forward pass that encode information about the input. This includes intermediate states such as residual stream vectors, attention head activations, and MLP activations. In a broader sense, one could also include final pre-softmax logits, since they are derived from the model’s internal processing.

model representations. This section reviews recent developments in this area and outlines the conceptual foundations underlying the methods central to this thesis.

2.3.1 Probing of Representations

Probing (Alain and Bengio, 2017; Belinkov, 2022b) is a global explanation technique that analyzes a model internal representations by training a simpler, auxiliary model—the “probe”—to predict specific properties from those representations. The performance of this probe, typically measured by accuracy, is then interpreted as a proxy for *how well* the original model encodes the target property, as it effectively serves to answer the question: “*given the static features generated by the PLM, what linguistic properties can a new model learn from them?*”. The fundamental objective is to translate the dense numerical vectors of a PLM hidden states into human-understandable concepts, such as part-of-speech (POS) tags, syntactic dependencies, semantic roles, or the presence of factual knowledge. This approach provides a more granular and nuanced evaluation of a model’s learned capabilities compared to simply measuring its performance on downstream tasks. The classic approach to probing involves training a separate classification model on the frozen hidden-state representations extracted from a PLM. During the training of the probe, which can be a simple model such as a logistic regression classifier or a more complex one like a multi-layer perceptron (MLP), the parameters of the underlying PLM are kept fixed.

More formally, the probing framework involves several key components. Let f denote the original model trained on a dataset $\mathcal{D}_O$. This model produces a set of intermediate representations (e.g., residual stream states, MLP activations, etc.) where $f_l(x)$ represents the internal representation of an input x at layer l . A separate probing classifier g is then trained on a probing dataset $\mathcal{D}_P = \{(f_l(x^{(i)}), z^{(i)})\}$ to predict a specific property z . The resulting performance, $\text{PERF}(g, \mathcal{D}_P, f, \mathcal{D}_O)$, can be expressed as a function of all these components, emphasizing that the outcome of a probing experiment depends jointly on the probe design, the probing data, and the characteristics of the original model and its training.

Perhaps one of the most striking discoveries to emerge from probing research is the phenomenon of “*hidden knowledge*” (Gekhman et al., 2025): PLMs often encode information correctly in their internal representations even when they fail to express that information in their generated output.

In the context of NLP, research has evolved from early analyses of static word embeddings like Word2Vec (Yaghoobzadeh et al., 2019) and GloVe (Klubicka and Kelleher, 2022) to methods investigating the complex layered knowledge within LLMs (Chen et al., 2023; Pirozelli et al., 2024). Prior work has probed LLMs for various fine-grained properties, including linguistic properties (Vulic et al., 2020), factual knowledge (Petroni et al., 2019; Wu et al., 2023; De Bellis et al., 2024), beliefs (Azaria and Mitchell, 2023), cross-lingual alignment (Wang et al., 2024a), logical reasoning capabilities (Manigrasso et al., 2024), privacy leakages (Kim et al., 2023; Di Palma et al., 2025c), toxicity (Wen et al., 2023; Roy et al., 2023). These studies approach probing either through a prompt-based method, where the LLM’s performance is evaluated using specifically designed prompts, or by applying trained classifiers to analyze the model’s intermediate layers.

2.3.2 Recent Advances and Causal Probing

Standard probing is fundamentally an *observational method*: it reveals whether certain information is encoded in the internal representations of a model, but not whether the model actually uses that information to make predictions. This distinction is crucial, as the detected signal may simply reflect a correlational artifact rather than a functional dependency.

Causal probing (Canby et al., 2024) aims to address this limitation by moving beyond correlation toward causation, by investigating whether internal representations play a causal role in a model’s decision-making process. It is a core component of *mechanistic interpretability* (Olah, 2022; Elhage et al., 2022), a field dedicated to reverse-engineering neural networks by mapping specific computations to understandable components, such as individual neurons or attention heads. Within this framework, *circuit discovery* (Conmy et al., 2023; Marks et al., 2025) refers to methods that identify the computational subgraphs responsible for a target behavior. These subgraphs consist of nodes (e.g., neurons, attention heads, embeddings) and interaction edges (e.g., residual connections, attention flows) that together execute specific linguistic or reasoning tasks.

Among others, Nikankin et al. (2025) use causal analysis to show that large language models solve simple arithmetic tasks by encoding basic heuristics at the neuron level. Similarly, Niu et al. (2025) leverage circuit discovery to identify attention heads that induce distraction effects, causing the model to overemphasize tokens appearing in the input. Although this thesis does not directly employ

causal probing techniques, these approaches have recently gained substantial attention within interpretability research and represent a promising direction for future work.

2.4 Language Models and Knowledge Graphs

Symbolic knowledge bases (KBs) store information explicitly: a fact such as “Paris is the capital of France” exists as a discrete entry that can be retrieved through direct lookup. In contrast, pretrained language models (PLMs) encode such facts implicitly within distributed patterns of weights and biases learned from pre-training. While this implicit encoding enables greater contextual flexibility compared to rigid symbolic systems, it also introduces challenges in verifiability, interpretability, and controllability of the stored knowledge.

This section explores the interplay between language models and knowledge graphs, two paradigms that approach knowledge representation from complementary perspectives. Despite their differences, both aim to capture, organize, and utilize knowledge to support reasoning and inference. The analogy between them lies in how PLMs can be viewed as implicit, continuous knowledge stores, while knowledge graphs serve as explicit, discrete repositories. Recent research has increasingly focused on integrating these paradigms, recognizing that knowledge graphs can provide structure, grounding, and interpretability to PLMs, whereas PLMs can enrich and expand knowledge graphs by extracting, contextualizing, and generating knowledge from text. Together, they form the foundation for hybrid *neuro-symbolic* systems that combine the strengths of symbolic reasoning with the adaptability and generalization capacity of neural models.

2.4.1 Knowledge Graphs

Knowledge graphs (KGs) (Hogan et al., 2021) are a structured paradigm for knowledge representation, modeling factual information as a multi-relational, directed graph. In this formalism, nodes represent entities (e.g., **Paris**, **France**) and labeled, directed edges represent relations between them (e.g., **is_capital_of**). Each fact is thereby encoded as a triple in the form of (subject, predicate, object)—or, in knowledge representation learning (KRL) literature, (head, relation, tail)—such as (**Paris**, **is_capital_of**, **France**).

Formally, a KG is defined as $G = (E, R, F)$, where E is the set of entities, R

is the set of relations, and $F \subseteq E \times R \times E$ is the set of observed facts (triples). Each fact $f \in F$ is denoted (h, r, t) , with $h \in E$ (head), $r \in R$, and $t \in E$ (tail).

A fundamental challenge in the practical application of large-scale KGs is their inherent incompleteness and sparsity; they typically capture only a small subset of the true facts that hold in the world. This limitation motivates the task of **link prediction**, where the objective is to infer missing but valid triples, such as identifying the tail entity t in the query $(h, r, ?)$, e.g., **(Rome, is_capital_of, ?)**.

The field of knowledge representation learning (KRL) addresses this challenge by mapping the discrete, symbolic components of the graph (entities E and relations R) into a low-dimensional, continuous vector space $\mathbb{R}^d$. These *embeddings* are learned such that the semantic properties and relational structure of the graph are preserved in the geometry of the embedding space.

KRL models for knowledge graph completion (KGC) are trained by optimizing a plausibility scoring function, $f(h, r, t)$. This function is designed to assign high scores to valid (observed) triples and low scores to invalid or unobserved triples (often generated via negative sampling). For instance, the foundational translational model TransE (Bordes et al., 2013) learns vector representations $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ and defines its scoring function based on a distance metric, modeling the relation $\mathbf{r}$ as a translation from the head $\mathbf{h}$ to the tail $\mathbf{t}$ (i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$).

At inference time, to answer a query like **(Rome, is_capital_of, ?)**, the model computes the score $f(\text{Rome}, \text{is_capital_of}, e')$ for all $e' \in E$. These candidate entities are then ranked by their plausibility scores. Performance is evaluated using standard, rank-based metrics, primarily mean reciprocal rank (MRR) (the average reciprocal rank of the correct entity) and Hits@ k (the proportion of queries where the correct entity is ranked in the top k results). Beyond KGC, these learned embeddings serve as potent, general-purpose representations for various downstream tasks, including entity classification, relation prediction, and clustering.

2.4.2 Knowledge Graphs for PLMs

Knowledge graphs play an essential role in improving PLMs for two key purposes (Pan et al., 2024): (1) **Interpretability** — understanding what a PLM *knows*; and (2) **Adaptation** — manipulating, updating, or constraining a PLM’s internal knowledge to produce more KG-grounded outputs.

KG-enhanced PLMs (Pan et al., 2024) incorporate knowledge graphs during the pre-training and inference stages, or for the purpose of interpreting the internal knowledge learned by the models. Integrating KGs can significantly improve both the performance and interpretability of PLMs across diverse downstream tasks. Research in this area can be broadly categorized into three directions: (1) *KG-enhanced PLM pre-training*, which integrates KGs during the pre-training stage to enrich the factual and relational understanding of PLMs; (2) *KG-enhanced PLM inference*, which utilizes KGs at inference time to provide access to up-to-date knowledge without requiring retraining; and (3) *KG-enhanced PLM interpretability*, which employs KGs to analyze, interpret, and explain the reasoning processes and factual grounding of PLMs.

KGs have been leveraged to probe pretrained language models (Youssef et al., 2023) and examine the factual knowledge encoded within them. Such probing has been applied to general knowledge domains (Petroni et al., 2019) as well as specialized fields, including biomedical (Sung et al., 2021) and scientific (Hardalov et al., 2020) contexts.

2.4.3 PLMs for Knowledge Graphs

Conversely, PLMs can be employed to enhance KGs, giving rise to *PLM-augmented knowledge graphs* (Pan et al., 2024). Traditional KG-based methods often struggle with incompleteness and lack of generalization. PLMs, with their strong generalization and linguistic understanding, offer promising avenues for overcoming these limitations.

Research on PLM-augmented KGs can be categorized into five primary directions (Pan et al., 2024): (1) *PLM-augmented KG embedding*, which enriches KG representations by encoding textual descriptions of entities and relations through PLMs; (2) *PLM-augmented KG completion*, which leverages PLMs to infer or generate missing facts for improved KGC performance; (3) *PLM-augmented KG construction*, which applies PLMs to entity discovery, coreference resolution, and relation extraction for building or updating KGs; (4) *PLM-augmented KG-to-text generation*, which employs PLMs to produce natural language descriptions of facts stored within a KG; and (5) *PLM-augmented KG question answering*, which bridges the gap between natural language queries and structured knowledge retrieval from KGs.

A straightforward application involves using PLMs as text encoders for KG-

related tasks. By encoding textual descriptions of entities and relations, PLMs can generate semantically rich embeddings that enhance KG representations (Anelli et al., 2022), or be used to infer missing links in KGs by reframing their language modeling task as triple completion (Yao et al., 2019).

Early research (Wu et al., 2016) explored integrating textual information into KRL, seeking unified representational spaces that combine text and structured knowledge. This integration has also proven beneficial for Semantic Annotation (Wang et al., 2022), where textual concepts are identified and linked to corresponding entities in a KG.

PLMs can also be used to process large text corpora and extract entities and relations to construct or expand KGs.

Together, these research directions illustrate a growing synergy between symbolic and neural paradigms, where structured knowledge graphs provide explicit factual grounding, and pretrained language models contribute flexible generalization and linguistic understanding. This interplay forms the foundation for hybrid neuro-symbolic systems that aim to combine the strengths of both worlds.

Chapter 3 will further explore this bidirectional connection between PLMs and structured knowledge, leveraging structured information to probe the understanding encoded in PLMs and utilizing latent information in PLMs to enhance graph-based relational inference.

This page intentionally left blank

Chapter 3

Explicit Semantics in PLMs

In its broadest sense, **semantics** concerns the study of meaning. Within the fields of knowledge representation and the *semantic web*, semantics refers to the formal specification of how symbols and expressions correspond to entities, concepts, and relations in a given domain of discourse. By providing an explicit, machine-interpretable meaning layer, semantics enables computational systems to move beyond the manipulation of raw strings towards interpreting, reasoning about, and integrating data in a consistent and interoperable way.

Traditionally, explicit semantics are realized through structured knowledge representations such as ontologies and knowledge graphs, often implemented using standards like the *Resource Description Framework* (RDF) and the *Web Ontology Language* (OWL). In these frameworks, entities are uniquely identified using Uniform Resource Identifiers (URIs), and relationships between them are represented as triples of the form:

$$(Dante, \text{bornIn}, Florence)$$

Such explicit representations enable several capabilities that are central to the semantic web vision: (i) querying data using standardized languages such as SPARQL, (ii) deriving new facts through logical inference (e.g., if Dante was born in Florence and Florence is in Italy, then Dante was born in Italy), and (iii) ensuring consistency through well-defined ontological constraints. This is critical for enabling interoperable, verifiable, and reasoning-capable systems.

The importance of explicit semantics is not merely theoretical; it underpins many industrial-scale applications across diverse domains. In modern data-driven environments, organizations increasingly face the challenge of integrating heteroge-

neous, distributed, and rapidly evolving datasets. Without a shared semantic layer, data from different sources often remains isolated, inconsistent, or ambiguous.

Explicit semantics address key challenges by providing a common, formal meaning layer that enables seamless data integration and interoperability, allowing enterprises to merge information from disparate databases, APIs, and ontologies into unified knowledge graphs. For instance, large e-commerce platforms can combine product catalogs, supplier metadata, and customer behavior logs into semantically consistent models. They also enhance semantic search and information retrieval by reasoning over entity relationships, which allows search engines to deliver more relevant and context-aware results; practical applications include personalized recommendations, legal document retrieval, and biomedical literature search. Moreover, explicit semantics support compliance and explainability in regulated sectors such as healthcare, finance, and autonomous systems by providing transparency and traceability through explicit modeling of decision-relevant knowledge and rules. Finally, they facilitate knowledge management at scale, enabling companies to structure vast amounts of domain knowledge to support applications such as conversational agents, digital twins, and predictive maintenance.

In all these scenarios, semantics transform raw data into actionable knowledge by making its meaning explicit and machine-interpretable. This capacity to query, reason and explain is essential for robust, trustworthy, and scalable AI systems. In contrast, pretrained language models (PLMs) such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2019) acquire semantics *implicitly* by optimizing predictive objectives over massive unstructured corpora. Rather than encoding meaning through formal ontologies, these models learn distributed representations that capture statistical regularities in language. For example, when presented with the cloze-style query “Dante was born in [MASK] in the year 1265”, a model like BERT can often predict “Florence” correctly. However, this success arises not from explicitly storing the symbolic fact (*Dante*, *bornIn*, *Florence*), but from leveraging distributional co-occurrence patterns found in its training data. As a result, the semantics captured by PLMs remain *latent* and *opaque*, lacking the explicit structure necessary for symbolic reasoning.

The distinction between implicit statistical semantics and explicit symbolic semantics is central to understanding the role of PLMs in knowledge representation. A seminal study by Petroni et al. (2019) explored this question by asking: “Can language models serve as knowledge bases?” Using the LAMA (Language Model

Analysis) probe, they reformulated facts from structured knowledge bases into cloze-style templates to test whether PLMs could recover the correct answers. Their findings revealed that PLMs can indeed recall certain types of relational knowledge with surprising accuracy. However, performance varies considerably across relation types, one-to-one relations (e.g., place of birth) are captured more reliably than many-to-many or compositional ones. In addition to factual knowledge, other work (Wu et al., 2023) has shown that PLMs store *ontological knowledge* (including types of entities, hierarchical relationships between classes and properties, domain and range constraints of properties) and have, to some extent, semantic understanding of this knowledge rather than shallow memorization.

Building on the pioneering work of Petroni et al. (2019), subsequent research has increasingly sought to characterize the nature of factual knowledge encoded within PLMs, framing them as “soft” knowledge bases (Youssef et al., 2023). Yet, the knowledge embedded in PLMs is not represented in an explicit, verifiable symbolic form, resulting in limited transparency, interpretability, and logical consistency. While these findings demonstrate that PLMs exhibit impressive factual recall, they also underscore a fundamental limitation: such knowledge is not grounded in an explicit structure that supports symbolic reasoning, explanation, or seamless integration with external data sources.

Our central hypothesis is that by processing large volumes of text, PLMs learn representations whose structure has a decodable correspondence to the explicit symbolic knowledge codified in knowledge graphs (KGs). Uncovering these structural alignments is the goal of this work. We aim to build procedures that can map the implicit knowledge latent within PLMs to explicit semantics, that is, to symbolic facts that can be verified against a ground truth. Uncovering these structural alignments could drive progress in the field, enabling new semantic applications and facilitating the extraction of novel facts directly from PLMs.

This investigation is directly motivated by our previous study (Anelli et al., 2022), which provided the foundational evidence for this hypothesis. In that work, we established that BERT latent space contains semantically meaningful regions. By leveraging an open-domain KG, we successfully assessed the model’s capacity for factual recall and verified that distinct concepts from the KG’s taxonomy were separable within its representations. Those findings provide the foundation for our present broader exploration into the structural parallels between the latent

spaces of modern PLMs and the formal structure of KGs.

Contributions. As discussed in Section 2.4, there is a rich and mutually beneficial interplay between KGs and PLMs. This chapter investigates how PLMs understand and encode **explicit semantics**, focusing on type knowledge and hierarchical type relationships grounded in real-world KGs. It also examines how such knowledge can be leveraged in *KGs-for-PLMs* applications (Section 2.4.2), such as entity typing and zero-shot named entity recognition (NER), as well as in *PLMs-for-KGs* applications (Section 2.4.3), such as knowledge graph completion (KGC).

Our main contributions are centered on the semantic knowledge acquired by pretrained language models (PLMs) through extensive pre-training on large corpora. First, we provide an example of how real-world KGs can be used to probe PLMs understanding of a semantic relationship (semantic containment). Then, we show how, in turn, PLMs can be directly employed to benefit KG applications and mitigate the issues of data sparsity and incompleteness. More specifically:

1. Section 3.1: We design and validate **PRONTO**, a novel probing methodology for decoding latent hierarchical knowledge from frozen pretrained language models (PLMs). We demonstrate that by learning a "verbalizer" (i.e., a mapping function between a model prediction head and a classification label) over the model's entire output probability distribution, we can verifiably map its internal representations to the symbolic, ground-truth relationship of *semantic containment* (i.e., the hierarchical relationship that occurs between an entity/concept and a more general one). This investigation reveals a key insight: this knowledge is encoded by non-linear patterns in model prediction heads, as evidenced by the superior performance of non-linear probes over simple linear ones. Furthermore, we provide strong evidence that this decoded knowledge is fundamental and not a mere statistical artifact by showing it is highly sensitive to relational directionality and. Lastly, we assess the generalization capabilities of learned verbalizers, testing them in zero-shot entity typing.
2. Section 3.2: We introduce **TyleR**, a novel framework for inductive link prediction that successfully integrates the latent type knowledge of PLMs with graph-based structural reasoning. TyleR builds upon the insights from our first contribution by introducing a multi-faceted semantic enrichment

module to provide better node representations for graph-based inductive link prediction. This module extracts rich, type-aware features from a frozen PLM using a set of diverse prompts, allowing the model to operate without any explicit type annotations. We demonstrate the practical value of this "probe-and-apply" approach, showing how the PLM can act as a powerful semantic feature extractor to effectively fuel a GNN-based reasoning model. Our empirical results show that TyleR effectively compensates for data sparsity, outperforming strong baselines that rely on explicit (but often incomplete) type annotations, particularly in challenging scenarios where no explicit type data is present.

3.1 PRONTO: Prompt-Based Detection of Semantic Containment Patterns in MLMs ¹

Numerous studies have investigated the knowledge embedded within PLMs, demonstrating that these models effectively capture a significant amount of information from their pre-training content, encompassing both factual and ontological knowledge. Motivated by the work of Petroni et al. (Petroni et al., 2019), several works investigate factual (Youssef et al., 2023) and ontological (Wu et al., 2023) knowledge in PLMs by means of templated prompts. The idea is that given a prompt in the form of "Paris is a [MASK]", a PLM might predict "capital" with significantly higher probability than other possible tokens. Regardless of the methodology, many works concur on the idea that PLMs inherently possess some knowledge modeling capabilities that transcend mere textual co-occurrence patterns. Nevertheless, this inherent knowledge is rarely exploited in applicative scenarios as these models are often fine-tuned to achieve competitive levels of performance in downstream tasks.

In this work, we try to develop an understanding of how bidirectional PLMs might be inherently aware of a specific type of semantic relationship: **ontological containment**, i.e. *subclass* relation between ontological classes and *instance of* between an entity (instance) and an ontological class. Ontological containment refers to a hierarchical relationship between entities, where one is more general and encompasses another, indicating an "is a" relationship. This kind of knowledge is particularly interesting, as it is especially useful for various downstream tasks, such as entity typing and ontology completion. In particular, we are interested in whether a PLM can recognize a semantic containment relationship in the case where two entities are explicitly provided in the prompt (e.g. "Paris [MASK] city"). In other words, the overarching question of our investigation is: *are PLMs zero-shot semantic containment learners?*

Developing on several trends of prompt-learning and knowledge probing in PLMs, we propose PRONTO, a novel procedure aimed at the extraction of semantic containment relations from bidirectional PLMs based on the examination of their masked language modeling prediction head.

¹Based on our paper, "PRONTO: Prompt-Based Detection of Semantic Containment Patterns in MLMs", in *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part II*, pp. 227–246 (2024).

Our key contributions can be summarized as follows:

- We propose a general procedure to probe semantic containment knowledge from MLMs by means of automatically learned *verbalizers*, i.e. mappings between a MLM prediction head and a label.
- Through extensive analysis, we reveal how vanilla (i.e., not fine-tuned) MLMs exhibit an inner awareness with respect to semantic containment.

Our work significantly differs from prior literature. Prior prompt-based evaluation of taxonomical knowledge in PLMs formulates the task as object prediction (Petroni et al., 2019; Jain and Anke, 2022; Wu et al., 2023). In addition, Jain and Espinosa-Anke (Jain and Anke, 2022) analyze zero-shot taxonomy learning in PLMs considering joint probabilities of sentence tokens to predict a broader class for a given instance. Similarly, Huang et al. (Huang et al., 2022) frame entity typing as a fill-in-the-blanks task, learning a linear mapping between an MLM head and a fixed set of classes. We extend beyond the mapping to capture a general containment relationship that can be applied to an arbitrarily large set of classes. Specifically, we propose a methodology to train (linear and non-linear) verbalizers to detect whether an ontological containment relation (i.e., a different and broader task) holds between two given instances (i.e., a different prompt). Furthermore, differently from the work of Huang et al., we leverage vanilla PLMs. To the best of our knowledge, this is the first attempt to use the knowledge stored in PLMs to detect ontological containment through relation prediction with automatically extracted verbalizers. Finally, we highlight the significance of the findings by presenting practical applications in zero-shot entity typing.

The extraction of general semantic containment patterns from PLMs has significant potential to enhance downstream tasks at the intersection of text and structured knowledge, such as ontology completion and entity typing. The study may lead to more advanced methodologies for integrating text with structured ontological frameworks, advancing the field of knowledge representation.

3.1.1 Methodology

In this section, we formally introduce our **containment prediction** task that we schematize in Figure 3.1. Let $C = \{c_1, c_2, \dots, c_n\}$ represent the set of classes in a reference ontology O . Each class c_i is a node within the ontology graph. Let E_C be the set of edges representing the *subclass* relations among these classes,

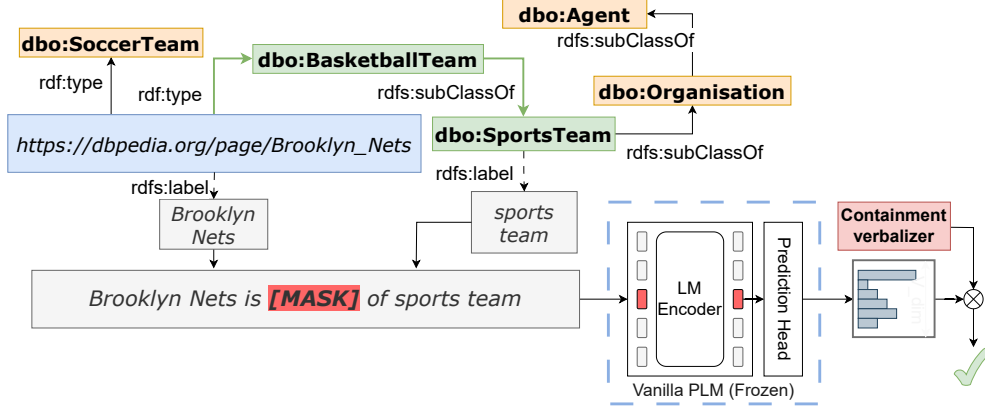


Figure 3.1: PRONTO Schematization: given a pair representing a containment in a reference taxonomy, representing a semantic containment relationship ("is-a"), PRONTO predicts the plausibility of the pair with a learned verbalizer.

where each edge $(c_i, c_j) \in E_C$ denotes that class c_i is a subclass of class c_j . Let $I = \{i_1, i_2, \dots, i_m\}$ denote the set of instances of classes in C , and E_I be the set of edges denoting the *instance of* relation, where each edge $(i_k, c_i) \in E_I$ indicates that instance i_k is of type c_i , linking instances to their respective classes. We define the **semantic containment graph** G as the union of the two sets of edges E_C and E_I , combined with their respective node sets C and I . Formally, $G = \langle C \cup I, E_C \cup E_I \rangle$. For any two nodes $v_i, v_j \in G$, we aim to determine whether there exists a path from v_i to v_j that signifies an “is-a” relationship within the ontology O . This relationship is characterized by a sequence of edges each representing either a direct subclass relation between classes or an instance belonging to a class, thereby forming a chain of semantic containment. Formally, we aim to learn a model $M_\theta : (v_i, v_j) \mapsto \hat{y}$ with:

$$\hat{y} = \begin{cases} 1 & \text{if there exists a path from } v_i \text{ to } v_j \text{ in } G, \\ 0 & \text{otherwise.} \end{cases}$$

The function M_θ is parameterized by the parameters θ derived from a vanilla PLM (e.g., BERT). The aim of M_θ is to learn the mapping $M_\theta : (v_i, v_j) \mapsto \hat{y}$, where $\hat{y}$ represents the predicted probability that a containment relationship exists between the concepts v_i and v_j .

Let us define a function P that constructs a prompt for a pre-trained MLM, given two nodes v_i and v_j . The function obtains the verbalized forms of v_i and v_j through $V(\cdot)$ and inserts a mask token [MASK] between them to form the prompt.

Formally, the prompt construction can be represented as

$$P(v_i, v_j) = V(v_i) \oplus \text{"[MASK]"} \oplus V(v_j), \quad (3.1)$$

where $V(v_i)$ and $V(v_j)$ are two natural language representations for the nodes v_i and v_j , respectively. The symbol $\oplus$ stands for string concatenation. Without loss of generality, in our implementation $V(v_j)$ is the `rdfs:label` associated to v_j .

Automatic Extraction of a Containment Verbalizer

Given the prompt $P(v_i, v_j)$ as input to a bidirectional PLM capable of mask-filling, the output of its MLM prediction head consists of the predicted probability distribution over possible tokens that could replace the [MASK] (Fig. 3.1). We propose investigating whether these predicted probabilities can help determine the existence of a containment relationship between v_i and v_j .

Given a PLM capable of mask-filling trained on a vocabulary of size V_{dim} and a prompt function $P(\cdot, \cdot)$, we aim to create a mapping between the prediction head output and a discrete label y . Prior work formulate the concept of verbalizer (Ding et al., 2022) as a discrete mapping between a subset of tokens $v_y = \{v_{y1}, \dots, v_{ym}\}$ and a label y . Formally:

$$p(y|x) = \frac{1}{m} \sum_{j=1}^m p(\text{[MASK]} = v_{yj}|x), \quad (3.2)$$

with m being the number of tokens in v_y and x being the prompt. The construction of v_y is often done manually: for instance, if $y = \text{"city"}$, a reasonable although simplistic verbalizer construction could be $v_y = \{\text{city}, \text{town}\}$. In this work, we formulate the construction of a verbalizer v_y as a search problem over the whole vocabulary. This enables our verbalizer to fully exploit the expressiveness of such a large vocabulary and possibly capture associations between labels and tokens that could be not easily identifiable even for domain experts. We want to design a verbalizer as a direct mapping function between the PLM prediction head and a label. A naive implementation of such a verbalizer is the following:

$$p(y|x) = \sum_{j=1}^{V_{dim}} \lambda_j p(\text{[MASK]} = v_j|x) = \sum_{j=1}^{V_{dim}} \sigma(\beta_j) p(\text{[MASK]} = v_j|x), \quad (3.3)$$

where $\lambda_j \in [0, 1]$ is a weighting factor that modulates the contribution of

each token v_j in the vocabulary to the probability of predicting y given x . The λ_j weights can be learned through an optimization process aiming to minimize a specified loss function. In fact, we learn the β_j parameters jointly in our optimization procedure, constraining them in a range $[0, 1]$ by means of a sigmoid. Ideally, we want the verbalizer to satisfy two useful properties:

P1 *Noise Resilience*: Since we are dealing with large vocabularies, the significant tokens’ marginal probabilities in a PLM prediction head tend to be diluted by the presence of many less relevant tokens. This dilution is linked to the softmax function’s property of distributing probabilities across all logits, diminishing the impact of pivotal tokens as the vocabulary size expands.

P2 *Sparsity*: We aim to enforce a sparsity constraint on the λ_j weights to promote interpretability. This constraint facilitates the identification of the most influential tokens minimizing the influence of less relevant ones. In fact, a PLM vocabulary is highly populated even for smaller models (30000+ tokens). Therefore, sparsity can aid interpretability for humans, which can only realistically focus on a smaller set of informative tokens simultaneously.

To satisfy **P1**, MLM prediction head logits pass through a weighted-softmax (Balazy et al., 2023):

$$\text{softmax}(x, w) = \left(\frac{w_1 \exp(x_1)}{\sum_{i=1}^n w_i \exp(x_i)}, \dots, \frac{w_n \exp(x_n)}{\sum_{i=1}^n w_i \exp(x_i)} \right), \quad (3.4)$$

where w are parameters learned jointly in the optimization process and constrained in the $[0, 1]$ range.

To satisfy **P2**, we impose an L1 regularization term over the learned weights λ_j in our loss function. L1 regularization is known to promote sparsity over other alternative regularization strategies, as well as improving generalization.

To investigate the potential benefits of non-linearity within our verbalization strategies, we draw inspiration from MAV (Mapping-Free Automatic Verbalizer) (Kho et al., 2023), in which the authors formulate a mapping-free verbalizer as a non-linear projection of a MLM prediction head in a latent vocabulary space. In our own adaptation, we substitute the inner *Tanh* activation function with *LayerNorm* for numerical stability. This modification is motivated by the observation that MLM logits can vary in unnormalized ranges, and the *Tanh* function

suppresses information associated with high activations:

$$p(y|x) = \sigma(W_2^T \cdot \text{Tanh}(W_1^T \cdot \text{LayerNorm}(\mathbf{logits}_{\text{MLM}}))). \quad (3.5)$$

In summary, we experiment with different verbalization strategies:

- **PRONTO-VF**: a *verbalizer-free* baseline approach, where the hidden state of the [MASK] token is fed into two fully connected layers with a final sigmoid activation, as in Equation (3.5);
- **PRONTO-LIN**: a naive linear *direct-mapping* approach, based on Eq. (3.3);
- **PRONTO-WS**: a *direct-mapping* approach where logits are re-weighted before the Softmax as in Equation (3.4), and the final label probability is obtained as in Equation 3.3;
- **PRONTO-MAV**: a *mapping-free* approach where logits fed into two fully connected layers as in Equation (3.5).

The direct-mapping verbalizers (PRONTO-WS, PRONTO-LIN) are inherently interpretable, since each λ_j can measure the contribution of the j -th token for the final label prediction. On the other side, PRONTO-MAV and PRONTO-VF can give an indication on more subtle patterns in the prediction heads that can only be acquired by means of non-linearities.

Data Preparation

Given a semantic containment graph $G = \langle C \cup I, E_C \cup E_I \rangle$, we denote Π^+ as the set of all the pairs of nodes (s, o) that can be found along a path of G . In other words, we compute the transitive closure of each node in G .

Since G does not contain negative information, this leaves an important decision: how to extract useful negative pairs. This decision is crucial since it impacts both the efficacy and generalizability of our learned verbalizers and the reliability of our evaluation. Intuitively, we want our model to be capable of distinguishing between semantically similar classes, although disjoint ones (e.g., "*city*"/"*region*"). However, we want it to be also able to distinguish among completely unrelated classes (e.g. "*city*"/"*person*"). Furthermore, we want it to correctly model a

Table 3.1: Visualization of the different hard/soft prompts used in this study. [s#] denotes a soft token.

Template Type	Template ID	Prompt Example
Hard Templates	h_1	Paris is [MASK] of capital
	h_2	Paris is a [MASK] of capital
	h_3	Paris [MASK] capital
	h_4	Paris is [MASK] capital
	h_5	Paris [MASK] of capital
Soft Templates	s_1	Paris is [s1][MASK][s2] of capital
	s_2	Paris [s1][MASK][s2] capital

semantic containment relationship that is non-commutative, instead of just discriminating based on word similarity. Based on these considerations, we devise three strategies to build the set Π^- of negative samples:

- **Reverse negatives:** given a positive pair (s, o) we obtain a negative pair by inverting subject and object (o, s) ;
- **Soft negatives:** given a positive pair (s, o) , we replace o with a random class sampled based on the class distribution in the data;
- **Hard negatives:** given a positive pair $(s_i, o_i) \in \Pi^+$, we build the two sets $\pi^+(s_i, o_i) = \{o_j \mid (s_i, o_j) \in \Pi^+\}$ and $\hat{\pi}^+(s_i, o_i) = \{\hat{o}_j \mid (\hat{o}_j, o_j) \in \Pi^+ \text{ and } \hat{o}_j \notin \pi^+(s_i, o_i) \text{ and } o_j \in \pi^+(s_i, o_i)\}$. While $\pi^+(s_i, o_i)$ represents the set of nodes along a path starting from s_i in the original graph G , namely all the nodes in a semantic containment relation with s_i , the set $\hat{\pi}^+(s_i, o_i)$ contains the nodes on the paths arriving in $\pi^+(s_i, o_i)$. These nodes are not in a semantic containment relation with s_i but are semantically "close" to it. Given a node s_i , the hard negatives are then built as $(s_i, \hat{o}_j)$ with $\hat{o}_j \in \hat{\pi}^+(s_i, o_i)$.

Table 3.3 illustrates some examples of negative pair construction, including a positive pair compared to a Hard, Soft, and Reverse negative pair, respectively.

Prompt Construction

Prior work has demonstrated the sensitivity of PLM outputs to prompt selection (Brown et al., 2020). In order to provide a more extensive analysis, we choose to experiment over different prompt templates. We report our prompt choices

Table 3.2: Summary of the statistics for the four constructed DBPedia dataset splits.

	Train	Val	Eval	Eval (Hard)
Total Pairs	141,090	372	36,824	27,286
Positive Pairs	18,597	182	18,416	18,416
Negative Pairs				
Total	122,493	190	18,408	8,870
Hard Negatives	59,632	96	8,870	8,870
Soft Negatives	30,556	40	4,702	0
Reverse Negatives	32,305	54	4,836	0

in Table 3.1. We design various hard templates to capture various linguistic manifestations of the containment relationship. Regardless of the prompt, both subject and object follow the same verbalization strategy, i.e., the `rdfs:label` literal value. In addition to manually designed prompts, we explore the integration of soft tokens (Qin and Eisner, 2021), i.e. word vectors jointly fine-tuned during the optimization process.

Training Procedure

To account for the significant number of constructed negative pairs in our dataset, we leverage α -balanced focal loss (Lin et al., 2017b) with label smoothing in order to reduce overfitting. We tune the α parameter based on the validation F1 score. All the verbalizers are tuned using Adam optimizer, selecting the optimal learning rate based on validation set performance. L1 regularization is employed for direct mapping verbalizers (as described in section 3.1.1), while PRONTO-MAV and PRONTO-VF are regularized with weight decay.

3.1.2 Experiments

This section outlines the experimental setup to probe the ability of PLMs to understand ontological containment relationships. We specifically focus on evaluating the inherent capacity of vanilla pre-trained MLMs prediction heads to recognize the hierarchical relation between instances and classes. The experiments are structured around three core research questions:

RQ1: *Do Masked Language Models (MLMs) capture semantic containment?*

Table 3.3: Examples of negative pair construction. The examples are generated from the DBPedia dataset following the strategy described in Section 3.1.1.

	Example Pairs #1	Example Pairs #2
Positive	Baltimore Ravens, american football Team	1,2,6-Hexanetriol, chemical compound
Hard Neg.	Baltimore Ravens, soccer club	1,2,6-Hexanetriol, monoclonal antibody
Soft Neg.	Baltimore Ravens, album	1,2,6-Hexanetriol, animal
Reverse	american football Team, Baltimore Ravens	chemical compound, 1,2,6-Hexanetriol

RQ2: *How does contextual information influence MLM performance in semantic containment prediction tasks?*

RQ3: *Can MLMs generalize their semantic containment reasoning abilities to new data and tasks?*

Dataset. We base our study on the dataset introduced by Wu et al. (Wu et al., 2023), a reputable dataset from recent literature on probing. This dataset is based on a restriction of DBPedia, containing 783 classes and up to 20 instances per class, with 8753 unique instances. The restriction is necessary because using the entire DBPedia is impractical due to resource limitations. Moreover, multi-hop link extraction scales exponentially as $(\text{entities} \times \text{branching_factor})^{\text{hops}}$. To extract positive and negative pairs, we follow the procedure described in section 3.1.1. We construct the set of negative pairs Π^- as follows: for each pair in Π^+ , we sample two hard, one soft and one reverse negative. We determine this to be a good ratio empirically through various tests on the validation set (F1-score showed a substantial improvement for the 2-1-1 ratio compared to 1-1-1, 1-2-1, and 1-3-1 for containment prediction; specific details were omitted for brevity). From the union of negative samples and positive samples $\Pi = \Pi^+ \cup \Pi^-$, we extract training and evaluation splits with holdout.

We find that the obtained evaluation split contains a significative amount of soft and reverse negatives, that could potentially inflate performances. For this reason, we extract a more challenging evaluation dataset, that we refer to as Eval (hard), removing all the soft and reverse negatives from the original evaluation split. We use the Eval (hard) dataset as evaluation dataset in all our experiments. The statistics of the obtained splits are reported in Table 3.2.

Probed PLMs. It is worth noticing that the proposed probing procedure is versatile and can be readily applied to any bidirectional PLM with mask-filling capabilities. For this investigation, we focus on two prominent encoder-only

PLMs,² BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), that leverage a masked language modeling objective during their pre-training stage. Extending this study to include unidirectional LMs requires additional considerations due to inherent architectural diversities. This would necessitate constructing “unnatural” prompts, such as “The relation between Paris and Capital is”. Moreover, access to prediction heads is not always available. Therefore, we choose to restrict this study to the two aforementioned models. For the chosen models, we employ the case-sensitive variants to ensure a consistent and fair comparison.

Hyperparameter Tuning We select optimal learning rate among {1e-1, 1e-2, 1e-5}, L1 coefficient among {1e-3, 1e-4} based on the validation F1 score. The training procedure is carried for a maximum number of epochs with early stopping if the validation F1 score does not improve after three consecutive epochs.

Semantic Containment Understanding in PLMs (RQ1)

To evaluate the effectiveness of the probed PLMs in identifying semantic containment relationships, we analyze the performance of various combinations of verbalization strategies, templates, and PLMs (the interested reader may take a look to Section 3.1.1 for further details). We report the results in Table 3.4, presenting accuracy, precision, recall, and F1-score for each combination. A decision threshold of 0.5 was used for all models.

PLM Comparison. The analysis reveals several interesting trends. The first finding is that the *verbalization strategy matters*. The Mapping-Free Automatic Verbalizer (MAV) consistently outperforms those based on direct mapping (LIN and WS). This suggests token probabilities likely contain complex relationships that direct mapping approaches might miss. The MAV strategy seems to capture these more effectively. The RoBERTa-Large model generally achieves better and more consistent results, particularly with direct-mapping verbalizations (LIN and WS). For the MAV verbalizer, RoBERTa-Base outperforms the larger model with specific template choices (h_4, h_2, s_1, and s_2). This suggests that prompt design plays a crucial role in performance, even for larger models.

There is *no clear correlation between model size and the PLMs’ discriminative ability* to distinguish containment relationships. MAV verbalizers show similar

²For all the adopted PLMs, we employ the pre-trained checkpoints available at <https://huggingface.co/>.

Table 3.4: Results over all combinations of Verbalizer-Template and different PLMs on the DBpedia evaluation (hard) dataset. All negative samples in this evaluation split are hard negatives, constructed as detailed in section 3.1.1. "Acc", "P", "R" and "F1" denote, respectively, accuracy, precision, recall and F1-score. In **bold**, we report the best results for each column.

	TID	RoBERTa-L				RoBERTa-B				BERT-B			
		Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
PRONTO-VF	h_1	80.22	95.28	74.37	83.54	75.99	95.57	67.55	79.15	77.09	94.73	69.94	80.47
	h_2	79.98	95.10	74.16	83.34	79.23	96.28	72.00	82.39	77.93	95.69	70.48	81.17
	h_3	78.48	95.36	71.61	81.79	74.47	95.60	65.18	77.51	77.75	95.43	70.40	81.03
	h_4	80.63	96.40	74.06	83.77	78.32	96.32	70.58	82.46	78.16	95.46	71.01	81.44
	h_5	75.94	95.37	67.64	79.14	74.34	95.47	65.07	77.39	76.91	94.02	69.51	80.25
	s_1	76.01	94.54	68.40	79.37	80.07	95.11	74.30	83.42	75.76	95.07	67.58	79.00
	s_2	80.99	95.59	75.30	84.24	80.51	96.14	74.10	83.69	80.38	95.20	74.69	83.71
PRONTO-LIN	h_1	56.32	88.78	40.38	55.51	52.71	89.16	34.08	49.31	46.37	89.04	23.42	37.09
	h_2	60.00	84.05	50.27	62.91	58.36	87.68	44.56	59.09	58.90	84.71	47.72	61.05
	h_3	41.99	81.69	18.10	29.64	46.04	80.22	26.62	39.97	46.98	75.99	31.36	44.39
	h_4	52.48	85.32	35.75	50.39	51.66	86.74	33.49	48.32	42.34	82.60	18.46	30.17
	h_5	49.09	89.34	27.90	42.53	51.45	91.00	31.15	46.41	51.83	89.54	32.41	47.59
	s_1	83.47	84.82	79.88	86.71	79.38	96.97	71.69	82.44	82.81	95.08	78.60	86.06
	s_2	80.42	92.93	76.83	84.12	79.57	94.98	73.63	82.95	78.75	94.59	72.67	82.19
PRONTO-WS	h_1	76.22	91.58	71.32	80.19	75.96	90.40	72.02	80.17	70.98	89.63	64.46	74.99
	h_2	73.91	91.56	67.58	77.76	70.47	93.89	60.16	73.33	72.40	90.25	66.27	76.42
	h_3	70.30	89.89	63.08	74.14	67.49	86.13	61.78	71.95	70.50	88.31	64.88	74.80
	h_4	76.05	90.57	72.01	80.23	68.96	87.59	62.92	73.23	71.24	87.46	66.99	75.87
	h_5	72.59	91.95	65.08	76.22	70.44	89.46	63.71	74.42	67.37	89.95	58.15	70.64
	s_1	80.59	93.48	76.57	84.19	81.04	94.21	76.63	84.51	80.89	91.91	78.61	84.74
	s_2	84.60	95.09	81.38	87.70	82.50	95.37	77.84	85.72	83.81	92.61	82.61	87.33
PRONTO-MAV	h_1	84.51	94.50	81.80	87.70	81.01	96.41	74.64	84.14	83.82	96.79	78.64	86.78
	h_2	81.67	93.97	77.84	85.15	82.77	96.97	76.87	85.76	84.32	96.83	79.37	87.24
	h_3	84.00	95.65	79.92	87.08	79.70	93.72	74.94	83.28	83.54	96.09	78.82	86.60
	h_4	84.38	96.72	79.55	87.30	85.06	95.63	81.60	88.06	84.71	96.98	79.83	87.57
	h_5	82.31	95.66	77.30	85.51	81.32	96.08	75.40	84.49	83.06	92.16	81.87	86.71
	s_1	79.20	95.51	72.59	82.49	83.09	95.84	78.35	86.22	83.42	95.61	79.05	86.55
	s_2	82.59	95.00	78.33	85.87	84.01	95.71	79.89	87.09	83.94	96.78	78.83	86.89

Table 3.5: GPT-3.5 turbo results (zero-shot) on the Eval (hard) dataset.

	Acc	P	R	F1
GPT-3.5 turbo (zero-shot)	60.18	96.36	42.61	59.10

performance across model sizes while direct-mapping variants tend to improve with larger models. We hypothesize that smaller PLMs may exhibit more nuanced activation patterns for containment, requiring a non-linear verbalizer like MAV to capture them. This result is in line with previous works that reached conflicting conclusions on this matter: indeed, Petroni et al. (Petroni et al., 2019) showed overall better results for larger PLMs in ontological memorization capabilities, while a more recent study (Wu et al., 2023) proved that model size does not have reasonable impact on stored ontological knowledge.

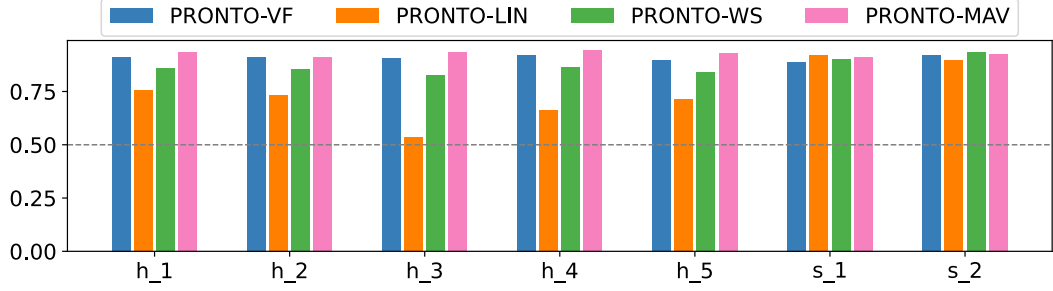


Figure 3.2: Comparison of AUC scores for different verbalization strategies on the DBPedia Eval (hard) dataset. The reference model is RoBERTa-Large.

The analysis suggests that *vocabulary size might not be the primary factor influencing performance in this task*. Interestingly, BERT-Base, with a smaller vocabulary compared to RoBERTa-Base (approximately 20,000 fewer tokens), outperforms RoBERTa-Base for PRONTO-WS and PRONTO-MAV verbalizations across most prompts. This indicates that other factors, potentially the specific tokenization strategies or the training data used for each model, may play a more significant role in capturing semantic relationships.

Figure 3.2 shows the Area Under the ROC Curve (AUC) scores for all verbalizer-prompt combinations using the RoBERTa-Large PLM. These scores reflect the model’s ability to distinguish between positive and negative containment pairs. While overall performance varies with prompt choice for the same verbalizer, the results indicate some general trends. While PRONTO-LIN achieves the lowest accuracy and F1 scores for hard prompts, it exhibits good AUC scores, particularly for the h_1 and h_2 prompts. This suggests that PRONTO-LIN might benefit from optimizing the decision threshold used to classify positive and negative pairs. A potential explanation is in its underlying architecture. Indeed, PRONTO-LIN computes the label probability as a linear sum of individual token probabilities. These token probabilities can be noisy and potentially influenced by irrelevant factors, especially as vocabulary size increases. However, *adjusting the decision threshold could help mitigate the impact of this noise and potentially improve PRONTO-LIN’s performance*.

Table 3.5 presents the performance of GPT-3.5 turbo on the same containment prediction task. Here, the containment hypothesis is phrased in natural language as " $V(v_i)$ is kind of $V(v_j)$ " and GPT-3.5 is asked to judge its truth value (true or false). This evaluation is conducted in a zero-shot scenario, meaning GPT-3.5 receives no task-specific training. Therefore, the results are not directly

Table 3.6: Performance comparison with the reverse negatives-only dataset. All results are in terms of AUC score. The reference model is RoBERTa-Large.

Verbalizer	TID	DBPedia Eval (Hard)	DBPedia Eval (Reverse Neg.)
PRONTO-WS	h_1	85.83	97.26
	h_2	85.39	97.79
	h_3	82.54	97.45
	h_4	86.19	98.03
	h_5	84.29	96.53
	s_1	90.12	99.08
	s_2	93.27	99.35

comparable to the performance of the other verbalizer models discussed earlier. Despite the limitations, this zero-shot evaluation provides a valuable insight into the inherent difficulty of the Eval (hard) dataset. The performance of GPT-3.5 on this task can serve as a baseline for comparison with the other models and potentially indicate the complexity of the overall reasoning required for the task.

Sensitivity to the Relative Positioning of Instances and Classes in the Unstructured Text. To gauge whether the models’ predictions extend beyond simply memorizing word co-occurrences, we analyze their performance on a specifically-designed evaluation set. This set consists solely of positive and "reverse negative" examples (see Section 3.1.1 for a detailed definition), where the less specific concept appears on the right-hand side of the prompt (opposite to the positive examples). We compare these results to those obtained on the Eval (hard) set. The AUC scores for both sets are presented in Table 3.6.

The rationale behind this experiment is to assess whether the verbalizers can understand the underlying meaning of containment relationships beyond just memorizing word co-occurrences. If the verbalizers perform well on "reverse negative" evaluation set, it suggests a deeper understanding of containment since reverse negatives flip the concept order compared to positive examples, making memorization less effective.

The results show that the verbalizer performs significantly better on the reverse negative set. This suggests *the verbalizer can indeed distinguish between the relative specificities of concepts within the prompts, even when the relationship is flipped*. This finding supports the notion that the containment relationship depends on the relative positioning of the concepts and models appear to be sensitive to the

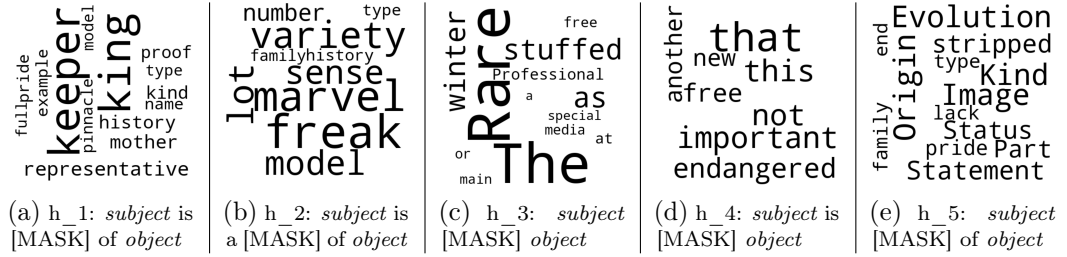


Figure 3.3: Visualization of learned verbalizers: the wordclouds report the top weighted tokens for the "is-a" relationship in hard templates. The reference model is RoBERTa-Large under the PRONTO-WS setting.

order in which concepts are presented. The higher AUC scores for reverse negative examples compared to hard negatives indicate that the PLMs are generally adept at identifying specific concepts from more general ones.

Insights from Learned Verbalizer Weights. Figure 3.3 provides valuable insights into how the PRONTO-WS verbalizer discriminates between positive and negative containment pairs, showing the most important tokens according to the associated verbalizer weights. While some top tokens like "variety", "type", and "kind" are easily interpretable, many others are less intuitive. This suggests that the *model captures nuanced patterns in the data that might not be readily apparent to humans*.

In addition, we found the presence of some counter-intuitive tokens (e.g. "not") that would suggest negating the relation. This is likely due to the model trying to compensate for discrepancies between the MLM prediction and the ground truth—a degree of statistical "disagreement" (i.e. the LM wrongly predicts the "not" token for certain relationships where it lacks background knowledge). This could also be motivated by the presence of specific token bias, although this necessitates further investigation.

These findings support the notion that *containment relationships are more intricate than what can be fully expressed through manually designed verbalizers*. The model appears to leverage a broader range of cues within the vocabulary. Overall, this analysis highlights the necessity of exploring the full vocabulary to develop more effective verbalizers benefitting from a wider range of interactions.

Enhancing Verbalizers with Knowledge Graph Descriptions: The Impact of Context (RQ2)

Building upon the learned verbalizers, we investigate the feasibility of leveraging textual descriptions from our knowledge graph (KG) to potentially improve their performance in addressing **RQ2**. This exploration is rooted in the hypothesis that enhancing our prompts with relevant context about the entities involved can reinforce the model’s understanding of the underlying semantic relationships and lead to better discrimination between positive and negative containment pairs.

To address RQ2, we reformulate the original containment prediction task as a textual entailment task (García-Silva et al., 2023). Here, we aim to infer whether a hypothesis $H(v_i, v_j)$ holds true based on a provided natural language premise $P(v_i, v_j)$. The hypothesis is formulated using the same prompt construction method detailed in Section 3.1.1. For the premise, we leverage the textual descriptions associated with entities v_i and v_j from the KG. We construct the premise by concatenating the textual descriptions for entities v_i and v_j . Specifically, we use the `dbo:abstract` property for the instances (v_i) and `rdfs:comment` property for the classes (v_j) from DBPedia’s Eval (hard) dataset, if available. Since PLMs have a maximum window size, we further process the dataset by removing textual descriptions exceeding 50 tokens in length.

Table 3.7 presents the final results on the Eval (hard) dataset after incorporating textual descriptions from the knowledge graph (KG). The results reveal an interesting trend. Contrary to expectations, adding context generally leads to a decrease in performance across most verbalizer-prompt combinations. This suggests that the KG descriptions might be introducing noise rather than providing beneficial information. This negative impact can be attributed to architectural factors. The prediction heads of the PLMs used may be sensitive to variations in input data, struggling to integrate the additional context effectively. Moreover, the verbalizers themselves might be susceptible to changes in the input, hindering their ability to leverage the supplementary information. Interestingly, direct-mapping verbalizers (like PRONTO-LIN) are less affected by the inclusion of context, showing improvements for specific prompts (h_3, h_4, h_5).

This experiment highlights the need for further investigation into effective strategies for incorporating contextual information from knowledge graphs.

Table 3.7: Results over all combinations of Verbalizer-Template on the DBPedia Eval (hard) dataset, in the "with context" setting. The reference model is RoBERTa-Large. "Acc", "P", "R" and "F1" denote, respectively, accuracy, precision, recall and F1-score. ↓ and ↑ denote, respectively, a decrease or an improvement in F1-score with respect to the results in absence of context (Tab. 3.4). In **bold**, we report the improved F1-scores.

DBPedia (Hard) w. context																
TID	PRONTO-VF				PRONTO-LIN				PRONTO-WS				PRONTO-MAV			
	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
h_1	70.25	89.30	63.53	74.24 ↓	53.58	79.37	42.20	55.10 ≈	65.18	84.67	59.13	69.63 ↓	72.35	88.83	67.52	76.72 ↓
h_2	76.20	90.80	72.05	80.34 ↓	57.12	78.56	50.17	61.23 ↓	68.93	86.34	64.10	73.58 ↓	76.21	89.91	72.94	80.54 ↓
h_3	77.23	92.27	72.31	81.08 ≈	40.87	74.55	18.80	30.03 ↑	67.38	85.00	62.74	72.20 ↓	81.07	93.23	77.59	84.69 ↓
h_4	71.40	93.20	62.17	74.58 ↓	51.84	81.34	37.16	51.02 ↑	65.37	86.15	58.02	69.34 ↓	75.99	94.20	68.65	79.42 ↓
h_5	77.30	88.12	76.71	82.02 ↑	56.17	83.73	43.53	57.28 ↑	72.49	83.78	73.46	78.28 ↑	79.96	89.14	80.06	84.35 ↓
s_1	68.99	90.40	60.47	72.47 ↓	73.99	87.52	71.68	78.81 ↓	69.05	86.74	63.92	73.60 ↓	71.75	91.46	64.14	75.40 ↓
s_2	78.40	92.25	74.23	82.27 ↓	58.88	91.21	43.24	58.67 ↓	76.23	87.66	75.40	81.07 ↓	78.91	91.32	75.98	82.95 ↓

Generalizability of Verbalizers (RQ3)

Generalizability to Unseen Instances. To address **RQ3**, we investigate the impact of training data size on the generalizability of the verbalizers to unseen entities. This mimics a real-world scenario of ontology completion, where we want a model to leverage its knowledge to predict relationships for new entities. We transform the containment prediction task into an inductive setting. Here, the model predicts containment relationships for potentially unseen entities not present in the training data. We process our training dataset as follows: we randomly select 80 % of the instances in the Eval (hard) dataset and we remove all the pairs in the training set with any of these entities. This operation results in a total of 20690 test pairs ($\approx 75\%$ of the total Eval (hard) dataset) with an unseen instance. The resulting (reduced) training set is composed by a total of 38742 pairs ($\approx 27\%$ of the original size). We retrain our verbalizers based on this reduced training split and report the results on the Eval (hard) dataset in Table 3.8. Overall, reducing training size has a negative impact on performance across all metrics, although the decrease is not substantial. Interestingly, this trend does not hold for certain verbalizer and prompt combinations. For example, PRONTO-MAV outperforms its counterpart that was trained on the full dataset in both F1 score and accuracy when using the h_4 prompt template. Despite this, performance remains comparable with the other verbalizer-prompt combinations, demonstrating robust generalization.

Table 3.8: Results over all combinations of Verbalizer-Template on the DBPedia Eval (hard) dataset, when **80% of instances present in the evaluation set have zero occurrences in the training set**. The reference model is RoBERTa-Large. "Acc", "P", "R" and "F1" denote, respectively, accuracy, precision, recall and F1-score. In **bold**, we report the best results over all verbalizers.

DBPedia (Hard) w. 80% unseen instances																
TID	PRONTO-VF				PRONTO-LIN				PRONTO-WS				PRONTO-MAV			
	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
h_1	78.61	93.99	72.98	82.16	55.77	89.91	38.82	54.23	75.07	90.54	70.42	79.22	81.47	93.60	77.87	85.01
h_2	79.09	93.20	74.46	82.78	58.59	85.96	46.19	60.09	74.69	89.80	70.50	78.99	81.97	94.17	78.12	85.40
h_3	75.73	94.99	67.60	78.99	41.32	71.16	21.95	33.55	70.18	90.07	62.73	73.95	79.72	96.04	72.96	82.92
h_4	81.48	95.60	76.07	84.72	52.48	88.54	33.99	49.12	72.56	92.40	64.67	76.08	84.69	89.20	87.97	88.58
h_5	75.47	94.75	67.40	78.77	50.53	89.89	30.08	45.08	75.02	92.13	68.88	78.82	80.04	92.41	76.73	83.85
s_1	75.30	92.88	68.66	78.96	58.01	86.82	44.54	58.88	81.83	92.66	79.37	85.50	79.85	92.93	75.93	83.58
s_2	78.62	94.50	72.55	82.08	77.58	92.48	72.69	81.40	80.75	94.16	76.22	84.24	80.74	94.37	75.99	84.19

Zero-shot Entity Typing. To validate the generalizability of the developed verbalizers, we investigate an entity typing task within a zero-shot setting. The aim is to assess the type of an entity based on its surrounding context. Given a tokenized text sequence $S = (t_1, t_2, \dots, m_i, \dots, m_j, \dots, t_T)$, where $m = (m_i, \dots, m_j)$ represents an entity mention, the aim is to assign an entity type label y from a predefined set of types Y to the entity mention m . We can reformulate this task as a textual entailment task: given an premise $P = S$, the goal is to predict whether an hypothesis $H = P(m, y)$ holds, where $P(m, y)$ is a cloze prompt constructed as in eq. 3.1. We formulate a separate entailment prompt for each of the types in Y , selecting the one that achieves the highest probability as the final entity type prediction. We selected the Few-NERD dataset for our experiments (Ding et al., 2021), a high-quality, manually annotated NER dataset featuring both fine-grained and coarse-grained tags. This dataset poses challenges such as varied types that lack clear textual representations and potential overlaps among types. For example, the fine-grained type "Living Thing" may overlap with the fine-grained types within the "Person" coarse type, hence it is not directly exploitable in a zero-shot setting. Consequently, we focus our experiments on the fine-grained splits corresponding to the well-defined and disjoint coarse categories of Person (7 types), Location (6 types), and Organization (9 types). Additionally, we exclude all types labeled as MISC or "Other" due to their ambiguous nature. In our entity typing experiments, we utilize PRONTO-MAV, the model that achieved the highest overall performance in our prior tasks. In this case, no training is conducted, as we directly employ the same verbalizer obtained from

Table 3.9: Entity Typing results (zero-shot) on Few-NERD. The reference model is RoBERTa-Large.

	Few-NERD (COARSE)		Few-NERD (PER)		Few-NERD (ORG)		Few-NERD (LOC)	
	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1
PRONTO-MAV (zero-shot)	50.11	55.91	61.44	68.05	64.06	69.28	49.59	70.67

our containment prediction task detailed in section 3.1.2. Table 3.9 reports the results on Few-NERD (Coarse) and three fine-grained splits (Person, Location, Organization). Despite the considerable challenges posed by the dataset, requiring type inference from sentences where entities are often mentioned without sufficient context, we achieved remarkable results in terms of micro F1 score on the fine-grained splits. On the coarse-grained dataset, the results are lower, due to the presence of semantically similar and possibly overlapping types (e.g. Location and Building, Product and Art).

3.1.3 Limitations

Our analysis is inherently constrained by the ontological knowledge encoded in pretrained language models (PLMs), acquired to their exposition to large corpora. Although these models exhibit a remarkable ability to acquire world knowledge through self-supervised learning, the depth and accuracy of this knowledge are limited to the content available in their pretraining corpora and their ability to store this knowledge. Consequently, our insights into the extent of ontological knowledge within PLMs and their capacity to reason with this knowledge are confined to what is already present in these models.

Moreover, like many large language models trained on internet data, PLMs may also reflect societal biases and toxicity found in their training materials. Our ontological probing experiments do not directly address these issues, which could negatively affect the behavior of these models in real-world settings. Identifying and developing strategies to counteract these biases and toxicities is a crucial area for future research.

3.1.4 Summary

This study investigated the ability of pre-trained Masked Language Models (MLMs) to understand hierarchical semantic relationships. The findings suggest that MLMs

exhibit some grasp of ontological containment, as evidenced by consistent patterns in the prediction heads. This knowledge about containment holds significant potential for tasks where labeled data is scarce, such as ontology completion and entity typing. We explored the generalizability of this approach, including learning specific verbalizers, inductive containment prediction, and zero-shot entity typing. While non-linear verbalizers showed remarkable performance, there is room for further exploration on developing more advanced verbalization strategies to better integrate textual information with structured ontological frameworks. Additionally, research on improving the relevance of descriptions used for context integration or exploring alternative context integration methods could be fruitful avenues for further advancement. Our codebase is available on GitHub ³.

³<https://github.com/sisinflab/PRONTO>

3.2 Type-Less yet Type-Aware Inductive Link Prediction with Pretrained Language Models ⁴

Knowledge graphs (KGs) represent complex relationships between entities in a structured, graph-based format (Hogan et al., 2021). Their ability to encode semantic information and support reasoning makes them valuable in a variety of applications, such as natural language processing (Peters et al., 2019), recommendation systems (Wang et al., 2024b), and biomedical research (Gema et al., 2023). However, KGs are notoriously incomplete: many valid relations are absent, reducing their effectiveness in downstream tasks (Rossi et al., 2021).

Link prediction aims to infer these missing relationships by analyzing the existing graph’s structure and patterns. Traditional link prediction methods aim to predict links among entities observed during training. Although effective in static settings, they are limited in dynamic environments where new entities are incrementally introduced. Inductive link prediction (ILP) addresses this challenge by aiming to generalize to previously unseen entities, leveraging transferable features such as structural information and type information.

Prior work has demonstrated that incorporating entity type information can enhance generalization capabilities of ILP models. For instance, Zhou et al. (2023) explicitly integrate type annotations and ontological constraints into the learning process. Yet, these methods face a critical bottleneck: available explicit type information in real-world KGs is often coarse-grained, incomplete, or even erroneous. This limitation is particularly acute when facing structural sparsity. Consider, for instance, the triple $\langle \text{Lionel Messi}, \text{playedFor}, \text{Barcelona FC} \rangle$. A model might assign similar plausibility to $\langle \text{Cristiano Ronaldo}, \text{playedFor}, \text{Barcelona FC} \rangle$ if both subject entities (i.e., Lionel Messi and Cristiano Ronaldo) lack distinct neighborhood information and are categorized only under the broad type "Footballer". This highlights a fundamental inadequacy of type-informed ILP approaches when explicit type signals are weak and local graph structure is uninformative.

To address this gap, our idea is to leverage the rich semantic knowledge captured by pre-trained language models (PLMs). We hypothesize that the

⁴Based on our paper, “Type-Less yet Type-Aware Inductive Link Prediction with Pretrained Language Models”, in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 27181–27197 (2025).

semantic understanding these models acquire during their extensive pre-training on vast textual corpora (Petroni et al., 2019; Hao et al., 2023) offers a pathway to a more fine-grained representation of entities. This "inner knowledge," encoded within the PLM's parameters, offers a dense representation of diverse semantic facets. For example, prompting a PLM like BERT (Devlin et al., 2019) with "Paris is located in __," generates a hidden representation for the missing token that (ideally) enables it to correctly predict "France," reflecting the model's "*understanding*" of the Paris's geographical location. We aim to utilize the implicit semantic insights of PLMs to derive fine-grained entity representations, overcoming limitations in explicit type information. We start from these two observations: (i) an entity can be described by a set of assertions defining its properties; (ii) the same assertions, when used as prompts for a PLM, can elicit dense, multifaceted representations that implicitly capture a "type-aware" understanding of the entity. This potential led us to ask: *Can PLM-derived entity representations compensate for structural and type sparsity in inductive knowledge graph completion?* To investigate this question, we introduce **TyleR**—*Type-less yet type-aware*—a novel inductive link prediction framework that leverages PLMs to embed implicit type-aware signals within node representations, thus eliminating reliance on explicit type annotations. Our contributions are:

1. We introduce a novel methodology for harnessing PLMs to derive and embed implicit type semantics within an ILP model, thereby enabling nuanced entity representations without relying on explicit type data.
2. We demonstrate TyleR's effectiveness on multiple benchmark datasets, showing its capability to perform competitively, especially in settings with limited or coarse-grained type information and sparse graph structures.
3. We conduct an empirical analysis investigating the interplay between PLM-derived semantic features and varying levels of type and structural sparsity, thereby characterizing the resilience of our approach.

3.2.1 Background and Motivation

Inductive link prediction aims to predict the likelihood of triples (h, r, t) , where h and t are unseen entities. In practice, this is done by means of a scoring function $f(h, r, t)$. At training time, f is optimized on the triples in a training graph $\mathcal{G}_{train}$. At test time, the same scoring function is used to predict the plausibility of triples

(h', r, t') belonging to a test graph $\mathcal{G}_{test}$, based on the triples in an inference graph $\mathcal{G}_{inf}$. Unlike traditional embedding-based approaches, subgraph-based relation prediction methods such as GraIL (Teru et al., 2020) can be viewed as learning logical rules that capture entity-independent relational semantics. For example, one can derive the simple rule:

$$spouse_of(X, Y) \wedge lives_in(Y, Z) \rightarrow lives_in(X, Z).$$

As demonstrated by Zhou et al. (2023), the reasoning capabilities of GraIL can be enhanced by incorporating explicit *type information* about entities. This additional semantic context enables the model to induce more precise and type-aware rules:

$$Employee(X) \wedge Department(Y) \wedge Office(Z) \wedge \\ \wedge part_of(X, Y) \wedge located_in(Y, Z) \rightarrow works_in(X, Z).$$

Type-constrained rules enhance both accuracy and interpretability in relational inference by reducing spurious predictions and enforcing semantic validity. However, explicit type information is often incomplete or missing in real-world knowledge graphs. To address this, we propose learning a function τ_{PLM} , parameterized by a pre-trained language model, that maps entities to implicit type representations capturing their latent semantics. These PLM-derived embeddings enable type-aware reasoning without explicit type labels and can be integrated into the logical rule induction process. For example, a type-aware rule may take the form:

$$\tau_{PLM}(X) \wedge \tau_{PLM}(Y) \wedge \tau_{PLM}(Z) \wedge part_of(X, Y) \wedge \\ \wedge located_in(Y, Z) \rightarrow works_in(X, Z),$$

with $\tau_{PLM}(X)$, $\tau_{PLM}(Y)$, and $\tau_{PLM}(Z)$ such that

$$\begin{aligned} \tau_{PLM}(X) &\approx Employee(X), \\ \tau_{PLM}(Y) &\approx Department(Y), \\ \tau_{PLM}(Z) &\approx Office(Z), \end{aligned}$$

where $\tau_{PLM}(\cdot)$ for X is an approximation of the logical statement $Employee(\cdot)$ while, for Y and Z , $\tau_{PLM}(\cdot)$ is an approximation of their types, $Department$ and

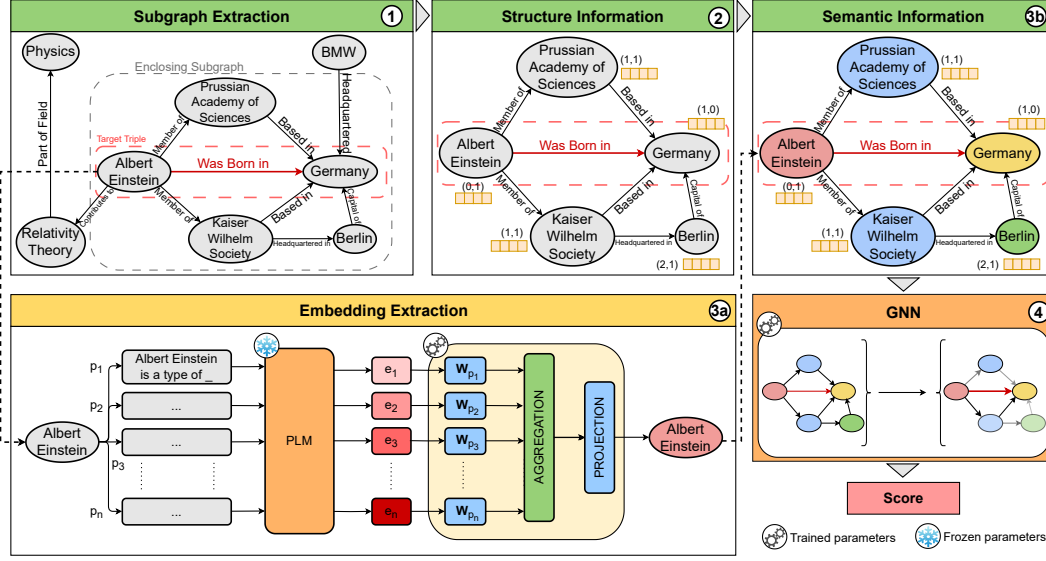


Figure 3.4: Overview of TyleR. The process begins with ① extracting the enclosing subgraph and ② applying a node labeling strategy. Multi-faceted, semantic representations are then derived using a pre-trained language model ③a, ③b. Finally, a graph neural network ④ integrates structural and semantic information to obtain the final prediction.

Office, respectively (more details in Section 3.2.2). This guides the rule induction process towards more meaningful and generalizable patterns, allowing us to infuse latent type semantics into subgraph-based link prediction models, even when explicit type information is absent.

3.2.2 Methodology

In this section, we introduce **TyleR** (**T**ype-less yet type-awa**R**e inductive link prediction with pre-trained language models). Building on Graph Inductive Learning (Teru et al., 2020), which infers relations from local subgraph patterns, TyleR leverages PLM-derived semantics to enrich node representations. However, integrating PLMs into full-graph models is computationally expensive due to high-dimensional embeddings and large graph size. TyleR adopts a subgraph-reasoning approach, restricting triple scoring to compact and informative subgraphs, making PLM integration tractable.

As illustrated in Figure 3.4, TyleR’s pipeline consists of four stages: ① extracting an enclosing subgraph, ② structurally labeling nodes (following GraIL (Teru

et al., 2020)), ③a, ③b enriching nodes with PLM-based semantic embeddings, and ④ feeding the enhanced subgraph into a GNN architecture from Zhou et al. (2023). The following sections provide further details on each step.

Subgraph Extraction ①. Given a target triple (u, r_t, v) , we define $\mathcal{N}_k(u)$ and $\mathcal{N}_k(v)$ as the sets of k -hop neighboring nodes of u and v , respectively. We also define a specific distance metric $d(i, u)$ as the shortest path from a node i to u that does not pass through v , and $d(i, v)$ is similarly the shortest path distance from i to v that does not pass through u . The *enclosing subgraph* of triple (u, r_t, v) is computed by (i) forming an initial set of candidate nodes by taking the intersection $\mathcal{N}_k(u) \cap \mathcal{N}_k(v)$ and (ii) pruning nodes that are either isolated (i.e., have no edges connecting it to other nodes within the subgraph after this pruning step) or for which $d(i, u) > k$ or $d(i, v) > k$. The remaining nodes and their edges form the enclosing subgraph.

Subgraph Labeling ②. Each node i in the extracted subgraph is labeled with a pair of shortest path distances $(d(i, u), d(i, v))$ to the target nodes u and v , respectively, within the subgraph. This pair captures the relative *position* of node i with respect to the target nodes u and v . The final positional embedding $\mathbf{h}_i^{\text{pos}}$ is:

$$\mathbf{h}_i^{\text{pos}} = \text{one-hot}(d(i, u)) \oplus \text{one-hot}(d(i, v)), \quad (3.6)$$

where $\oplus$ denotes the concatenation operator and $\text{one-hot}(\cdot)$ is the one-hot encoding function. All nodes in the enclosing subgraph are within k hops of u or v , so $\mathbf{h}_i^{\text{pos}} \in \mathbb{R}^{2k+2}$.

Semantic Enrichment ③a③b. Semantic enrichment leverages a pre-trained language model (PLM), supporting either masked token prediction or next-token generation. A straightforward approach for encoding entity type semantics involves prompting the PLM with an explicit query to elicit the most plausible type for a given entity. For masked language models (MLMs) (e.g., RoBERTa), this operation results in a prompt such as “*The type of Paris is [MASK].*”, with the type semantics encoded in the last hidden layer representation of the [MASK] token; for causal language models (CLMs) (e.g., Llama), this representation corresponds to the last hidden representation of the final sequence token. However, relying solely on representations derived from such direct type queries can be

suboptimal. Prior research has shown that transformer-based representations tend to be highly anisotropic, often concentrated in narrow cones (Ethayarajh, 2019), which can limit their discriminative utility. To address this, we propose to refine type semantics through multiple prompts, designed to extract different semantic aspects. Given an entity i with textual label l_i , we define a set of **assertion prompts** $P = \{p_1, p_2, \dots, p_n\}$, where each p_k targets a semantic facet of the entity (e.g., type, location, membership). Each prompt $p_k(l_i)$ is processed by the PLM (3a) to yield a latent representation:

$$\mathbf{z}_{p_k, i} = \text{Extract}(\text{PLM}(p_k(l_i))). \quad (3.7)$$

Here, $\text{PLM}(\cdot)$ denotes the forward pass of the language model given an input prompt, and $\text{Extract}(\cdot)$ selects the relevant hidden state. More specifically, for MLMs like RoBERTa, if $p_k(l_i)$ is formulated as, for example, “*The type of Paris is [MASK].*”, then $\mathbf{z}_{p_k, i}$ corresponds to the final hidden layer representation of the [MASK] token. For CLMs like Llama, given a prompt such as “*The type of Paris is*”, $\mathbf{z}_{p_k, i}$ is taken as the final hidden layer representation of the last token in the input sequence. These representations $\mathbf{z}_{p_k, i}$ are refined and projected into a unified space using an assertion-specific projection block:

$$\mathbf{z}_{p_k, i}^h = \mathbf{W}_{p_k} \text{LN}(\mathbf{z}_{p_k, i}) + \mathbf{b}_{p_k}, \quad (3.8)$$

where $\mathbf{W}_{p_k}$ and $\mathbf{b}_{p_k}$ are specific learnable parameters for each assertion prompt p_k , and LN denotes layer normalization (Ba et al., 2016). We aggregate ($\text{AGG}_p(\cdot)$) the prompt representations with different strategies such as *sum*, *mean* or *concatenation*:

$$\mathbf{z}_i^{\text{agg}} = \text{AGG}_p(\{\mathbf{z}_{p_k, i}^h\}_{k=1}^n). \quad (3.9)$$

The semantic embedding h_i^{sem} is obtained as:

$$\mathbf{h}_i^{\text{sem}} \equiv \tau_{\text{PLM}}(i) = \sigma(\mathbf{W}_o \text{ReLU}(\mathbf{z}_i^{\text{agg}})), \quad (3.10)$$

where $\tau_{\text{PLM}}(i)$ is a function capturing the semantics of i by aggregating multiple prompt-based representations via a PLM (as introduced in Section 3.2.1), and $\sigma(\cdot)$ is the sigmoid function. Given a node i , we then construct the embedding h_i^0 as (3b):

$$\mathbf{h}_i^0 = [\mathbf{h}_i^{\text{pos}} \oplus \mathbf{h}_i^{\text{sem}}]. \quad (3.11)$$

GNN Scoring ④. Inspired by Zhou et al. (2023), our base GNN follows the R-GCN (Schlichtkrull et al., 2018) architecture. The goal of this module is to iteratively propagate and transform node representations by aggregating information from their relation-specific neighborhoods, while conditioning the message passing process on the target relation r_t . At each layer l , the representation of node i is updated by combining its previous-layer embedding with aggregated messages from its neighbors. Formally, the node update rule is defined as:

$$\mathbf{h}_i^{(l)} = \text{ReLU}(\mathbf{W}_0^{(l)} \mathbf{h}_i^{(l-1)} + \mathbf{a}_i^{(l)}), \quad (3.12)$$

where $\mathbf{W}_0^{(l)}$ is a learnable self-loop transformation that preserves and reprojects a node’s own information. The aggregation term $\mathbf{a}_i^{(l)}$ encodes the relational context gathered from the neighborhood of node i . Concretely, the aggregation operates over all relation types and their associated neighboring nodes, accumulating weighted messages from each relation-specific neighborhood:

$$\mathbf{a}_i^{(l)} = \sum_{r \in R} \sum_{j \in \mathcal{N}^r(i)} \alpha_{rr_i j i}^{(l)} \mathbf{W}_r^{(l)} (\mathbf{h}_j^{(l-1)} - \mathbf{e}_r^{(l-1)}). \quad (3.13)$$

Here, $\mathcal{N}^r(i)$ denotes the set of neighboring nodes connected to node i via relation r . Each neighbor $j \in \mathcal{N}^r(i)$ contributes a message that is first transformed by a relation-specific matrix $\mathbf{W}_r^{(l)}$, allowing the model to capture relation-dependent patterns in the message-passing process. Specifically, the aggregation term $\mathbf{a}_i^{(l)}$ is designed to integrate three complementary modeling principles: (i) *entity-relation composition* (Vashishth et al., 2020), which explicitly models the interaction between entity and relation representations during message passing; (ii) *relation-specific transformations*, as introduced in R-GCNs, which allow messages conveyed through different relations to be processed distinctly; (iii) *edge-level attention* (Teru et al., 2020) conditioned on the target relation r_t , which enables the model to selectively emphasize the most informative neighbors for the current prediction task.

(i) Entity-relation composition: To explicitly model entity-relation interactions, we subtract the relation embedding $\mathbf{e}_r^{(l-1)}$ from the neighbor node representation $\mathbf{h}_j^{(l-1)}$ prior to transformation, following the idea of entity-relation composition (Vashishth et al., 2020). The subtraction of the relation embedding $\mathbf{e}_r^{(l-1)}$ implements an entity-relation composition operation, which aligns entity and relation representations in a shared embedding space and enables relation-

conditioned message passing. As a result, the propagated messages encode how well a neighboring entity semantically fits the relation under which the information is transmitted, rather than relying solely on entity-specific features. In addition, relation representations are also updated across layers to maintain consistency with evolving node embeddings:

$$\mathbf{e}_r^{(l)} = \mathbf{W}_{rel}^{(l)} \mathbf{e}_r^{(l-1)}. \quad (3.14)$$

This linear transformation allows relation embeddings to adapt as information is propagated deeper into the network, enabling richer interaction patterns between entities and relations at higher layers.

(ii) Relation-specific transformations: The result of the entity-relation composition operation is transformed by a learnable relation-specific transformation ($\mathbf{W}_r^{(l)}$). This transformation enables the model to capture relation-dependent patterns in how information is propagated across the graph, ensuring that messages passed along different relation types are processed in distinct ways. By parameterizing message transformations at the relation level, the model can better account for the semantic heterogeneity of edges in multi-relational graphs and generate relation-aware messages that are tailored to the underlying graph structure. To reduce the number of parameters and improve generalization, we employ basis decomposition (Schlichtkrull et al., 2018) for the relation-specific transformation matrices $\mathbf{W}_r^{(l)}$.

(iii) Edge-level attention: The contribution of each edge (j, r, i) is further modulated by an attention weight $\alpha_{rrtji}^{(l)}$, allowing the model to selectively focus on the most informative neighbors when reasoning about the target relation r_t . The edge attention weight $\alpha_{rrtji}^{(l)}$ quantifies the importance of an edge (j, r, i) when inferring relation r_t at layer l :

$$\alpha_{rrtji}^{(l)} = \sigma(\mathbf{W}_\alpha^{(l)} \mathbf{s}_{rrtji}^{(l)} + b_\alpha^{(l)}), \quad (3.15)$$

$$\mathbf{s}_{rrtji}^{(l)} = \text{ReLU}(\mathbf{W}_s^{(l)} [\mathbf{h}_j^{(l-1)} \oplus \mathbf{h}_i^{(l-1)} \oplus \mathbf{e}_r^{(l-1)} \oplus \mathbf{e}_{r_t}^{(l-1)}] + \mathbf{b}_s^{(l)}). \quad (3.16)$$

The attention weight $\alpha_{rrtji}^{(l)}$ in Equation (3.15) is obtained by applying a sigmoid function to a learned linear projection of the intermediate representation $\mathbf{s}_{rrtji}^{(l)}$. This design maps the computed compatibility score to a scalar value in the range $(0, 1)$, which can be interpreted as the relative importance of the edge (j, r, i) when reasoning about the target relation r_t .

The intermediate representation $\mathbf{s}_{rrtji}^{(l)}$, computed as in Equation (3.16), cap-

tures the semantic compatibility between the edge and the target relation. It is constructed by concatenating the source node embedding $\mathbf{h}_j^{(l-1)}$, the target node embedding $\mathbf{h}_i^{(l-1)}$, the embedding of the edge relation $\mathbf{e}_r^{(l-1)}$, and the embedding of the target relation $\mathbf{e}_{r_t}^{(l-1)}$. This concatenation allows the attention mechanism to jointly reason over entity-level context and relational semantics, explicitly comparing the relation expressed by the edge with the relation being queried. The subsequent linear transformation and nonlinearity enable the model to learn complex interaction patterns among these components and to assign higher attention to edges that provide stronger evidence for the target relation.

To obtain the final representation of a node, Teru et al. (2020) suggests adopting JK-Connections (Xu et al., 2018), i.e., by concatenating all the intermediate layer representations. After the aggregation, the final score is computed as

$$f(u, r_t, v) = \mathbf{W}_f^T \bigoplus_{l=1}^L [\mathbf{h}_g^{(l)}(u, r_t, v) \oplus \mathbf{h}_u^{(l)} \oplus \mathbf{h}_v^{(l)} \oplus \mathbf{e}_{r_t}^L], \quad (3.17)$$

where $\mathbf{h}_g^{(l)}(u, r_t, v)$ is the subgraph representation, obtained via average pooling over all node representations at level l in the subgraph.

Loss Function. We adopt a margin-based pairwise loss function, which aims at maximizing the score on positive triples and minimizing the score on randomly sampled negative triples:

$$\mathcal{L} = \sum_{(u, r_t, v) \in G} \max(0, f_e(u', r_t, v') - f_e(u, r_t, v) + \gamma), \quad (3.18)$$

where γ is a margin hyperparameter, (u, r_t, v) is a positive triple and (u', r_t, v') is a negative triple.

3.2.3 Experimental Setup

In this section we detail our experimental setup, including datasets, baselines, training and evaluation details. Experiments were conducted with Python 3.8.19 and PyTorch 2.3.0, using an NVIDIA Ampere A100 GPU (64GB VRAM) and CUDA 12.1.

Table 3.10: Statistics of the datasets used in our experiments. The YAGO21K-610 (Zhou et al., 2023) dataset includes ontology triples and entity-type links, while the FB237 dataset variants (Teru et al., 2020) are further processed to extract ontology triples, type links and textual labels.

Dataset	Split	Entities	Relations	Triples	Types	Meta Rel.	Onto. Triples	Type Links	Text Labels
fb237_v1	train	1,594	180	4,245	458	29	680	2,163	1,516
	valid	567	103	489	124	15	86	756	539
	test	550	102	492	113	13	85	764	517
fb237_v1_ind	train	1,093	142	1,993	458	29	680	1,525	1,041
	valid	287	66	206	124	15	86	406	275
	test	301	68	205	113	13	85	434	289
fb237_v2	train	2,608	200	9,739	575	33	865	3,586	2,489
	valid	1,139	143	1,166	160	15	109	1,511	1,083
	test	1,142	140	1,180	153	16	108	1,515	1,094
fb237_v2_ind	train	1,660	172	4,145	575	33	865	2,257	1,561
	valid	548	92	469	160	15	109	757	516
	test	562	107	478	153	16	108	745	524
fb237_v3	train	3,668	215	17,986	732	31	1,060	5,114	3,484
	valid	1,882	183	2,194	196	17	133	2,575	1,787
	test	1,871	179	2,214	192	16	133	2,520	1,773
fb237_v3_ind	train	2,501	183	7,406	732	31	1,060	3,426	2,379
	valid	973	120	866	196	17	133	1,275	920
	test	981	128	865	192	16	133	1,290	924
YAGO21K-610	train	16,357	30	30,000	610	24	1,983	4,861	16,357
	valid	4,388	21	3,000	166	14	248	1,783	4,388
	test	3,938	25	6,970	159	13	248	1,898	3,938

Datasets

We conduct experiments on YAGO21K-610 (Zhou et al., 2023) and three FB15K-237 (FB237 in short) variants (v1–v3) from Teru et al. (2020).

Table 3.10 provides a statistical overview of the datasets utilized in our experiments, detailing their key characteristics, including the number of entities, relations, triples, types, meta-relations, ontology triples, type links, and textual labels.

The model from Zhou et al. (2023) relies on explicit entity-type pairs and an ontology graph for training. FB237 initially lacks these annotations. Therefore, we processed the FB237 variants to extract the necessary type information and construct a corresponding ontology using the following procedure.

Dataset density, defined as $2|T|/|E|$ (Pujara et al., 2017), is the lowest for YAGO21K-610 train (3.67) and increases across FB237 variants (i.e., from 5.33 to 9.80). This pattern also holds for the inference graphs (density ranging from 3.54

for YAGO21K-610 to 5.92 for FB237-V3), allowing us to analyze the impact of type information under varying graph sparsity. For YAGO21K-610, we use the original splits with the provided ontology graph and type links; test entities are unseen during training, while relations are shared. Each FB237 variant contains disjoint train and inductive test graphs with distinct entities but shared relations. For each FB237 variant, we train on its designated training set and evaluate using its corresponding "ind" (inductive) set as the inference graph, with testing performed on its test set. For YAGO21K-610, when evaluating a specific target triple, the inference graph includes all other test triples (excluding the target itself), following Zhou et al. (2023). Since FB237 lacks concept annotations, we build ontology graphs and type links for all variants. Specifically, we construct the ontology graph for our experiments mapping all the Freebase entities appearing in the dataset to their Wikidata identifier, using the publicly available Freebase-Wikidata mappings ⁵. Using the public Wikidata API ⁶, we then retrieved for every mapped entity its respective textual label and the values associated with its "*instance of*" property, which indicates the type(s) an entity is associated to. With the set of relevant concepts established, we constructed the schema-level ontology. For each concept identified in the previous step, its full set of concepts was fetched from Wikidata.

A schema-level triple $\langle \text{Concept}_1 \text{ PropertyLabel } \text{Concept}_2 \rangle$ was generated and added to our ontology graph if, and only if, the target value of a Concept (Concept_2) was itself one of the recognized concepts.

In the entity triples, the entities in the test set do not appear in the train set and valid set, while the relations in both the test set and valid set are included in the train set. We train on the train graph and test on the test graph. In addition, to achieve ontology training, we randomly divide the ontology triples into a train set, a valid set, and a test set using hold-out splitting in the ratio of 80%, 10%, 10%, respectively.

Metrics

We evaluate models using Mean Reciprocal Rank (MRR) and Hits@ K for $K \in \{1, 10\}$, averaging over 5 evaluation runs. Following standard protocol (Teru et al., 2020), each positive test triple is ranked against 50 negative triples generated by

⁵<https://developers.google.com/freebase>

⁶<https://www.wikidata.org/w/api.php>

Table 3.11: Assertion prompts (p_1 - p_6) used in the semantic enrichment step (Section 3.2.2). These templates, with a placeholder for the entity, are fed to the Pre-trained Language Model to elicit representations capturing different semantic aspects (type, geographic context, membership, equivalence, difference, similarity) of the entity.

ID	Aspect	Template
p_1	type	Paris is a type of __
p_2	geographic	Paris is located in __
p_3	membership	Paris is member of __
p_4	equivalence	Paris is equivalent to __
p_5	difference	Paris is different from __
p_6	similarity	Paris is similar to __

Table 3.12: Link Prediction (LP) evaluation on multiple FB237 variants and YAGO21K-610. Best and second-best scores are in **bold** and underlined, respectively. Evaluation uses the strictest tie-breaking policy (Section 3.2.3), assigning the highest (worst) possible rank to the positive triple in case of ties.

Inductive LP Model	FB237-V1			FB237-V2			FB237-V3			YAGO21K-610		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
GraIL (Teru et al., 2020)	.456	34.97	64.44	.618	50.46	<u>82.70</u>	.609	<u>49.94</u>	82.26	<u>.661</u>	62.76	68.68
Zhou et al. (2023)	.398	27.85	64.55	.576	44.69	82.45	.554	41.85	81.42	.673	60.36	<u>76.56</u>
TyleR-RoBERTa-L (2025)	<u>.470</u>	<u>35.66</u>	<u>69.95</u>	.602	47.51	83.28	.630	50.60	86.72	.660	58.26	79.68
TyleR-Llama3-8B (2025)	.481	36.88	70.63	<u>.610</u>	<u>49.37</u>	82.01	<u>.620</u>	<u>49.94</u>	<u>84.46</u>	.651	59.70	69.30

randomly corrupting either its head or tail entity.

Tie resolution markedly affects these metrics. While methods like random tie-breaking (Rossi et al., 2021)—which randomly assign ranks among tied entities—are prevalent, they can lead to an overestimation of true model performance. This issue is particularly evident in sparse settings where limited structural or type information leads to frequent ties, an issue amplified by the candidate pool of 50. To address these concerns and provide a more stringent and reliable evaluation, we adopt a **strict tie-breaking strategy**. This approach assigns the positive triple the highest (i.e., worst-case/pessimistic) rank when its score is identical to one or more negative triples.

Models

To isolate the contribution of our semantic-enrichment module, we focus the comparison on methods that share a similar subgraph-reasoning backbone as

Tyler. We evaluate TyleR against GraIL (Teru et al., 2020), a type-agnostic baseline that relies solely on subgraph structure, and the ontology-enhanced method of Zhou et al. (2023), which explicitly incorporates type information via learnable embeddings and ontological constraints, even though its effectiveness is tied to the availability and quality of type annotations and ontology triples. We chose these baselines to enable a focused and meaningful comparison: (1) GraIL serves as the foundational type-agnostic framework upon which many subsequent methods build (Chen et al., 2021; Mai et al., 2021); (2) the method of Zhou et al. (2023) exemplifies a type-enhanced approach, with few comparable models in literature. In contrast, TyleR is designed for scenarios where explicit type information is scarce as it is able to infer implicit type semantics from PLMs. Our semantic enrichment strategy, detailed in Section 3.2.2, is model-agnostic and compatible with any PLM supporting masked or causal language modeling. We use RoBERTa-Large (Liu et al., 2019) and Llama3-8B (Dubey et al., 2024a) without fine-tuning, aggregating representations from six manually crafted assertion prompts (Table 3.11). To further evaluate the robustness of TyleR with respect to prompt selection, we report the results on FB15K-237-V1 for both RoBERTa and Llama3-8B across varying numbers of templates (Table 3.15). The prompt aggregation function $\text{AGG}_p(\cdot)$ was set to SUM, as it yielded the most consistent results in our experiments (Table 3.13). Moreover, SUM offers better scalability by producing fixed-size outputs regardless of the number of prompts and may also help regularize prompt-specific noise.

3.2.4 Results

Experiments aim to answer three core questions:

- RQ1.** *Does explicit type information improve subgraph-based inductive link prediction?*
- RQ2.** *Can PLMs enhance node representations for subgraph-based inductive link prediction?*
- RQ3.** *Can PLMs mitigate type and structural sparsity challenges in inductive link prediction?*

Table 3.13: Ablation study on the FB237-V1 dataset, evaluating the impact of different Pre-trained Language Models and aggregation functions (Equation 3.9) for semantic embeddings within TyleR. 'TYPE-ONLY' uses only the representation from the p_1 prompt (Table 3.11).

PLM	Aggregation	MRR	Hits@1	Hits@10
TyleR-RoBERTa-L	TYPE-ONLY	.442	32.93	66.49
TyleR-RoBERTa-L	SUM	.470	35.66	69.95
TyleR-RoBERTa-L	MEAN	<u>.468</u>	<u>35.22</u>	<u>68.05</u>
TyleR-RoBERTa-L	CONCAT	.455	34.10	67.76
TyleR-Llama3-8B	TYPE-ONLY	<u>.477</u>	37.12	68.29
TyleR-Llama3-8B	SUM	.481	<u>36.88</u>	<u>70.63</u>
TyleR-Llama3-8B	MEAN	.465	35.51	68.73
TyleR-Llama3-8B	CONCAT	.474	35.95	71.12

Table 3.14: Performance of Llama3-8B when directly evaluating the likelihood of verbalized triples on the YAGO21K-610 dataset by scoring them using the (negated) model perplexity.

PLM	MRR	Hits@1	Hits@10
Llama3-8B (No GNN)	.264	13.21	55.06

Type Information in Subgraph-Based Inductive Link Prediction (RQ1)

Table 3.12 presents link prediction results across various models and datasets, emphasizing the role of type information in inductive link prediction. GraIL, which operates without type information, performs competitively overall. It achieves strong results in both MRR and Hits@10, and obtains the highest Hits@1 on the sparse YAGO21K-610 dataset. This suggests its ability to rank correct entities precisely in low-density settings without relying on type cues. When explicit type information is incorporated, as in Zhou et al. (2023), performance patterns shift: while Hits@10 often remain competitive—or even surpass GraIL on sparse datasets like YAGO21K-610—Hits@1 consistently decline. This indicates that explicit types may mitigate sparsity by providing useful semantic signals, but also introduce complexity that reduces precision in top-ranked predictions. As dataset density increases, the performance gap between GraIL and type-informed models narrows, and in some cases, GraIL even outperforms the latter. This trend suggests that **explicit type information becomes less helpful—and potentially detrimental—in denser graphs**, where structural cues are already

Table 3.15: Ablation study on the FB237-V1 dataset, evaluating the impact of chosen number of prompts (Equation 3.9) for semantic embeddings within TyleR. For all configurations, the aggregation function was set to SUM.

PLM	Templates (p_i)	MRR	Hits@1	Hits@10
TyleR-RoBERTa-L	1 (TYPE-ONLY)	.442	32.93	66.49
TyleR-RoBERTa-L	1-2	.459	34.97	68.00
TyleR-RoBERTa-L	1-3	.451	34.87	65.32
TyleR-RoBERTa-L	1-4	.472	36.00	68.58
TyleR-RoBERTa-L	1-5	.468	35.32	<u>69.46</u>
TyleR-RoBERTa-L	1-6 (ALL)	<u>.470</u>	<u>35.66</u>	69.95
TyleR-Llama3-8B	1 (TYPE-ONLY)	.477	37.12	68.29
TyleR-Llama3-8B	1-2	.458	35.07	69.36
TyleR-Llama3-8B	1-3	.470	35.95	69.46
TyleR-Llama3-8B	1-4	.467	35.22	<u>70.19</u>
TyleR-Llama3-8B	1-5	.471	37.36	65.36
TyleR-Llama3-8B	1-6 (ALL)	.481	36.88	70.63

sufficient. In contrast, implicit type information, as leveraged by TyleR, generally leads to more robust and consistent improvements. While not always achieving the best Hits@1, models using implicit types (TyleR variants) rank first or second across most datasets for both Hits@10 and Hits@1. These models show particular strength on sparse datasets, such as YAGO21K-610, where the gap in Hits@10 is most pronounced. This suggests that **implicit typing is more robust against topological variations in the data**, exhibiting a higher generalization potential.

Addressing RQ1, the benefit of explicit type information is dataset-dependent. It aids relational inference in sparse graphs lacking rich topology, but can add detrimental complexity in denser graphs. Implicit type signals, however, consistently enhance inference, mitigating structural sparsity.

Usefulness of PLM Representations for Implicit Type Signal (RQ2)

The results in Table 3.12 provide compelling evidence that PLMs can significantly enhance node representations in subgraph-based link prediction. However, **the impact of PLMs is not uniform across all datasets**. For example, on FB237-V1 and FB237-V3, RoBERTa-L and Llama3-8B models exhibit competitive performance, especially in terms of Hits@1 and MRR, suggesting that PLMs provide a strong inductive bias for relational reasoning. In contrast, models without PLMs, like GraIL, show lower performance on these datasets, particularly

regarding the Hits@10 metric. This highlights the ability of PLMs to generalize and make more accurate predictions in larger, more complex graphs, where non-PLM models may struggle. Table 3.13 shows the impact of different aggregation strategies on the FB237-V1 dataset, with SUM showing the most consistent results. For example, Llama3-8B with SUM outperforms GraIL across all metrics. In addition, we compare the results of different aggregation strategies with the scenario where only the "type" prompt is considered (i.e., p_1 in Table 3.11), showing consistent improvements.

To comprehensively evaluate the utility of PLMs in link prediction, we further explore the potential of using LLMs directly. Specifically, we verbalize all evaluation triples (including both positive and negative candidates) in the form $label(h) \oplus label(r) \oplus label(t)$. We then employ Llama3-8B to score these verbalized triples based on their (negated) sentence perplexity and report the resulting rankings in Table 3.14. We exclude the FB15K-237 variants from this evaluation due to the lack of clear and consistent relational labels, which makes verbalization unreliable. The performance on YAGO21K-610, which is substantially lower than that achieved by GNN-based approaches, highlights the critical importance of incorporating neighborhood structural information for this task.

*Regarding **RQ2**, PLMs effectively enhance node representations for subgraph-based inductive link prediction. By providing richer semantic features, models like RoBERTa-L and Llama3-8B improve relational inference. Aggregating diverse PLM-derived semantic embeddings (i.e., from different prompts) boosts representation expressiveness.*

Effect on Type and Structural Sparsity (RQ3)

To investigate the effect of our approach on *type sparsity*, we categorize entities into four groups based on the number of explicit type annotations they possess. Group 0 consists of entities with *no explicit type*. Group 1 includes entities with *exactly one* explicit type. The remaining entities (those with *more than one* type) are split into two additional groups based on the median number of types in this subset: the lower 50% form the group labeled ">1 (1st)", and the upper 50% form ">1 (2nd)". For each test triple, we determine the group membership of the *known* entity (irrespective of the type information available for the candidate entities) and report the model performance in Figure 3.6. We report the average Hits@10 across the three FB237 variants for each group. Group 0 represents the most

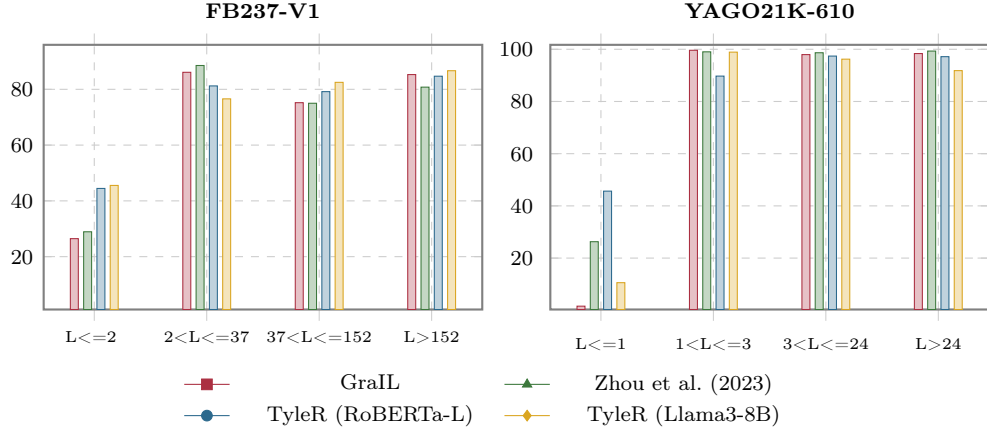


Figure 3.5: Link Prediction (Hits@10) evaluation under **varying structural sparsity conditions** (i.e., the number of edges L in the enclosing subgraph of the target triple, including the target triple) on FB237-V1 (top) and YAGO21K-610 (bottom).

type-sparse setting with entities lacking any explicit type; in this scenario, both variants of TyleR consistently outperform the typeless baseline GraIL across all dataset variants. This reinforces the intuition that **type signals derived from PLMs can enhance inference capabilities in sparse settings**. For instance, on FB237-V1, TyleR (RoBERTa-L) yields a 6.15% relative improvement over the non-PLM baselines, while on FB237-V2, it achieves an even greater gain of 10.75% over GraIL. Interestingly, in the case of entities with multiple types, TyleR continues to outperform the explicit-type-based method of Zhou et al. (2023). This suggests that while explicit type information is useful, its effectiveness may diminish when type annotations are noisy or overly numerous, highlighting the need for better strategies to aggregate multiple type signals.

We further analyze the role of *implicit type information* in addressing *structural sparsity*. For this analysis, we consider the two datasets with the lowest graph density: YAGO21K-610 and FB237-V1. Evaluation triples are grouped into four bins based on the number of edges in their enclosing subgraphs, using percentiles to capture varying levels of sparsity. Figure 3.5 presents Hits@10 across these structural sparsity conditions. The results indicate that PLM-based approaches, particularly those using RoBERTa-L, demonstrate strong performance in extremely sparse subgraphs. For example, RoBERTa-L performs best in scenarios with one or fewer edges (for YAGO21K-610) and more than 152 edges (for FB237-V1), demonstrating its robustness at both ends of the sparsity spectrum. However, in

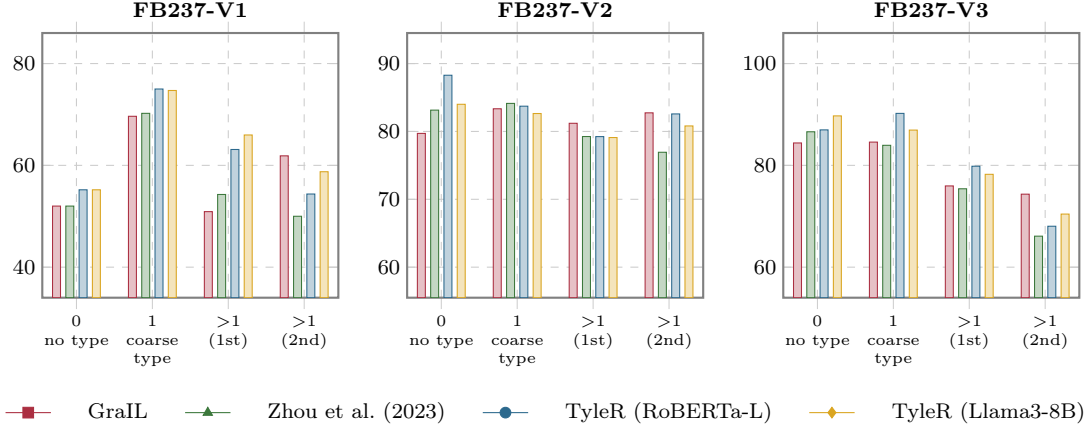


Figure 3.6: Hits@10 performance across four **type sparsity** groups for three FB237 variants, computed according to the number of explicit types linked to each entity (details in Section 3.2.4). The groups, from left to right, represent scenarios with an increasing number of explicit types associated with the known entity.

moderate sparsity settings (e.g., $2 < L \leq 37$ in FB237-V1), models such as GraIL and Zhou et al. (2023) perform comparably or better, due to their reliance on structural patterns that are still informative in such contexts.

Answering **RQ3**, PLM-based approaches such as TyleR address both *type* and *structural sparsity*. They consistently outperform baselines in scenarios with minimal explicit type information or sparse subgraph structures by inferring meaningful semantics from PLMs. Although challenges persist with moderate sparsity and noisy types, PLMs show significant potential.

3.2.5 Limitations

This study employed a set of predefined prompts, which, while effective for the scope of the experiments, may not represent the most informative or optimal configurations. More sophisticated strategies for adaptive prompt selection or prompt tuning could potentially enhance model performance. Exploring these approaches is left as a direction for future research. Additionally, the model hyperparameters were selected empirically, based on extensive experimentation and informed judgment. While this approach yielded strong results, it may not guarantee optimal configurations. A more systematic or exhaustive hyperparameter search could lead to improved outcomes. Nonetheless, the computational cost and complexity associated with such procedures, particularly given the scale

and resource demands of our training setup, render them infeasible within the constraints of this study.

3.2.6 Summary

This section introduced TyleR, a novel inductive link-prediction approach designed to handle incomplete or noisy type information. By leveraging pre-trained language models (PLMs), TyleR enriches node representations with implicit type signals, overcoming the limitations of methods reliant on explicit annotations. Experiments show that TyleR exhibits competitive performance, particularly when type data are sparse or unreliable. The results underscore the potential of PLMs for semantic enrichment, enabling robust link prediction without complete type supervision. Future work will examine domain-specific PLMs, more embedding-aggregation strategies, and broader applications to graph-based tasks. Our codebase is available on GitHub ⁷.

⁷<https://github.com/sisinflab/tyler>

3.3 Related Work

Other works attempted to associate a human interpretable meaning to the behavior of LMs at a more deep inner level (i.e. hidden activations or attention weights). Works such as the one of Bolukbasi et al. (2021) attempt to measure the impact of various inputs on the neural activations of the BERT (Devlin et al., 2019) model, showing that this model exhibits recognizable activation patterns for similar concepts and proving the existence of meaningful global directions in the BERT space. However, the authors also found that the activation interpretation varied to some extent with respect to the dataset of choice. Manning et al. (2020) analyzed the structural properties of the BERT latent space, proving through structural probing that a parse tree can be reconstructed from the hidden representations of the architecture. This work demonstrates that latent representations coming from a self-supervised approach such as Masked Language Modeling indirectly acquire implicit information about syntactic intra-sentence relations.

Petroni et al. (2019) assessed the ability of several popular pretrained language models to act as queryable "off-the-shelf" factual knowledge bases. The underlying assumption is that the standard unidirectional or bidirectional language modeling problem formulation is strictly related to the task of cloze-style question answering. However, the latent representations are not explored in this work, as the task is based on leaving the end-to-end architecture unchanged while solving the task at prompt-level. Prompt-based approaches might fail at retrieving relevant knowledge from an LM based on the prompt choice. As noted by Jiang et al. (2020), existing experimental evidence about the ability of LMs to capture factual knowledge might only represent a lower bound of what these architectures actually know, since we do not possess accurate and proven ways to probe this knowledge.

Several works have attempted to extract factual knowledge from LMs embedding spaces. Yao et al. (2019) attempt to employ LMs in the context of knowledge graph completion: they represent KG triples as textual sentences and extract a representation space through BERT; this representation space is then conditioned on a plausibility function through fine-tuning. In a similar fashion, Biswas et al. (2021) proposed an approach for KGC based on the fine-tuning of GPT-2 (Radford et al., 2019). These approaches prove to be effective in surpassing traditional knowledge graph embedding ones, for their ability to model rich semantic information instead of simple structure. However, they rely only on fine-tuning end-to-end architectures, while the existing semantic information

about factual knowledge in the pre-trained space is not investigated. Contrarily, this proposal aims to investigate whether various pretrained LLM latent spaces can directly incorporate this knowledge as well as preserving the desired language understanding capabilities that come with a task-agnostic pretraining procedure.

An attempt to interpret what the BERT (Devlin et al., 2019) Language Model learns through its pretraining task was made in our previous work (Anelli et al., 2022). This work was moved by the assumption that pre-trained BERT latent space encapsulates semantics about real-world objects that can be leveraged to construct facts via a link prediction procedure. The assumption was then proven by evaluating against the FB15K-237 (Toutanova and Chen, 2015) benchmark dataset. Following this intuition, an entity classification task was then performed to assess the separability of the same space with regard to the taxonomy enforced by the Freebase ontology.

3.3.1 Ontological and Type Knowledge in PLMs

The problem of probing factual and ontological knowledge in PLMs has already been tackled by prior work (Youssef et al., 2023). Among this vast body of literature, our work is particularly aligned with the line of research that focuses on probing conceptual (De Bellis, 2023; Peng et al., 2022) and ontological knowledge (Anelli et al., 2022; De Bellis et al., 2023; Wu et al., 2023) using manually constructed cloze prompts: this kind of approach is based on the construction of ad-hoc prompts to verbalize triples in a KG, where a part of the triple (usually the object entity) is masked (e.g. "*Paris is the capital of [MASK]*"); the PLM is then asked to fill in the blanks, evaluating the ability to infer the missing object. This research was initially propelled by the seminal work of Petroni et al. (Petroni et al., 2019). Subsequent work (Kassner et al., 2021; Pörner et al., 2020) follows the same general idea, proposing refinements over the methodology and datasets. For instance, Jain and Espinosa-Anke (Jain and Anke, 2022) test taxonomical knowledge in PLMs by means of prompt-based object prediction and sentence scoring. More recently, Wu et al. (Wu et al., 2023) proposed a systematic procedure to probe ontological knowledge in PLMs, in terms of real-world classes and relationships between them. A key challenge in prompt-based evaluation of ontological knowledge in PLMs resides in how to construct the prompt and how to map discrete concepts (i.e. entities, classes) to a meaningful textual representation. Most prior works craft prompts manually, while other works (Qin and Eisner, 2021)

enhance the pre-trained embedding vocabulary of existing PLMs with fine-tuned word vectors, i.e. "soft prompts".

3.3.2 Entity Typing with PLMs

Understanding semantic containment in PLMs can enhance tasks fundamentally reliant on the concept of containment, such as entity typing. Prior works (Ding et al., 2022; Huang et al., 2022) frame entity typing as a fill-in-the-blank task, using the predictions from a masked language model (MLM) to assess the class membership of an entity mention m in a sentence x (e.g. " x , m is a type of [MASK]"). They fine-tune a linear verbalizer that maps the predictions to a limited set of class labels. However, this approach learns a static mapping tied to predefined classes, limiting its applicability to new, unseen classes. In our work (Section 3.1), we expand upon the concept of the verbalizer to investigate if it is possible to learn a more general notion of containment extending beyond a fixed set of classes.

3.3.3 Verbalizers and Prompt Learning

A *verbalizer* is a general mapping between a masked language modeling problem and a classification problem. Verbalizers can be manually crafted (Schick and Schütze, 2021), constructed leveraging external knowledge sources (Chen et al., 2022; Ding et al., 2021) or learned (Kho et al., 2023). Verbalizers are used as a conditioning signal to tune an MLM prediction head on a downstream task, typically in few-shot settings.

3.3.4 PLMs for Inductive Link Prediction

Inductive Link Prediction

Inductive Link Prediction (ILP) in Knowledge Graphs (KGs) aims to infer missing links that involve entities unseen during training, thereby enabling models to generalize to evolving KGs. Unlike traditional embedding-based models (Lin et al., 2015; Bordes et al., 2013; Wang et al., 2014), inductive methods explicitly handle unseen entities. Early approaches relied on rule-based reasoning (Yang et al., 2017; Meilicke et al., 2018), but graph neural networks (GNNs) soon became dominant (Hamilton et al., 2017), with GraIL (Teru et al., 2020) leveraging enclosing

subgraph structures for relational inference. Extensions include CoMPiLE (Mai et al., 2021), emphasizing relational directionality, and TACT (Chen et al., 2021), introducing relation-level reasoning. Zhou et al. (2023) incorporate ontological data, but assume complete type information, an assumption that seldom holds in real-world KGs.

Entity Representation with Language Models

Pretrained language models (PLMs) capture factual and relational knowledge from large corpora (Petroni et al., 2019; Brown et al., 2020), encoding rich entity semantics (Zhu et al., 2024) and retrieving factual information via prompting (Wei et al., 2023b). This makes PLMs well-suited for link prediction because they can enrich entity representations. For example, KGBERT (Yao et al., 2019) verbalizes triples as text and fine-tunes BERT to classify their plausibility. Subsequent methods (Zhang et al., 2020b; Daza et al., 2021; Wang et al., 2021) integrate entity descriptions into KG completion to induce embeddings for new entities via PLMs. BERTRL (Zha et al., 2022) exemplifies this trend by injecting GNN-discovered reasoning paths into a BERT-based model. A promising direction involves integrating LLMs with subgraph-based methods to reduce model queries while preserving structural reasoning. Li et al. (2025) propose CATS, a hybrid model that leverages latent type cues and neighbor facts to fine-tune an LLM for triple scoring, combining semantic understanding with explicit subgraph evidence. Unlike prior approaches that fine-tune PLMs, our method (Section 3.2) extracts semantic knowledge from a frozen PLM, and we investigate how effectively such pre-trained models enable a subgraph-reasoning module to capture the type semantics underlying each relation.

3.4 Overview: Choice of Representation

This section provides a high-level overview of the internal representations of PLMs that are accessed and analyzed in the works presented in this chapter. Rather than introducing new methodological components, its purpose is to serve as a unifying reference that clarifies *where* and *how* information is extracted from frozen PLMs across the different approaches, thereby helping readers less familiar with probing-based techniques to better contextualize the chapter.

Although the specific tasks addressed in this chapter differ, ranging from semantic containment detection to inductive link prediction, all methods share a common paradigm: PLMs are treated as fixed computational artifacts, and task-relevant signals are obtained by selectively reading representations produced during a forward pass. The approaches primarily differ in the stage of the computation from which these representations are extracted, and in whether the extracted signal corresponds to a latent hidden state or to the output of the model’s prediction head. Figures 3.7 and 3.8 visually summarize these representation choices for PRONTO (Section 3.1) and TyleR (Section 3.2), respectively.

In the case of PRONTO (Figure 3.7), the choice of representation directly reflects the formulation of semantic containment as a mask-filling problem. Given a pair of nodes from the semantic containment graph, the model is prompted with a natural-language template in which the subject and object are verbalized and separated by a [MASK] token. The resulting prompt is processed by a frozen masked language model, and containment prediction is reduced to interpreting the model’s internal response at the masked position.

Two distinct representation access points are considered. In the verbalizer-free setting (PRONTO-VF), the hidden state produced by the final transformer layer at the [MASK] token position is used as a contextualized semantic representation of the input pair and directly fed to a prediction head. This choice leverages the model’s internal encoding of the masked token before any explicit decoding over the vocabulary takes place. In contrast, all verbalizer-based variants leverage the pre-softmax logits associated with the [MASK] token as initial input, which expose the model’s lexical preferences over the vocabulary. These logits serve as input to learned verbalizers that map token-level likelihoods to a containment score, enabling both interpretable linear mappings and more expressive non-linear transformations (see Section 3.1.1 for more details).

This distinction highlights how PRONTO probes different stages of the same

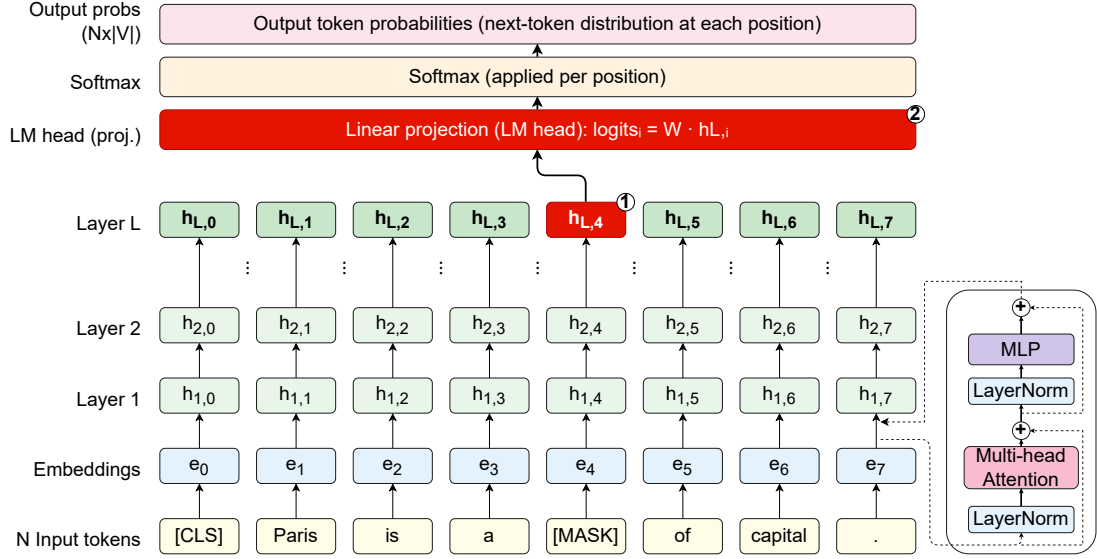


Figure 3.7: Overview of the representations extraction process adopted in Section 3.1. The last-layer hidden state ① is used for the verbalizer-free approach, while the pre-softmax logits ② are used as initial input for all the other verbalizer-based classifiers. More details are in Section 3.1.1.

forward pass: either the latent, continuous representation encoded in the final hidden state, or the explicit lexical signal exposed by the prediction head. Importantly, in both cases, the underlying PLM remains frozen, and the containment reasoning capability is assessed solely through the information already encoded in its internal representations.

For Tyler (Figure 3.8), the representation choice is dictated by the goal of constructing *implicit, type-aware* entity embeddings to compensate for missing or coarse-grained type annotations in inductive link prediction (Section 3.2). Concretely, each entity is described through a set of natural-language assertion prompts (e.g., type, location, membership) (see Section 3.2.2), and each prompt is processed by a frozen PLM to elicit a token-level latent vector. In the masked language modeling setting, this corresponds to the final-layer hidden state at the [MASK] position, which is the model’s internal state used to predict the missing token (e.g., “The type of Paris is [MASK]”). For causal language models (CLMs), the prompt is reformulated without a mask token (e.g., “The type of Paris is:”), the chosen representation is the final-layer hidden state of the last input token, i.e., the state from which next-token prediction is computed.

These prompt-conditioned representations are then normalized and projected

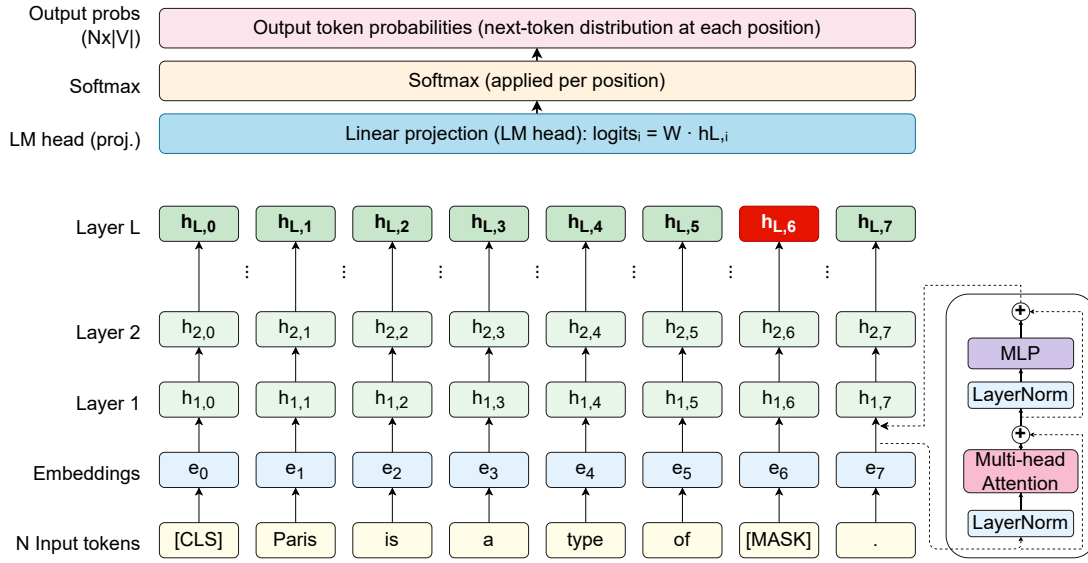


Figure 3.8: Overview of the representation extraction process adopted in Section 3.2. The language model is assumed to be pretrained with a mask-filling objective (e.g., BERT (Devlin et al., 2019)), and representations are extracted from the final transformer layer at the [MASK] token position. For causal language models (LLaMA-3-8B (Dubey et al., 2024b) in our experiments), the input is reformulated without the mask token, which is replaced by a delimiter (":"); in this case, the extracted representation corresponds to the final-layer hidden state of the last input token.

through assertion-specific linear blocks, aggregated across prompts, and transformed into a final semantic embedding. The resulting embedding is concatenated with the structural positional encoding to form the initial node representation, which is subsequently consumed by a GNN. More extensive details on the procedure are presented in Section 3.2.2. Overall, TyleR reads a single token-level final-layer hidden state per prompt as a compact summary of the entity under that semantic facet, enabling consistent semantic enrichment across both MLMs and CLMs while keeping the PLM frozen.

Chapter 4

Factuality in LLMs

Large Language Models (LLMs) have rapidly evolved from experimental research prototypes to central components of modern AI-driven systems. They now power a wide range of applications (Di Palma et al., 2025c) such as healthcare (Pham and Vo, 2024), education (Upadhyay et al., 2023), legal advice (Dahl et al., 2024), recommender systems (Di Palma, 2023), search engines (Yu et al., 2024; Katz et al., 2024), software development (Chen et al., 2024b).

However, the deployment of LLMs in real-world applications has revealed a fundamental tension between their *linguistic competence* and their *factual reliability*. Although LLMs excel at producing natural-sounding text, their outputs are not guaranteed to be accurate or faithful to reality. In low-stakes settings, such as creative writing, this limitation may be acceptable, or even desirable. In contrast, in domains such as healthcare, finance, law, and scientific research, factual errors can propagate misinformation, undermine user trust, and in extreme cases, lead to costly or harmful outcomes.

Consequently, ensuring the reliability of LLM-generated content has emerged as a first-class objective. Reliable systems must not only strive for correctness but also demonstrate appropriate *calibration*: they should be aware of their own limitations and avoid producing statements that cannot be supported by known evidence. Achieving this balance between fluency, informativeness, and accuracy remains an open challenge.

A central manifestation of unreliability in LLMs is the phenomenon commonly referred to as *hallucination*. Broadly speaking, a hallucination occurs when a model produces a statement that is *not supported* by either the provided context or established sources of truth. While this definition is intuitive, hallucinations

are multifaceted and vary significantly in nature depending on the task setting.

We adopt a taxonomy inspired by recent large-scale evaluations of factuality in LLMs (Ravichander et al., 2025):

- **Open-domain hallucinations** arise in tasks where models are queried about factual knowledge of the world. These errors occur when the generated statement is simply false or fabricated. For example, claiming that “*Marie Curie discovered the neutron*” constitutes a hallucination because the asserted fact is objectively incorrect.
- **Content-grounded hallucinations** occur in tasks where models are expected to remain faithful to a provided source, such as summarization, simplification, or paraphrasing. Here, a hallucination corresponds to any information introduced in the output that is *not entailed* by the input, even if it may happen to be factually correct in the real world.

This distinction is critical because hallucinations differ not only in expression but also in their underlying causes. Open-domain hallucinations are often linked to gaps or inconsistencies in the model’s pretraining data, while content-grounded hallucinations frequently arise from semantic processing failures or overgeneralization beyond the scope of the given context. Moreover, factuality assessments must be task-dependent: the same statement may be correct in one scenario but unsupported in another if the source of truth differs. Another useful taxonomy for hallucinations can be defined in terms of the alignment between the LLM’s internal knowledge state about a fact and the faithfulness of its expression of that fact (Liang et al., 2024). Four distinct scenarios can be identified:

1. The model possesses relevant knowledge about a fact and expresses it accurately.
2. The model possesses relevant knowledge about a fact but expresses it inaccurately (we refer to this type of hallucination as a *known hallucination* in Section 4.1).
3. The model lacks relevant knowledge and expresses this faithfully (e.g., by abstaining or indicating uncertainty).
4. The model lacks relevant knowledge and nevertheless produces an inaccurate or fabricated response.

Hallucinations are not merely academic curiosities; they present concrete challenges in practical deployments. As LLMs become deeply integrated into workflows, the cost of incorrect or unsupported outputs grows substantially. Consider the following illustrative examples:

- In scientific assistance tools, hallucinated citations or misattributed claims can undermine reproducibility and compromise trust in automated literature review systems.
- In software engineering contexts, hallucinated package names or APIs can silently propagate faulty dependencies into generated code, resulting in cascading downstream failures.
- In legal and medical advisory systems, unsupported statements can carry significant ethical, financial, and even life-threatening consequences.

Compounding these risks is the deceptive fluency of LLM outputs. Unlike traditional information retrieval systems, which present documents for user inspection, LLMs deliver synthesized responses with high linguistic confidence, making falsehoods difficult to detect without external verification. Recent findings suggest that state-of-the-art models exhibit high variability in hallucination rates across domains and tasks, with even the strongest models producing unsupported statements in non-trivial proportions (Ravichander et al., 2025).

A substantial body of work has investigated strategies for mitigating hallucinations, which can be broadly categorized into three complementary approaches:

One strategy focuses on improving the quality of pretraining and fine-tuning corpora. Noisy, conflicting, or low-quality documents are filtered to reduce exposure to spurious associations. While this can improve overall factual alignment, large-scale datasets inevitably contain outdated information, inconsistencies, and fabricated content, making it infeasible to fully eliminate hallucination risks at the source.

Another line of work seeks to modify model behavior directly through fine-tuning, reinforcement learning from human feedback (RLHF), or reinforcement learning from AI feedback (RLAIF). These techniques encourage models to produce more cautious outputs, hedge uncertain statements, and abstain when appropriate. However, empirical studies have shown that even highly aligned models remain

vulnerable to confident errors, particularly when trained on noisy sources (Kadavath et al., 2022). Alignment alone does not guarantee faithfulness when internal representations are themselves miscalibrated.

A third strategy integrates external verification mechanisms into the generation pipeline. Retrieval-Augmented Generation (RAG) equips LLMs with access to up-to-date evidence sources, while entailment-based verifiers, citation checkers, and domain-specific indices validate generated claims post hoc. While effective in many settings, these approaches face two limitations: (i) retrieval may fail when relevant evidence is unavailable or ambiguous, and (ii) verifiers often struggle with subtle semantic inferences, especially in long-form outputs (Ravichander et al., 2025).

Despite these advances, hallucinations remain pervasive even in state-of-the-art systems. This persistence suggests that hallucinations cannot be fully addressed by external constraints alone and highlights the need for approaches that leverage the model’s own internal representations to anticipate and mitigate factual failures.

4.1 The Case for Self-Verification

If LLMs already encode rich representations of world knowledge (Chapter 3), then it is natural to ask: *do they also encode signals about when they are likely to generate something that is factually incorrect?* Recent work suggests that they do. Probing-based studies have demonstrated that factuality-related signals are often linearly separable in model hidden states (Azaria and Mitchell, 2023; Marks and Tegmark, 2023; Kadavath et al., 2022). That is, models may internally “*know*” whether a generated statement is true, even when incorrect.

This observation opens the door to an alternative paradigm: *self-verification*. Instead of relying solely on retrieval, post-hoc validation, or ensembles of external verifiers, we can directly leverage the model’s own representations to estimate whether a generated statement is factually correct. Such an approach promises several advantages:

- **Scalability and efficiency.** Self-verification avoids the computational and engineering overhead of maintaining retrieval pipelines, external knowledge bases, or multi-model ensembles.
- **Task generality.** Because factuality signals are embedded in the model’s

internal activations, the same mechanism can potentially generalize across diverse tasks without redesigning bespoke verifiers for each domain.

- **Selective answering.** Self-verification enables adaptive response strategies: models can abstain, retrieve evidence, or flag uncertainty when confidence is low, without requiring expensive external checks for every query.

Known hallucinations, i.e., cases in which a model encodes the correct fact but fails to retrieve or express it consistently, represent particularly high-impact failure modes in deployed systems. Traditional mitigation strategies have limitations: retrieval-based approaches cannot fully resolve these errors, external verifiers may introduce unnecessary computational overhead when the knowledge is already present, and model alignment alone is often insufficient. Effectively addressing known hallucinations requires a deeper understanding of whether, and how, LLMs encode self-assessable signals of correctness, and how these signals can be harnessed to build systems that are both reliable and efficient.

In effect, if models already *implicitly know* when they are likely to hallucinate, leveraging these internal signals could yield more reliable systems at significantly lower cost. However, key challenges remain: much of the existing literature evaluates LLMs using artificially constructed, template-based datasets (Azaria and Mitchell, 2023). While such datasets are useful for isolating specific phenomena, they often fail to capture the variability, ambiguity, and diversity of real-world LLM outputs. In particular, negative examples in these datasets are frequently generated by randomly corrupting a ground-truth sentence, producing statements that are extremely unlikely or out-of-distribution for the model. In these scenarios, a probing task may end up measuring the model’s generation likelihood—how probable it is that the model would produce a given sentence—rather than its factual correctness. This distinction is critical: hallucinations arise precisely when a model generates statements that are plausible yet false, not when it rejects implausible or contrived examples.

This chapter lays the conceptual foundation for exploring these questions, positioning *self-verification* as a promising and cost-effective complement to existing mitigation strategies.

Contributions. This chapter investigates the encoding of factuality-related information within the hidden representations of LLMs. Prior work (Azaria and Mitchell, 2023) has demonstrated that such signals are present and detectable.

However, the reliance on manually crafted datasets in these foundational studies raises questions about the generality of the findings, as such datasets may not accurately reflect the distributions of text that LLMs produce in practice.

The work presented in Section 4.2 aims to rigorously validate and assess the generality of the claim that LLM hidden states encode factuality. This is achieved by moving beyond static, manual benchmarks and evaluating representations using datasets that are more closely aligned with the models’ own output characteristics. This investigation provides a more nuanced assessment of factuality encoding, evaluating the robustness of prior findings under realistic conditions. The primary contributions are twofold:

1. We **reproduce the methodology of Azaria and Mitchell (2023)** to ensure transparency and comparability.
2. We **propose two strategies for generating realistic datasets** and systematically assess how well both the original and newly developed models generalize.

Specifically, we develop two new dataset creation strategies:

- A **perplexity-based negative sampling strategy** that enhances the original generation mechanism by leveraging the LLM’s token distribution to produce more plausible, challenging negative examples.
- A **realistic LLM-generated fact sampling strategy**, which uses Question Answering datasets to elicit authentic LLM responses, better capturing the kinds of statements that models are likely to generate in practice.

Our empirical findings indicate that while hidden representations in two widely used large language models encode certain signals of factuality, enabling effective factuality assessment in controlled settings with limited sentence variability, simple factual structures, and narrow domains, these signals appear to be insufficient in more realistic contexts characterized by finer-grained factual distinctions and greater domain diversity.

Through these contributions, we lay the groundwork for more robust factuality assessments and provide practical guidance for improving the reliability of LLMs across diverse applications. Our goal is to rigorously test the claim that hidden states “*know when the model is lying*,” investigating whether they reliably signal factual correctness or potential hallucinations. In other words, ***are the hidden states hiding something?***

4.2 *Are the Hidden States Hiding Something?* Testing the Limits of Factuality-Encoding Capabilities in LLMs ¹

Recent research suggests that LLMs may encode internal representations of factuality in their hidden states, indicating an awareness of whether a generated statement is true or false (Chen et al., 2024a). These efforts led to the development of approaches to evaluate the factual accuracy of the LLM outputs given their internal representations (*factuality "self-evaluation"*). Self-evaluation can be used to identify gaps in the knowledge of an LLM, improving truthfulness and transparency through abstention mechanisms (Feng et al., 2024), fact verification (Wadden et al., 2020), and self-correction (Ji et al., 2023). Azaria and Mitchell (2023) suggest that LLMs have “some internal notion as to whether a sentence is true or false, as this information is required for generating (or predicting) following tokens.” Based on this assumption, they propose a neural classifier to discern factual from non-factual statements based on hidden layer activations. However, the datasets used to evaluate the probe present limitations since they contain trivially incorrect statements (e.g., “*The zebra uses flying for locomotion*”) that easily fail to align with the generative patterns of LLMs. Additionally, the false statements are generated using random substitutions of the true terms with little regard for the plausibility of negative samples. This misalignment not only weakens the generalizability of results but also raises concerns about the applicability of these models to real-world scenarios where false statements may be subtle or nuanced. Additional illustrative examples of such statements are provided in Table 4.5. This study addresses these gaps by generating more plausible datasets (see Figure 4.1) to explore LLM factuality encoding and evaluate refined models.

As a foundation for this analysis, the methodology of Azaria and Mitchell (2023) is first reproduced to ensure transparency and establish a baseline. The core contribution is the introduction and evaluation of two novel strategies for generating more realistic, LLM-aligned datasets. The first is a **perplexity-based negative sampling strategy** designed to enhance the original generation

¹Based on our paper, “Are the Hidden States Hiding Something? Testing the Limits of Factuality-Encoding Capabilities in LLMs”, in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pp. 6089–6104 (2025).

mechanism by leveraging the LLM’s token distribution. The second is a **novel strategy to sample realistic LLM-generated facts**, which utilizes Question Answering datasets to elicit responses directly from the LLM. This work then systematically assesses how well both the original and newly developed models generalize across these varied datasets. Through this analysis, the groundwork is laid for more robust factuality assessments, offering practical guidelines for enhancing the reliability of LLMs in diverse applications.

4.2.1 Reproduction of Prior Work: Settings

In their study, Azaria and Mitchell (2023) investigate whether LLMs internally represent the factuality of sentences. This section summarizes the dataset generation approach they employed and the specific probing architecture used in their study.

Dataset Generation Strategy

To explore whether LLMs internally represent the factuality of statements, the authors constructed a *True-False dataset* of facts labeled as either True or False, covering six disjoint topics: *Cities*, *Inventions*, *Chemical Elements*, *Animals*, *Companies*, and *Scientific Facts*. To generate the dataset, for the first five topics, i.e. with the exception of *Scientific Facts*, they selected **true statements** from reliable sources (see Table 4.1) and produced **false statements**, replacing part of a true statement (e.g. “*Hydrogen has an atomic number of 1*”) with randomly sampled incorrect information (“*Hydrogen has an atomic number of 34*”). For *Scientific Facts*, they employed ChatGPT (13 Feb 2023) as a generator of true/false sentences, and two human annotators manually verified their correctness. The authors publicly release the dataset, which we refer to as the “*True-False dataset*”. Furthermore, the authors constructed a second dataset using the OPT-6.7b model, which we refer to as the ‘**OPT-Generated Dataset**’. To create this dataset, the model was prompted with a true statement absent from the True-False dataset and then used to generate a subsequent sentence. The responses were manually fact-checked and annotated by three independent human judges. Non-factual responses were filtered out, resulting in a set of 245 statements.

Dataset	Sentences	Source	(%) True
Cities	1,458	SimpleMaps dataset	50
Inventions	876	Wikipedia’s list of inventors	53
Elements	930	PubChem’s periodic table	50
Animals	1,008	National Geographic Kids	50
Companies	1,200	Forbes Global 2000 List 2022: The Top 200	50
Scientific Facts	612	ChatGPT and human annotators	50

Table 4.1: True-False dataset categories, original sources, and label splits from (Azaria and Mitchell, 2023).

Internal States Analysis via SAPLMA

To investigate whether LLMs internally represent the factuality of statements, Azaria and Mitchell (2023) developed a probe (**S**tatement **A**ccuracy **P**rediction based on **L**anguage **M**odel **A**ctivations) that predicts the factual accuracy of a statement by analyzing the hidden layer activations of an LLM. SAPLMA is a feedforward neural network designed to classify statements as true or false. It consists of three hidden layers (256, 128, 64) and a sigmoid output activation. The model is trained using the Adam optimizer for five epochs without hyperparameter tuning. The authors studied two LLMs, namely OPT-6.7b (Zhang et al., 2022) and Llama 2-7b (Touvron et al., 2023b), both consisting of 32 layers. To identify which layers best capture factuality, they trained five SAPLMA models, forwarding each statement in the True-False Dataset as input to the LLMs and extracting the corresponding activation values from the 32nd (last layer), 28th, 24th, 20th, and 16th layers. These activations serve as input for training the classifiers. To ensure generalizability, i.e., making SAPLMA independent of specific topics, the authors adopted a cross-validation strategy using a leave-one-topic-out approach to train the classifier on five topics and test the probe on the held-out topic.

Reproducibility Settings

In this section, we provide details on the datasets and experimental settings for reproducing the work of Azaria and Mitchell (2023). Our goal is to answer the Research Question (**RQ0**): *Can we reproduce the results reported by Azaria and Mitchell (2023)?* Although the code is not publicly accessible, the authors made it available upon request.

Dataset generation. The code provided by Azaria and Mitchell (2023) contains all the necessary material to recreate the entire dataset generation process. How-

Table 4.2: Replicated SAPLMA performance on the True-False dataset across the selected layers. The results labeled as ‘Orig.’ are taken from the original work, while those labeled as ‘Repr.’ are the replicated results from this study.

Layer		Cities		Inventions		Elements		Animals		Companies		Facts		Average	
		Llama2 OPT		Llama2 OPT		Llama2 OPT		Llama2 OPT		Llama2 OPT		Llama2 OPT		Llama2 OPT	
Last	Orig.	.7574	.7796	.6735	.5696	.6814	.5760	.7338	.6022	.6736	.6925	.7444	.6498	.7107	.6449
	Repr.	.7939	.7836	.7470	.5603	.7057	.5656	.7133	.5984	.6463	.6900	.7894	.6640	.7326	.6437
28	Orig.	.8146	.7732	.7207	.5761	.6767	.5907	.7249	.5777	.6894	.7247	.7662	.6618	.7321	.6507
	Repr.	.8261	.8014	.7221	.5938	.6746	.5931	.7046	.5945	.6860	.7252	.7976	.6639	.7351	.6620
24	Orig.	.8722	.7963	.7816	.6712	.6849	.6211	.7394	.5800	.7094	.7758	.7858	.6868	.7622	.6886
	Repr.	.8619	.8043	.7737	.6604	.6789	.6172	.7415	.6095	.7049	.7844	.7910	.6804	.7586	.6927
20	Orig.	.8820	.8125	.8459	.7268	.6950	.6197	.7758	.6058	.8319	.8122	.8053	.6819	.8060	.7098
	Repr.	.8672	.8118	.8584	.7222	.6761	.6218	.7736	.6208	.8254	.8160	.8065	.6734	.8012	.7110
16	Orig.	.9223	.7435	.8938	.6400	.6939	.5645	.7774	.5800	.8658	.7570	.8254	.6237	.8298	.6515
	Repr.	.9174	.7554	.8847	.6403	.7005	.5732	.7883	.5693	.8672	.7760	.8104	.6340	.8281	.6580

ever, their template-matching code is influenced by randomness in the generation of false statements, and a random seed is not set. Due to this non-deterministic behavior, recreating their dataset using the original code was unfeasible. However, the authors released their dataset, allowing us to reproduce their exact dataset settings.

SAPLMA reproducibility. To reproduce the results of the original study, we trained 20 SAPLMA probes for each of the following layers: the 32nd, 28th, 24th, 20th, and 16th, over 5 epochs, resulting in a total of 100 probes. We employed Llama 2-7b and OPT-6.7b, both using half-precision (16-bit float) parameters, with a default temperature of 0.8 for Llama 2-7b and 1.0 for OPT-6.7b. The hardware used for the experiments was an Intel(R) Core(TM) i7-5820K paired with an NVIDIA RTX 3090 graphics card. The authors do not specify from which token they extract the associated hidden state. However, code inspection led to the identification of the last token as the target state.

4.2.2 Experimental Reproducibility Results

To answer **RQ0**, we report the results for the reproduction of the experiments in Tables 4.2, 4.3 and 4.4. The results labeled with ‘Orig.’ are retrieved from the original work, while the ones achieved in the reproducibility study are labeled with ‘Repr.’.

Table 4.3: Replicated baselines performance on the True-False dataset. The results labeled as ‘Orig.’ are taken from the original work, while those labeled as ‘Repr.’ are the replicated results from this study.

Model		Cities	Inventions	Elements	Animals	Companies	Facts	Average
BERT	Orig.	.5357	.5537	.5645	.5228	.5533	.5302	.5434
	Repr.	.5257	.5611	.5435	.5603	.5302	.5361	.5428
3-shot	Orig.	.5410	.4799	.5685	.5650	.5538	.5164	.5374
	Repr.	.5416	.4800	.5685	.5652	.5539	.5115	.5368
5-shot	Orig.	.5416	.4799	.5676	.5643	.5540	.5148	.5370
	Repr.	.5416	.4800	.5676	.5643	.5540	.5082	.5359
It-is-true	Orig.	.5230	.5068	.5688	.4851	.6883	.5840	.5593
	Repr.	.5233	.5046	.5688	.4831	.6875	.5856	.5588

Table 4.4: Reproduced SAPLMA performance on the **OPT-Generated Dataset** (Section 4.2.1). The results labeled as ‘Orig.’ are taken from the original work, while those labeled as ‘Repr.’ are the reproduced results.

Layer		Accuracy	AUC	Accuracy with Optimal Threshold	Average Threshold
Last-layer	Orig.	.6187	.7587	.7052	.8687
	Repr.	.6406	.7720	.7264	.8910
28th-layer	Orig.	.6362	.7614	.7134	.8838
	Repr.	.6410	.7686	.7203	.8276
24th-layer	Orig.	.6134	.7435	.6988	.8801
	Repr.	.6206	.7496	.6973	.8500
20th-layer	Orig.	.6029	.7182	.6587	.9063
	Repr.	.5965	.7183	.6669	.8868
Middle-layer	Orig.	.5566	.6610	.6500	.8123
	Repr.	.5579	.6760	.6468	.7948
BERT	Orig.	.5115	.5989	.5705	.9403
	Repr.	.5522	.6092	.5689	.7939
3-shot	Orig.	.5041	.4845	-	-
	Repr.	.5041	.4845	-	-
5-shot	Orig.	.5125	.4822	-	-
	Repr.	.5125	.4822	-	-

Reproduction of SAPLMA Results on the True-False Dataset

Table 4.2 reports SAPLMA’s performance on the True-False dataset across different layers of Llama 2-7b and OPT-6.7b for six categories: Cities, Inventions, Elements, Animals, Companies, and Facts. Results indicate that Llama 2-7b consistently outperforms OPT-6.7b across all layers and categories. Middle layers (16, 20, 24) achieve the highest performance, while accuracy declines toward the final layer. The reproduced results closely align with the original findings, with minor deviations observed across specific categories and layers. Moreover, although

OPT-6.7b shows greater variability, the ranking of layers remains consistent with the original work. Both experiments confirm that factuality information is more effectively encoded in the middle layers (16–24) than in the final layer. Additionally, we reproduce the baseline used by the authors to compare SAPLMA. Specifically, their baseline includes a trained SAPLMA on BERT activations, a few-shot approach where the LLM is prompted with a sentence and asked to label it as ‘True’ or ‘False,’ and an ‘It-is-true’ test. In this test, the LLMs were asked: *Is it true that X?* and *Is it false that X?*, where X is a dataset sample. A response was considered correct if the model assigned a higher probability to the ‘True’ token. Table 4.3 summarizes the baseline performance. BERT achieves the highest average performance among non-prompted methods, demonstrating its effectiveness in factual classification, and the results indicate high reproducibility. The It-is-true baseline yields the highest average performance, particularly in the ‘Companies’ topic. The results demonstrate consistent reproducibility.

In general, the reproduction on the True-False Dataset is considered successful based on the following observations: (i) the overall performance trends remain consistent, with Llama 2-7b outperforming OPT-6.7b; (ii) observed deviations are minor and do not indicate fundamental inconsistencies; (iii) the relative ranking of layers remains unchanged, reinforcing previous findings; and (iv) baseline methods retain their rankings, confirming the validity of the original results.

Reproduction of the Results on the OPT-Generated Dataset

Table 4.4 presents the results of the reproduced experiments on the OPT-Generated Dataset. Performance is evaluated using Accuracy, AUC, Accuracy with an Optimal Threshold (selected by estimating it from a held-out validation set), and the Average Optimal Threshold. The final layers, specifically the 28th and final layers, outperform the middle and lower layers in terms of Accuracy and AUC. The reproduced results closely align with the original findings, exhibiting only minor variations. Accuracy with an optimal threshold consistently exceeds raw accuracy, suggesting that tuning the decision boundary improves performance.

Regarding baselines, BERT exhibits lower accuracy compared to the LLMs’ last layers, with slight improvements over the original results. Notably, the 3- and 5-shot prompting results were identical between the original and reproduced experiments.

RQ0: *Can we reproduce the results reported by Azaria and Mitchell (2023)?*

This reproducibility study demonstrates a high degree of alignment with the original results, confirming the validity of previous findings. It shows that the ranking and trends remain unchanged, reinforcing the robustness of the results.

4.2.3 Novel Dataset Generation Strategies for Factuality Self-Evaluation

This section introduces two novel dataset generation strategies to investigate LLM factuality self-evaluation. To contextualize our approach, we first examine the limitations of the True-False dataset by Azaria and Mitchell (2023). While it provides a structured framework for evaluating LLMs, its construction imposes constraints that may limit generalizability of the findings. Specifically, true and false statements are derived from tabular data using predefined templates. We argue that this approach suffers from several limitations:

- **Adherence to predefined templates:** The use of fixed templates limits the linguistic expressiveness of the dataset, potentially constraining the probe classifier’s ability to generalize beyond rigidly structured statements (e.g., *<company> operates in the industry of <industry>*).
- **Distribution misalignment:** The statements are constructed from tabular data rather than generated by the LLM itself. As a result, the dataset may not align with the LLM’s generative distribution. For instance, a niche true fact in the dataset –but unknown by the model– could have high perplexity for the LLM, undermining the study’s core premise: evaluating an LLM’s intrinsic ability to "judge" its false claims.
- **Lack of consideration for the LLM’s knowledge state:** LLMs exhibit uneven factual knowledge based on their pretraining data, with strengths in some domains and gaps in others. A model can assess a statement factuality only if it has prior exposure to it. The dataset does not account for these inconsistencies: it evaluates whether an LLM can detect factual errors without considering whether the model actually possesses knowledge of the fact.
- **Differences in cardinality:** Some properties in the dataset have fewer admissible values, making certain facts easier to evaluate. For example, a statement like "*<element> appears in its standard state as _*" has fewer possible values (i.e., $\{Solid, Liquid, Gas\}$) compared to statements like "*<city> is a city in*

— " that involve a broader set. This imbalance in complexity may bias the evaluation process.

These limitations impact the interpretability of results when evaluating an LLM’s internal representations of factuality. To ensure a more realistic assessment of an LLM’s self-evaluation of factuality, we propose two strategies to address these issues. The first strategy samples statements from tabular data to better align with the LLM’s generative predictions. The second strategy involves sampling LLM-generated facts as answers to questions from a well-known Question Answering dataset.

Perplexity-based Dataset Construction

This section presents a novel dataset generation strategy to address limitations in the Azaria and Mitchell (2023) True-False dataset, particularly distribution misalignment and implausible negative samples. To improve negative sampling based on random property-object substitutions, we introduce a **perplexity-guided probabilistic sampling** method, which re-weights false statements based on perplexity for better alignment with LLM output distributions. Since perplexity depends on the model, the same LLM under evaluation must generate false statements to ensure consistency, resulting in a model-dependent dataset tailored to the characteristics and biases of the LLM being studied.

True statements are initially constructed, as in Azaria and Mitchell (2023), by directly inserting correct entity-property pairs into pre-defined sentence templates. Regarding the false statements generation, we proceed as follows: (i) for each true statement, all unique alternative property values are gathered from the entire dataset; (ii) a candidate sentence is created for each alternative property by inserting it into the corresponding template; (iii) the target LLM (i.e., OPT-6.7b or Llama 2-7b) computes the perplexity of each candidate sentence. This perplexity score serves as a plausibility metric, with lower perplexity indicating a more plausible (yet incorrect) statement.

Given a true statement, we define C as the set of potential candidate sentences, which includes the true statement. Furthermore, $C' \subset C$ is defined as the subset of candidate false sentences (i.e. excluding the true statement). Candidates $c \in C$ are ranked based on their perplexity scores, with lower scores indicating higher plausibility.

Since perplexity can be interpreted as a measure of plausibility, we operate

under the assumption that an LLM possesses factual knowledge about a fact if the fact is assigned a "sufficiently low perplexity". Conversely, a high perplexity score for the true statement suggests that the LLM lacks knowledge of the fact. Given the limitations discussed earlier, we aim to evaluate the LLM ability to discern between true and false statements when it possesses the relevant knowledge. Therefore, we exclude instances where the LLM exhibits limited knowledge about the true statement: if the true statement does not rank among the lowest k perplexity candidates, the generation of that instance is discontinued, and the next true fact is considered. In practice, we define k as $k = \alpha|C|$, where α is a hyperparameter ($0 < \alpha < 1$). This accounts for the varying cardinality of the property ranges in the dataset, ensuring that the threshold for "sufficiently low perplexity" is adjusted based on the number of possible values for a given property.

In addition, we want to simulate a real hallucination scenario where the LLM is uncertain between the true fact and plausible alternatives: given the perplexity score function $PP(\cdot)$, all false candidates c with a perplexity score $PP(c) < (1 + \beta)PP(true)$, where $0 < \beta < 1$ is a hyperparameter, are considered, resulting in a reduced set of candidates C^* . A min-max normalization is applied to their perplexity,

$$\text{NormPP}(c) = \frac{PP(c) - \min_{c \in C} PP(c)}{\max_{c \in C} PP(c) - \min_{c \in C} PP(c)}, \quad (4.1)$$

The normalized perplexities are transformed using a plausibility score function $s(\cdot)$, i.e., lower perplexity scores result in higher plausibility scores. The scores are then normalized to ensure that they sum to 1 and are treated as a probability distribution over the candidates:

$$s(c) = e^{-\text{NormPP}(c)}, \quad P(c_i) = \frac{s(c_i)}{\sum_{c_j \in C^*} s(c_j)}. \quad (4.2)$$

The normalization guarantees that $P(c_i)$ values are suitable for sampling. Finally, a mixture of top-k and nucleus sampling (Holtzman et al., 2020) is employed to sample the candidate for insertion into the template. Specifically, we apply top-k and nucleus sampling sequentially: we select the top-k highest-scoring candidates and then refine this set by choosing the smallest subset whose cumulative probability reaches a predefined threshold. This process generates a coherent yet factually incorrect statement that is more realistic and closely aligned with the LLM internal token prediction patterns. In Section 4.2.4 we detail the selected

values for the hyperparameters α , β , k , and p .

LLM-Generated Dataset Construction

The strategy in Section 4.2.3 constructs a balanced true-false dataset from tabular data but has inherent limitations. While enabling direct comparison with Azaria and Mitchell (2023), it restricts generative models to rigid templates, limiting expressiveness. Additionally, its reliance on fixed candidate sets can lead to easily classifiable false statements when the set is small. Finally, template-based sampling reduces diversity, likely due to token bias: we found LMs consistently assigning lower perplexity to certain property-object pairs, regardless of the subject. To more realistically assess an LLM factuality self-evaluation, we propose generating both true and false statements directly from the model, overcoming the limitations of template-based approaches. This involves (a) a method to elicit diverse factoid statements from the LLM, and (b) a strategy to annotate statement veracity, addressing biases and inconsistencies in the process. Consider a Question Answering dataset composed of N questions, $D_{QA} = \{(q_i, a_i)\}_{i=1}^N$, where each question q_i has a corresponding ground-truth answer a_i . Given a LLM $\mathcal{M}$, we prompt it with each question K times, yielding a set of generated answers. This results in an extended dataset $D_{QA}^{\mathcal{M}} = \{(q_i, a_i, \{a_{i,k}^{\mathcal{M}}\}_{k=1}^K)\}_{i=1}^N$. Following LLM-as-judge (Gu et al., 2024; Calderon et al., 2025) practices, the LLM-generated answers in $D_{QA}^{\mathcal{M}}$ can be annotated using an oracle LLM, which we assume is able to evaluate the veracity of each answer $a_{i,k}^{\mathcal{M}}$ given the ground-truth answer a_i and the question q_i . This operation results in an annotated dataset

$$\hat{D}_{QA}^{\mathcal{M}} = \{(q_i, a_i, \{a_{i,k}^{\mathcal{M}}, v_{i,k}^{\mathcal{M}}\}_{k=1}^K)\}_{i=1}^N, \quad (4.3)$$

where $v_{i,k}$ is a veracity label assigned by the oracle, indicating whether the generated answer $\hat{a}_{i,k}$ is correct ($v_{i,k} = 1$) or incorrect ($v_{i,k} = 0$).

Increasing K enhances the reliability of responses by offering a more accurate evaluation of the LLM knowledge state regarding a question. This evaluation can lead to three possible outcomes: (i) a high proportion of correct answers suggests the LLM fully understands the facts; (ii) a high proportion of incorrect answers indicates the LLM lacks or has partial knowledge, preventing correct responses; (iii) a mix of correct and incorrect answers implies knowledge with a tendency toward hallucination. This study assumes that an LLM can only encode factuality

regarding a generated fact if it has some knowledge about it. Therefore, we focus exclusively on the third scenario. This scenario also naturally produces a balanced dataset, including true and false variations of the same fact. We define the correct answer ratio $p_i^{\mathcal{M}} = \frac{1}{K} \sum_{k=1}^K v_{i,k}^{\mathcal{M}}$. We consider questions whose p_i is around 0.5 with a tolerance hyperparameter τ , that is $|p_i^{\mathcal{M}} - 0.5| < \tau$. The dataset is obtained by selecting the answers and their veracity labels satisfying the condition:

$$D_{Facts}^{\mathcal{M}} = \{(a_{i,k}^{\mathcal{M}}, v_{i,k}^{\mathcal{M}}) : |p_i^{\mathcal{M}} - 0.5| < \tau\}_{i=1}^N. \quad (4.4)$$

4.2.4 Experiments and Discussion

This section describes the experimental setup used to extend the prior investigation, presents the results, and discusses their implications. We address the following Research Questions:

RQ1 *Are probing classifiers trained as in Azaria and Mitchell (2023) capable of generalizing to True-False sentences with similar perplexity?*

RQ2 *Can the same probes generalize to facts generated by LLMs?*

To ensure a fair and direct comparison with the work of Azaria and Mitchell (2023), we conduct our experiments using the same LLMs, namely OPT-6.7b and Llama 2-7b.

Dataset Generation Setup

To generate the datasets for training the probe and alleviate the limitations of Azaria and Mitchell (2023), we employ the perplexity-based sampling procedure described in Section 4.2.3 on the same tabular data employed by Azaria and Mitchell, which was made available upon request. Specifically, we set the needed hyperparameters as follows: $\alpha = 0.1$, $\beta = 0.1$, $k = 10$ and $p = 0.9$. We excluded the *Scientific Facts* topic from our perplexity-based sampling procedure, as its original version was generated by ChatGPT and not from tabular data. The properties used for sampling are analogous to the ones employed by the original authors in their dataset generation. However, for the *Cities* topic, we restrict the analysis to facts like "<city> is city in <country>." Differently from the original authors, we avoid generating facts such as "<city> is the name of a city/country" having only two possible values, potentially resulting in too many easy-to-classify samples. Table 4.5 reports examples of false sentences generated using the Llama 2-7b model. Table 4.6 summarizes the statistics for the two refined True-False

Table 4.5: Examples of false sentences generated with the Llama 2-7b model, using the perplexity-based generation strategy described in Section 4.2.3.

True Sentence	Original Negative (Azaria and Mitchell, 2023)	Generated Negative (Llama 2-7b)
The crocodile has a habitat of freshwater.	The crocodile has a habitat of various .	The crocodile has a habitat of grassland/savanna .
The zebra has distinctive black and white stripes, which may help deter flies and provide camouflage.	The zebra is a fast swimmer and can maintain high speeds for extended periods of time.	The zebra is the fastest land animal , reaching speeds up to 60-70 mph.
Tantalum has the symbol Ta.	Tantalum has the symbol Cs .	Tantalum has the symbol Tm .

Table 4.6: Novel generation of our True-False dataset, following the approach described in Section 4.2.3, including number of sentences and percentage of true samples.

Dataset	Llama 2-7b		OPT-6.7b	
	Sentences	(%) True	Sentences	(%) True
Cities	674	50	756	50
Inventions	336	50	202	50
Elements	118	50	220	50
Animals	116	50	114	50
Companies	326	50	310	50

datasets, while Figure 4.1 shows the differences in average perplexity between the original and the proposed datasets.

We construct an LLM-generated dataset following the procedure described in Section 4.2.3. Our experiments use TriviaQA (Joshi et al., 2017), a dataset of question-answer pairs collected from 14 trivia and quiz-league websites. Given the limited computational resources, we limit our focus to the validation split of TriviaQA Wikipedia questions. For each LLM in our analysis, we set $K = 10$ and $\tau = 0.1$, and we discard the answers composed of less than 5 tokens to improve self-consistency. For annotation, we use **GPT-4o mini** with a 3-shot learning strategy, where three examples are provided. Figure 4.2 illustrates the example prompt used in our text generation pipeline for TriviaQA (Joshi et al., 2017). Ten examples are drawn from the training Wikipedia split, with their answers manually crafted based on the corresponding ground truth. These examples are prepended to the target question to establish a consistent response pattern for the model. To produce an

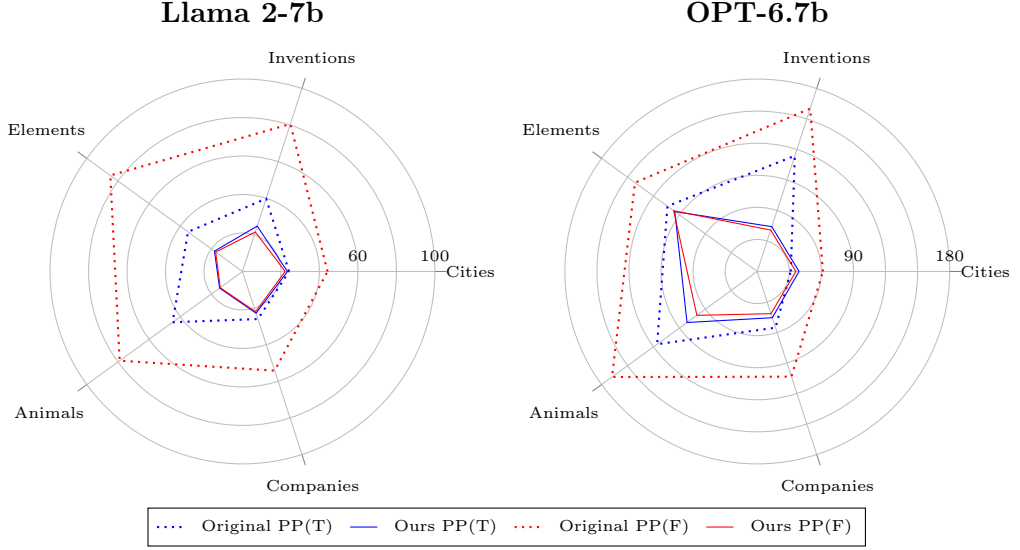


Figure 4.1: Comparison of average perplexity scores for Llama 2-7b and OPT-6.7b for the Original dataset by Azaria and Mitchell (2023) and our refined version. Lower perplexity indicates that the sentences are more likely to have been generated by the model. PP(T/F) denotes average perplexity of true/false sentences.

answer, the model receives a continuation prompt and generates a completion following the established format. The resulting outputs are subsequently filtered for both length and uniqueness to ensure concise, self-consistent, and high-quality responses. The generation process uses the following hyperparameters for all the employed models (i.e., Llama 2-7b and OPT-6.7b): `max_new_tokens=128`; `top_k=50`; `top_p=0.95`; `temperature=1.0`; `num_return_sequences=10`.

Figure 4.3 illustrates an example prompt to **GPT-4o mini** (2024-08-01), the LLM that we use as the oracle for the procedure described in Section 4.2.3. In this setup, we employ a 3-shot learning strategy, where the model is provided with three examples to establish a clear pattern of judgment. These examples demonstrate the task of evaluating answers against a given ground truth. The model is then tasked with using this context to evaluate a new answer to a question, based on the same criteria. In our experiments, we employ a temperature setting of 1.

Table 4.9 illustrates the number of annotated facts extracted from TriviaQA (Joshi et al., 2017) employing the procedure described in section 4.2.3. Statistics are presented for each of the 14 question sources. Table 4.10 reports examples of generated facts for the Llama 2-7b and OPT-6.7b models.

Few-Shot Text Generation Prompt for TriviaQA	
Question: Where in England was Dame Judi Dench born?	Answer: The English actress Dame Judi Dench was born in York, England.
Question: From which country did Angola achieve independence in 1975?	Answer: Angola achieved independence from Portugal in 1975.
Question: Which city does David Soul come from?	Answer: David Soul hails from Chicago, Illinois.
Question: Who won Super Bowl XX?	Answer: The Chicago Bears won Super Bowl XX.
Question: Which was the first European country to abolish capital punishment?	Answer: Norway was the first European country to abolish capital punishment.
Question: In which country did the widespread use of ISDN begin in 1988?	Answer: The widespread use of ISDN began in Japan in 1988.
Question: What is Bruce Willis' real first name?	Answer: Bruce Willis' real first name is Walter.
Question: Which William wrote the novel Lord of the Flies?	Answer: The William who wrote Lord of the Flies was William Golding.
Question: How is Joan Molinsky better known?	Answer: Joan Molinsky is better known as Joan Rivers.
Question: In which branch of the arts is Patricia Neary famous?	Answer: Patricia Neary is famous in the field of ballet.
Question: Ford Prefect came from a star in which constellation?	Answer:"

Figure 4.2: Text generation pipeline for TriviaQA (Joshi et al., 2017). The 10 examples are prepended to the actual question, after which the model is prompted to generate an answer. The answers for the examples are manually crafted based on the available ground truth.

We construct a True-False dataset from two additional QA datasets, SQuAD 2.0 (Rajpurkar et al., 2018) and TruthfulQA (Lin et al., 2022), employing the same approach. Following the same procedure detailed in Section 4.2.3, we extract true and false facts from TruthfulQA. Table 4.7 reports the statistics of the LLM-generated datasets constructed with Llama 2-7b and OPT-6.7b.

For the SQuAD 2.0 dataset, we limit our dataset construction procedure (Section 4.2.3) to the validation split and the five most common topics, as the

Example 3-Shot Prompt for LLM-Based Answer Evaluation

System: "You are a judge and your role is to judge whether the provided answer is correct for the given question, based on the provided ground truth. Answer with a 1 if the answer is correct and 0 if the answer is incorrect. Here are a few examples:"

User: "Question: Who was the next British Prime Minister after Arthur Balfour?
 Answer: Arthur Balfour was followed by David Lloyd George.
 Ground truth: {'aliases': [..., 'Henry Campbell-Bannerman'], 'normalized_value': 'campbell bannerman', 'value': 'Campbell-Bannerman'}
 Evaluation:

Assistant: "0"

User: "Question: Who had a 70s No 1 hit with Kiss You All Over?
 Answer: The band Exile had a 70s No 1 hit with Kiss You All Over.
 Ground truth: {'aliases': [..., 'Exile'], 'normalized_value': 'exile', 'value': 'Exile'}
 Evaluation:"

Assistant: "1"

User: "Question: Which common mineral is used to make casts, moulds, blackboard chalk and plaster of Paris?
 Answer: The common mineral used to make casts, moulds, blackboard chalk and plaster of Paris is calcium carbonate.

Ground truth: {'aliases': [..., 'Gypsum'], 'normalized_value': 'gypsum', 'value': 'Gypsum'}

Evaluation:"

Assistant: "0"

Figure 4.3: Example prompt provided to GPT-4o mini. This prompt implements a 3-shot learning strategy, supplying three examples that establish the expected judgment pattern for assessing answer correctness against a given ground truth. Following these examples, the model is instructed to evaluate a new question-answer pair using the same criteria.

large number of questions makes applying our procedure to the full dataset computationally impractical due to time constraints. Table 4.14 presents the performance of the SAPLMA classifier when trained and tested on the SQuAD 2.0 dataset.

Impact of Perplexity-Based Sampling on SAPLMA Accuracy

We assess the impact of our perplexity-based sampling strategy by training the SAPLMA probe classifier separately on the original True-False dataset from Azaria and Mitchell (2023) and our refined version. As shown in Table 4.11, classifiers tested on our refined dataset achieve the highest accuracy in deeper layers. For Llama 2-7b, Layer 16 shows the most consistent results, achieving the highest

Table 4.7: Summary of the dataset obtained by extracting factoid sentences from TruthfulQA (Lin et al., 2022), following the procedure described in Section 4.2.3.

Model	Total Samples	True Samples (%)
Llama 2-7b	842	49.76%
OPT-6.7b	331	49.85%

Table 4.8: Summary of the dataset obtained by extracting factoid sentences from SQuAD 2.0 (Rajpurkar et al., 2018), following the procedure described in Section 4.2.3.

Topic	Model	Sentences	(%) True
Economic_inequality	Llama 2-7b	224	50.45%
	OPT-6.7b	135	50.37%
Immune_system	Llama 2-7b	167	49.70%
	OPT-6.7b	72	50.00%
Rhine	Llama 2-7b	120	50.83%
	OPT-6.7b	59	47.46%
Warsaw	Llama 2-7b	140	51.43%
	OPT-6.7b	102	50.00%
Yuan Dynasty	Llama 2-7b	109	49.54%
	OPT-6.7b	53	52.83%

average accuracy across all tested configurations when trained on our dataset. Although accuracy fluctuates across different topics, overall performance remains largely consistent between the training configurations. Comparing the results with the original results in Table 4.2, it emerges that classifiers trained on the original dataset achieve higher accuracy when tested on the same dataset. This is particularly evident in the ‘Animals’ topic. However, it is important to consider that the perplexity values in our refined dataset are significantly lower than those in the original dataset for both true and false statements (see 4.1). Additionally, our refined dataset features lower and more closely aligned perplexity scores between true and false statements, with false statements often yielding even lower perplexity values than true ones. This finding implies the validity of Azaria and Mitchell hypothesis, which states that **factuality information is encoded in LLM hidden states**, although it might not be immediately evident from the model predicted probabilities.

RQ1: *Are probing classifiers trained as in Azaria and Mitchell (2023) capable of generalizing to True-False sentences with similar perplexity?*

Table 4.9: Summary of the dataset obtained by extracting factoid sentences from TriviaQA (Joshi et al., 2017) Wikipedia validation split, following the procedure described in Section 4.2.3.

Dataset	Llama 2-7b		OPT-6.7b	
	Sentences	(%) True	Sentences	(%) True
triviacountry	118	49.15%	118	49.15%
wwwodquiz	700	49.71%	525	49.71%
triviabug	97	50.52%	88	51.14%
derby*	342	50.00%	325	50.00%
quiz-zone	187	48.13%	239	50.63%
businessballs	433	50.84%	443	49.43%
wrexham**	216	50.00%	276	50.72%
sfquiz	1,054	49.80%	1,085	50.39%
quizwise	565	49.91%	704	50.42%
billturnbull	216	49.54%	105	50.48%
jetpunk	139	51.80%	364	50.27%
quizballs	500	49.60%	488	50.82%
quizguy	240	51.67%	341	50.74%
quiz4free	171	49.12%	141	48.94%

*: *derby is adopted as abbreviation of derbyshirepubquizleague*

** : *wrexham is adopted as abbreviation of wrexhamquizleague*

Surprisingly, the results between the probes trained on our refined dataset and the probes trained on the original dataset are mostly comparable. This proves that **the probes can generalize even when the train-test datasets have different perplexity**.

Generalization of SAPLMA on LLM-generated Sentences

We extend our experiments with the SAPLMA classifier to a new setting, where both training and evaluation are conducted on sentences generated by an LLM and sourced from TriviaQA, leveraging the procedure described in Section 4.2.3. Table 4.12 reports the performance of SAPLMA on a set of factual statements extracted from TriviaQA. The results indicate that the currently used probes are inadequate for factuality self-assessment in real-world scenarios, as the observed accuracy does not reach a noteworthy threshold. Furthermore, following the suggestion of Azaria and Mitchell, we optimize the classification threshold; however, this adjustment yields no significant improvement in performance.

We extended our LLM-generated dataset construction approach (Section 4.2.3)

Table 4.10: Examples of true and false facts generated by Llama 2-7b and OPT-6.7b based on the questions in TriviaQA (Joshi et al., 2017), obtained following the procedure detailed in Section 4.2.3.

Model	Label	Sentence
Llama 2-7b	1	Arthur was married to Guinevere.
	1	Arthur’s most famous wife was Guinevere.
	1	Guinevere was married to Arthur.
	1	Guinevere’s husband was King Arthur.
	0	Lancelot was married to Guinevere.
	0	Sir Lancelot was married to Queen Guinevere.
	0	Sir Leonne was married to Queen Guinevere.
OPT-6.7b	0	Anakin Skywalker is DARTH Vader’s son.
	0	DARTH Vader’s son is DARTH Vader.
	1	DARTH Vader’s son is Luke Skywalker, a member of the Rebel Alliance.
	1	DARTH Vader’s son is Luke Skywalker.
	1	DARTH Vader’s son is Luke.
	0	DARTH Vader’s son is known as DARTH Vader.
	1	Luke’s father is DARTH Vader.

to TruthfulQA (Lin et al., 2022) and SQuAD 2.0 (Rajpurkar et al., 2018) to further support our findings.

Table 4.7 reports the statistics of the LLM-generated datasets constructed with Llama 2-7b and OPT-6.7b. Table 4.13 presents the performance of the SAPLMA classifier when both trained and tested on the TruthfulQA dataset. Since the dataset lacks a topic-based split, we adopt a 50-50 holdout strategy for evaluation.

The results on TruthfulQA and SQuAD 2.0 further support our conclusion: existing probes fall short in accurately assessing factuality in real-world settings. Across all cases, accuracy remains below a meaningful threshold, highlighting the limited reliability of these methods for self-assessment in scenarios involving LLM-generated sentences.

RQ2: *Can the same probes generalize to facts generated by LLMs?*

In summary, this experiment partially contradicts the findings of Azaria and Mitchell (2023): **the trained probes are not capable of providing good generalization to an LLM-generated dataset**, even when the accuracy threshold is tuned. The motivation could stem from the nature of the dataset, as TriviaQA contains open-domain questions that result in more nuanced facts than the ones in the original True-False dataset (Section 4.2.1) or the OPT-generated dataset (Section 4.2.2).

We believe that further research is needed to enhance the effectiveness of factu-

Table 4.11: Accuracy values obtained training SAPLMA on the original True-False dataset and on our refined version, then **tested on our refined version**. Results are shown for the Llama2 and OPT models. ‘Orig.’ denotes the ‘original True-False dataset as training data, while ‘Novel’ denotes our version of the True-False dataset as training data. In **bold** we denote the best combination of layer/training dataset for each combination of model/topic.

Layer	Training Data	Cities		Inventions		Elements		Animals		Companies		Average	
		Llama2	OPT	Llama2	OPT	Llama2	OPT	Llama2	OPT	Llama2	OPT	Llama2	OPT
last	Orig.	.6882	.5724	.6409	.5094	.6314	.5482	.5685	.5259	.6290	.6984	.6316	.5709
	Novel	.6365	.6143	.6101	.5144	.5623	.5293	.5461	.4846	.7414	.7211	.6311	.5720
28	Orig.	.7056	.5870	.7001	.5178	.6161	.5832	.6013	.5566	.7079	.7234	.6662	.5936
	Novel	.5091	.6057	.6591	.5473	.5665	.5655	.6052	.4662	.7061	.7065	.6092	.5811
24	Orig.	.8286	.7026	.7250	.6022	.6432	.5918	.6310	.5439	.7121	.7366	.7080	.6354
	Novel	.6025	.6710	.6609	.6248	.5763	.5993	.5836	.4868	.7868	.7366	.6420	.6225
20	Orig.	.8272	.7313	.7741	.6230	.6492	.6255	.5832	.5075	.7583	.7502	.7184	.6475
	Novel	.7382	.7528	.6973	.6131	.6051	.5986	.6190	.4807	.8270	.7566	.6973	.6404
16	Orig.	.8941	.6433	.7888	.5698	.6801	.5930	.5836	.3816	.7768	.7426	.7447	.5860
	Novel	.9301	.7505	.7961	.5644	.6623	.5568	.6319	.5307	.8265	.7231	.7694	.6151

ality self-assessment techniques, particularly in settings involving LLM-generated content. Promising research directions may be leveraging datasets that are closely aligned with the distribution of LLM-generated text and exploring alternative techniques such as uncertainty-aware classification.

4.2.5 Limitations

There are a few aspects of our study that could be explored further. We made every effort to check the datasets for inconsistencies, but a more thorough manual verification by human annotators would be beneficial for ensuring their robustness and minimizing potential biases. Additionally, our analysis is based on a limited set of models. While these models provide valuable insights, it is possible that larger or more complex models could demonstrate enhanced performance, particularly in the context of self-evaluation. Future work could expand on this, incorporating a wider range of models to investigate whether scalability can improve results.

Lastly, we highlight important considerations regarding the use of perplexity as a proxy for plausibility in relation to model knowledge. Recent findings suggest that LLMs often assign similar perplexity scores to both seen and unseen sentences (Duan et al., 2024), which may limit the reliability of perplexity in distinguishing between plausible and implausible content. This challenge intersects with broader issues of memorization and data provenance. While our work

Table 4.12: Performance of SAPLMA on a fact dataset generated from TriviaQA. The original topic-wise leave-one-out strategy is adopted. Results are shown for the Llama 2-7b and OPT-6.7b models.

Dataset	Model	Threshold = 0.5					Optimal Threshold				
		last	28	24	20	16	last	28	24	20	16
billturnbull	Llama	.561	.576	.618	.621	.628	.547	.581	.618	.634	.639
	OPT	.547	.558	.591	.551	.537	.530	.558	.547	.530	.499
derby*	Llama	.568	.581	.575	.596	.617	.560	.565	.559	.571	.597
	OPT	.553	.564	.584	.587	.572	.562	.566	.583	.586	.580
quiz4free	Llama	.564	.547	.523	.561	.589	.548	.523	.521	.547	.573
	OPT	.559	.559	.575	.581	.560	.544	.530	.533	.546	.541
quizguy	Llama	.578	.585	.588	.607	.635	.576	.587	.601	.595	.637
	OPT	.579	.583	.589	.590	.584	.559	.555	.565	.559	.548
triviabug	Llama	.494	.518	.521	.525	.538	.508	.500	.530	.542	.545
	OPT	.620	.624	.607	.596	.528	.632	.635	.605	.598	.553
businessballs	Llama	.566	.558	.565	.574	.582	.558	.551	.555	.564	.575
	OPT	.559	.558	.578	.570	.553	.545	.547	.573	.562	.551
jetpunk	Llama	.587	.627	.620	.643	.654	.543	.580	.550	.554	.601
	OPT	.606	.612	.614	.596	.621	.618	.618	.621	.605	.619
odquiz	Llama	.551	.536	.546	.562	.573	.537	.525	.532	.550	.560
	OPT	.560	.573	.583	.583	.542	.551	.566	.571	.569	.521
quiz-zone	Llama	.556	.557	.558	.565	.611	.541	.549	.555	.569	.615
	OPT	.569	.570	.582	.592	.552	.537	.550	.535	.580	.508
quizballs	Llama	.603	.575	.572	.578	.571	.602	.561	.561	.570	.553
	OPT	.558	.565	.574	.571	.540	.550	.549	.564	.564	.539
quizwise	Llama	.560	.565	.579	.609	.618	.563	.551	.574	.605	.619
	OPT	.560	.563	.565	.577	.540	.555	.556	.552	.575	.537
sfquiz	Llama	.568	.554	.554	.559	.575	.564	.554	.559	.566	.583
	OPT	.584	.591	.589	.590	.547	.570	.582	.577	.573	.536
triviacountry	Llama	.536	.554	.559	.556	.566	.506	.530	.536	.524	.549
	OPT	.536	.551	.550	.587	.534	.449	.476	.472	.511	.472
wrexham**	Llama	.570	.563	.569	.553	.565	.567	.548	.561	.545	.566
	OPT	.548	.573	.578	.584	.554	.535	.552	.574	.579	.571
Average	Llama	.562	.564	.568	.579	.594	.552	.550	.558	.567	.587
	OPT	.567	.575	.583	.583	.555	.553	.560	.562	.567	.587

*: derby is adopted as abbreviation of derbyshirepubquizleague

** : wrexham is adopted as abbreviation of wrexhamquizleague

Table 4.13: Performance of SAPLMA on a fact dataset generated from TruthfulQA. A 50-50 holdout strategy is adopted. Results are shown for the Llama 2-7b and OPT-6.7b models.

Layer	OPT-6.7b		Llama 2-7b	
	Random Split 1	Random Split 2	Random Split 1	Random Split 2
last	.487	.488	.491	.504
28	.477	.495	.498	.506
24	.478	.513	.507	.507
20	.512	.535	.518	.509
16	.493	.492	.520	.514

Table 4.14: Performance of SAPLMA on a fact dataset generated from SQuAD 2.0, restricted to the validation set and the top-5 most popular topics. The original topic-wise leave-one-out strategy is adopted. Results are shown for the Llama 2-7b and OPT-6.7b models.

Dataset		last	28	24	20	16
Economic inequality	Llama	0.525	0.510	0.515	0.492	0.502
	OPT	0.517	0.516	0.499	0.496	0.464
Immune system	Llama	0.486	0.490	0.503	0.502	0.547
	OPT	0.436	0.439	0.401	0.438	0.435
Rhine	Llama	0.525	0.520	0.515	0.524	0.526
	OPT	0.484	0.485	0.522	0.547	0.521
Warsaw	Llama	0.558	0.542	0.545	0.546	0.555
	OPT	0.521	0.531	0.545	0.544	0.509
Yuan dynasty	Llama	0.509	0.522	0.514	0.529	0.544
	OPT	0.544	0.560	0.512	0.542	0.512

focuses on extending and refining prior approaches to factuality self-evaluation, we recognize that a rigorous assessment of perplexity effectiveness in this context is a critical avenue for future investigation. We therefore encourage further research into more robust and theoretically grounded measures of plausibility for LLM-generated content.

4.2.6 Summary

In this work, we investigate the factuality-encoding capabilities of LLMs. Our work replicates the methodology of Azaria and Mitchell (2023) to ensure reproducibility and extend their approach with two novel dataset construction strategies: perplexity-based negative sampling and fact generation based on QA datasets. We applied these strategies to analyze two open-source LLMs, and found that although the findings from previous studies are partially validated even on more challenging synthetic datasets, transferring these findings to LLM-generated datasets proves difficult. This study paves the way for more reliable LLM evaluations and offers practical guidelines for improving model transparency and trustworthiness in real-world applications. Our codebase is available on GitHub ².

²<https://github.com/sisinflab/HidingInTheHiddenStates>

4.3 Related Work

Probing techniques have become central for the layer-wise interpretation of deep learning models (Alain and Bengio, 2017). This approach was then extended to Large Language Models (LLMs) to assess LLMs ability to encode syntax and semantics (Conneau et al., 2018; Tenney et al., 2019). Specifically, among the different applications, a technique that emerges is **self-evaluation**, defined as *a model’s ability to assess the accuracy of its own outputs* (Kadavath et al., 2022). Among the various works, notable research by Kadavath et al. (2022) explored estimating a well-calibrated $P(\text{True})$ directly from output probabilities to reflect answer accuracy. Azaria and Mitchell (2023) demonstrate that LLMs are capable of detecting false claims in synthetic true-false datasets. Building upon this initial study, several works (Marks and Tegmark, 2023; Bürger et al., 2024; Levinstein and Herrmann, 2023) investigate the generalization capabilities of probes in detecting hallucinations in LLMs. While these studies offer valuable insights into how factual knowledge is encoded in LLMs, they stop short of examining whether LLMs can assess the factuality of their own generations. Instead, they primarily evaluate models on artificially constructed or perturbed datasets, rather than on claims produced by the LLMs themselves. Orgad et al. (2025) explore error detection and hallucination mitigation in Question Answering by probing the internal representations of question-answer pairs. Their study shows that while LLMs encode factuality signals, these signals do not generalize well across task-specific datasets, suggesting that factuality encoding is task-dependent rather than universal. Gekhman et al. (2025) show that LLMs can internally encode knowledge, detectable by probing the hidden states, yet still fail to express it in their generated output. Chen et al. (2024a) introduce the INSIDE framework, which detects hallucinations using the EigenScore metric to measure self-consistency across multiple LLM outputs for a single input. Similarly, Zhang et al. (2024c) leverage a probe model, PINOSE, trained via offline consistency-checking, to perform online hallucination detection. Although INSIDE and PINOSE primarily focus on self-consistency in internal representations across multiple generated responses, they do not interpret hidden states or look for explicit factuality encoding.

4.4 Overview: Choice of Representation

This section provides a consolidated overview of the internal representations of PLMs examined throughout this chapter. Its purpose is primarily expository: rather than introducing new methodological elements, it summarizes the representation access points used in the factuality analyses throughout the chapter. By clarifying which internal signals are extracted from frozen models and how they are analyzed, this section aims to support readers less familiar with probing-based or representation-centric approaches.

A central assumption underlying this chapter is that factuality-related information about generated utterances is reflected in the internal activations of the language models that produce them. Accordingly, factuality assessment is framed as a probing task: given an input statement, the model’s hidden representations are inspected to determine whether they encode sufficient information to distinguish between factually correct and incorrect content. This paradigm treats the language model as a fixed computational artifact, separating the analysis of internal knowledge encoding from generation and decoding behavior.

Figure 4.4 schematically summarizes the representation extraction process adopted in Section 4.2, following the SAPLMA probe (Azaria and Mitchell, 2023). Given an input statement, the language model processes the sequence autoregressively, producing a series of token-level hidden states at each transformer layer. These representations can be interpreted as progressively refined encodings of the input, reflecting different stages of contextual integration and abstraction within the model.

Rather than focusing exclusively on the final layer, the SAPLMA framework explicitly examines representations extracted from multiple depths of the transformer stack. Concretely, a single token-level hidden state corresponding to the final input token is selected and extracted from the 16th, 20th, 24th, 28th, and final transformer layers. This selection spans both intermediate and late layers, enabling an analysis of how factuality-related information evolves across the network. The same extraction strategy is applied consistently across the two models considered in this study, namely Llama 2-7B and OPT-6.7B, both of which consist of 32 transformer layers.

The extracted layer-wise representations are used as input features for a lightweight classifier trained to predict whether a statement is factually correct or incorrect. By decoupling the classifier from the language model, this setup

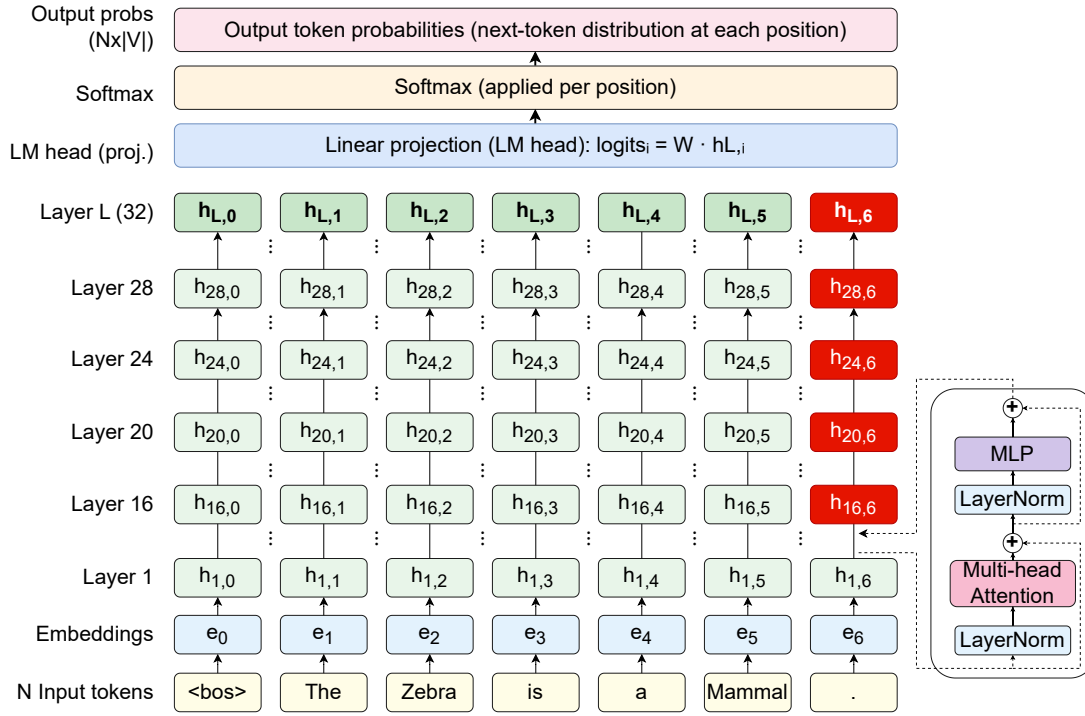


Figure 4.4: Overview of the representations extraction process adopted in Section 4.2. Hidden states corresponding to the final input token are extracted from multiple transformer layers of a frozen LLM and used as input features for factuality probing.

ensures that any predictive signal must already be present in the model’s internal activations, rather than arising from task-specific fine-tuning. This design choice enables a focused analysis of the representational properties of the underlying language models.

Examining multiple layers is particularly important for factuality analysis, as different layers are known to encode different types of information. While earlier and intermediate layers tend to capture lexical, syntactic, and relational patterns, later layers are more closely aligned with the next-token prediction objective. Comparing probing performance across layers therefore allows the analysis to assess whether factuality-related information is uniformly distributed or concentrated at specific depths. As shown in Section 4.2, this layer-wise perspective reveals systematic differences in factuality encoding, with intermediate layers often exhibiting stronger and more stable signals than the final layer.

Chapter 5

Moving Beyond: Hidden Capabilities in PLMs

So far, we have considered pretrained language models (PLMs) primarily as repositories of world knowledge, capable of recalling facts, definitions, and structured information embedded in their training data, and even conveying factual uncertainty about their generated utterances according to their prior knowledge.

However the potential of PLMs is not limited to their memorization capabilities and their potential to serve as dense knowledge repositories. In fact, recent research has revealed that these models possess capabilities that go far beyond simple memorization, driven by the relentless scaling of model size, dataset volume, and computational resources. Foundational work on neural scaling laws established that as these factors increase, a model’s performance on its core pre-training objective improves in a predictable, power-law fashion. This predictable improvement has given rise to unpredictable and powerful secondary capabilities, termed "emergent abilities" (Wei et al., 2022): skills that are not present in smaller-scale models but appear, often sharply, once a model surpasses a certain scale threshold. In particular, large-scale PLMs have demonstrated the surprising ability to perform complex reasoning, classification, and generation tasks without any task-specific supervision, behaving as powerful few-shot or zero-shot learners. The seminal work on GPT-3 (Brown et al., 2020) provided a watershed demonstration of this phenomenon, revealing that a sufficiently large model could be steered simply by providing a natural language description of a task (the zero-shot setting) or a handful of examples within the prompt itself (the few-shot setting), a capability now known as *in-context learning*.

The conventional pre-train/fine-tune paradigm presents significant limitations that challenge the scalable and widespread deployment of LLMs. A primary constraint is the resource-intensive nature of this approach. The necessity of fine-tuning a unique, multi-billion parameter model for every new task imposes substantial computational costs, limiting scalability. Moreover, large-scale, high-quality labeled datasets for effective training are often not available to many organizations, thus limiting the accessibility of model specialization.

Beyond the prohibitive resource requirements, the fine-tuning process itself risks degrading the model's core capabilities. This is exemplified by the phenomenon of "catastrophic forgetting," where a model's general linguistic knowledge erodes as it adapts to a narrow task. A related concern is "format specialization," whereby a model overfits to a specific output format (e.g., binary classification) and, in doing so, loses its crucial ability to generate flexible, open-ended text, effectively compromising its general-purpose utility. These limitations are particularly acute in dynamic commercial applications, such as e-commerce search and recommendation, where the constantly evolving nature of product catalogs and user interactions makes the continuous fine-tuning of models an especially demanding endeavor. Therefore, to facilitate the pervasive integration of LLMs in real-world applications, a paradigm shift beyond task-specific fine-tuning is not merely advantageous, but necessary.

Building on these insights, this chapter examines the potential of frozen pre-trained language models (PLMs) (i.e., models applied directly without additional weight updates or task-specific training) for real-world, task-oriented, applications. Instead of adapting models to each downstream task, we explore how the intrinsic generalization abilities of pretrained models can be leveraged "as is," drastically reducing the cost and complexity of deployment. We highlight scenarios where efficiency, adaptability, and robustness are especially valuable, such as sentiment analysis, recommender systems, and information retrieval. These examples illustrate how frozen PLMs can serve as versatile, plug-and-play components, offering substantial practical benefits while avoiding the overhead of extensive retraining, thus leading the way to a more sustainable employment of pretrained language models (PLMs) and large language models (LLMs) in the near future.

Contributions. In this chapter, we discuss our research contributions revolving around emergent capabilities in hidden LLMs representations, which can be summarized as follows:

1. Section 5.1: We analyze the latent sentiment and emotion representations encoded in frozen PLMs, focusing on where and how these features are distributed across model layers. Through layer-wise probing of Llama models, we identify that sentiment information is most concentrated in the mid-layers, with specific pooling strategies enhancing detection accuracy by up to 14% compared to prompting-based approaches. Furthermore, we show that sentiment encoding is not confined to the final token in decoder-only architectures, revealing a more distributed representation of affective signals. Finally, we demonstrate that models can be substantially compressed, achieving average memory savings of 57%, while maintaining strong performance on sentiment and emotion recognition tasks. These findings highlight layer-specific probing as an efficient and interpretable alternative to prompting for sentiment-related applications.
2. Section 5.2: We investigate the use of GTEs for recommendation and search tasks in a fully zero-shot setting, eliminating the need for task- or domain-specific fine-tuning. Our findings challenge the prevailing assumption that specialized PLMs are essential for high performance in e-commerce applications. Through comprehensive evaluations on established benchmarks, we show that generalist embedding models, pre-trained on large and diverse corpora, consistently achieve competitive or superior results in both sequential recommendation and product search, outperforming models explicitly optimized for these tasks. We attribute this advantage to their more balanced and expressive representation of semantic features across the embedding space. Moreover, we demonstrate that applying dimensionality reduction techniques such as PCA can further enhance specialized models by filtering noise and emphasizing the most informative representational directions. Overall, these results position GTEs as efficient, robust, and domain-agnostic alternatives for text-driven recommendation and search systems.

5.1 *LLaMAs Have Feelings Too: Unveiling Sentiment and Emotion Representations in LLaMA Models Through Probing*¹

Sentiment analysis (SA), which classifies opinions, emotions, and attitudes in text, is a cornerstone of natural language processing (NLP). SA has enabled the development of several applications, including social media monitoring (Camacho-Collados et al., 2022), customer feedback analysis (Lin et al., 2017a), and opinion mining (Xia et al., 2021). It comprises various tasks, such as polarity detection (classifying text as positive, negative, or neutral), emotion classification, and subjectivity detection (Khan et al., 2014).

Earlier approaches for the sentiment classification task relied on supervised learning algorithms and extensive feature engineering (Al-Mannai et al., 2014), requiring large annotated datasets and domain expertise to model sentiment effectively (Bordoloi and Biswas, 2023).

However, the advent of pre-trained models such as GPT-2 (Radford et al., 2019) has transformed NLP by enabling downstream tasks through prompting techniques (Brown et al., 2020; Di Palma, 2023), significantly reducing reliance on extensive labeled data.

While prompting has enhanced the applicability of LLMs for sentiment analysis (Deng et al., 2023; Xing, 2025; Ahmed et al., 2024), these techniques have often lacked fine-grained control of semantic features, such as context-dependent sentiments or subtle emotional tones. Moreover, the encoding of such features within model representations has not been well understood, limiting efforts to optimize and interpret sentiment analysis performance.

Efforts to interpret neural model representations have advanced significantly over time. Building on the linear representations hypothesis (Mikolov et al., 2013b), which suggests that high-level features can be represented as linear directions, researchers have explored where human-interpretable concepts are encoded within LLMs. While previous studies have examined concepts such as truthfulness (Burns et al., 2023), honesty (Azaria and Mitchell, 2023), and factual knowledge (Li

¹Based on our paper, “LLaMAs Have Feelings Too: Unveiling Sentiment and Emotion Representations in LLaMA Models Through Probing”, in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pp. 6124–6142 (2025).

et al., 2023b), sentiment encoding remains relatively underexplored, despite its importance in NLP applications.

This study aims to fill this gap by investigating where sentiment information is encoded within Llama models (Touvron et al., 2023a). We focus on both binary polarity (positive or negative sentiment) and fine-grained emotion detection (joy, sadness, anger, fear, love, and surprise), identifying the model layers that most effectively capture these sentiment concepts. Specifically, we trained classifiers to identify subspaces corresponding to specific concepts, such as positive sentiment, within model representations. We refer to these classifiers as probes (Alain and Bengio, 2017).

Furthermore, unlike previous studies that rely solely on the last token’s representation (as a sentence summary) for probe training (Burns et al., 2023; Azaria and Mitchell, 2023; Li et al., 2023b), we evaluate alternative pooling methods for sentence-level sentiment detection.

Our extensive experiments reveal several key insights into how sentiment and emotion are represented in LLMs. We find that (1) sentiment concepts are most detectable in mid-layer representations, while (2) emotions are more discoverable in early layers; (3) selecting the last token does not consistently yield the best results for probe training; (4) representation quality improves with model size; and (5) probe classifiers generally outperform prompting techniques for accurate sentiment and emotion classification.

In summary, our work makes the following contributions:

1. We perform a layer-wise analysis of sentiment encoding in Llama models, identifying the layers that most effectively capture sentiment information.
2. We evaluate six pooling methods to identify the most effective approach for sentence representation in sentiment analysis.
3. We investigate the impact of model size on sentiment representation by analyzing Llama-3 models of 1B, 3B, and 8B parameters.
4. We show that Llama-based classifiers outperform Llama in zero-shot, few-shot, and Chain-of-Thought settings while requiring fewer parameters.
5. We propose SENTRILLAMA, a task-specific adaptation of Llama that identifies and utilizes the most representative layer for sentiment analysis, replacing

the language modeling head with a lightweight classification head to significantly reduce inference costs while maintaining state-of-the-art performance.

5.1.1 Methodology

In this section, we detail the models used in the experiments, the datasets, the probe classifiers, and the experimental settings, with a focus on ensuring the reproducibility of our work.

Sentiment Detection in Hidden Space. We build on the concept detection framework proposed by Rütte et al. (2024), adapting it specifically for sentiment analysis. In this setup, we define a sentiment concept S and use a corresponding sentiment analysis dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where y_i represents the labeled sentiment of sentence x_i . Here, $\text{rep}_\theta(x_i)$ represents an intermediate representation of x_i , generated from a forward pass through the LLM_θ . The goal is to extract a collection of these representations $\{\text{rep}_\theta(x_i)\}_{i=1}^n$. We therefore train a classifier C_w on these representations to predict the presence of the sentiment concept S (i.e., positive or negative sentiment within x_i), effectively predicting y_i .

Implementing sentiment detection involves two fundamental design choices: (1) The selection of the intermediate representation rep_θ , which may vary depending on the layers or pooling strategies applied within the LLM. (2) The choice of classifier C_w , which serves to distinguish between different sentiment categories based on these embeddings.

In the following, we outline the fundamental structure of Transformer architecture and highlight common techniques for selecting representations.

Choice of Representation. Current state-of-the-art LLMs are based on the Transformer architecture (Vaswani et al., 2017), where sequential Transformer blocks generate intermediate hidden representations (h), each with potentially distinct functionalities. Let $l \in \mathbb{N}$ denote the l -th layer, and $\mathbf{x}^{(l)} \in \mathbb{R}^{T \times d_{\text{emb}}}$ represent the output, where T is the number of tokens and d_{emb} is the hidden dimension. A Transformer refines $\mathbf{x}^{(l)}$ using multi-head attention (MHA) and a feed-forward network (FFN):

$$h_{\text{attn}}^{(l)} = \text{MHA} \left(\text{LayerNorm} \left(x^{(l)} \right) \right),$$

$$\begin{aligned}
h_{\text{resid}}^{(l)} &= h_{\text{attn}}^{(l)} + x^{(l)}, \\
h_{\text{ffn}}^{(l)} &= \text{FFN} \left(\text{LayerNorm} \left(h_{\text{resid}}^{(l)} \right) \right), \\
x^{(l+1)} &= h_{\text{ffn}}^{(l)} + h_{\text{resid}}^{(l)}.
\end{aligned}$$

We exploit the Llama-3 architecture (Dubey et al., 2024a), leveraging its Grouped Query Attention (GQA) and RMSNorm features to extract hidden representations efficiently.

Previous studies have explored various representations, such as the residual stream ($\mathbf{x}^{(l+1)}$) (Marks and Tegmark, 2023; Burns et al., 2023; Zou et al., 2023; Gurnee and Tegmark, 2024), the normalized residual stream (nostalgebraist, 2020), or attention heads (Li et al., 2023b; Arditì and Obeso, 2023). Based on our preliminary experiments showing marginally higher detection accuracy, we adopt the residuals stream ($\text{rep}_\theta(\mathbf{x}) = \mathbf{x}^{(l+1)}$).

Instead of using the full prompt representation ($\mathbf{x}_{\text{rep}} \in \mathbb{R}^{T \times d_{\text{emb}}}$, where T is the token count), we focus on a subset ($\mathbf{x}_{\text{rep}} \in \mathbb{R}^{t \times d_{\text{emb}}}$) with $t \leq T$. Each token representation ($\mathbf{x}_{\text{rep}}[i, :] \in \mathbb{R}^{d_{\text{emb}}}$, for $i = 1, \dots, t$), is treated as an independent feature. This approach focuses on the parts of the prompt most likely to capture the sentiment concept. Prior work has carefully selected a single token (Arditì and Obeso, 2023; Zou et al., 2023; Gurnee and Tegmark, 2024) or relied on the last token of the prompt (Rimsky, 2023; Mallen and Belrose, 2023; Marks and Tegmark, 2023; Li et al., 2023b; Burns et al., 2023).

In our experiments, we explore six methods for selecting the representations:

- (1) Mean Pooling (Fig. 5.1): Compute the mean activation value across all tokens for each d_{embed} dimension, resulting in a single vector where each element corresponds to the average activation of a particular feature (embedding dimension) over the entire sequence.
- (2) Last-Token Pooling (Fig. 5.2): Uses the final token’s features of the last token in the sequence.
- (3) Max Pooling (Fig. 5.3): Identifies the most prominent feature across all tokens T for each feature, outputting a vector representing the most dominant features in the sequence.
- (4) Min Pooling (Fig. 5.3): Complementary to Max Pooling, producing a vector representing the least dominant features in the sequence.

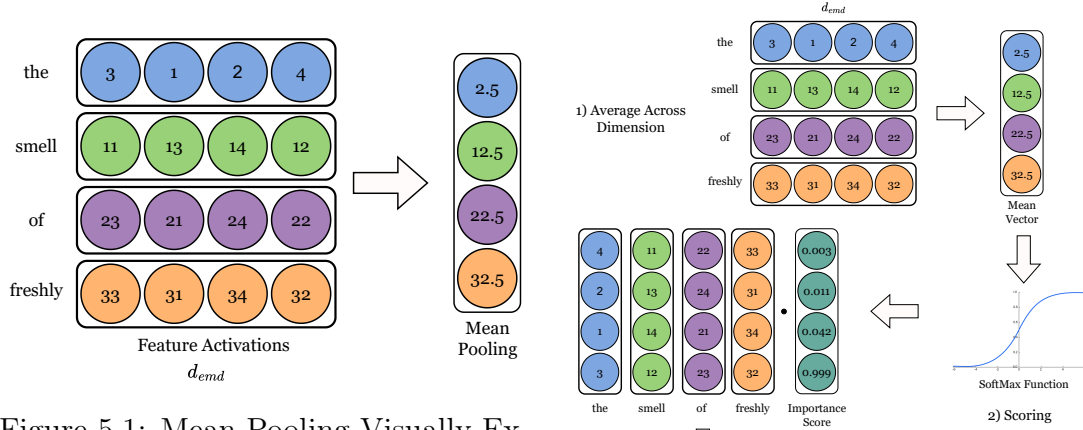


Figure 5.1: Mean Pooling Visually Explained

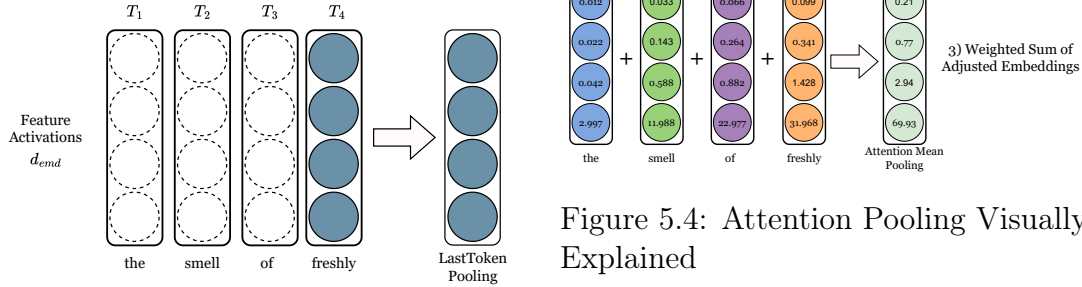


Figure 5.4: Attention Pooling Visually Explained

Figure 5.2: Last Token Pooling Visually Explained

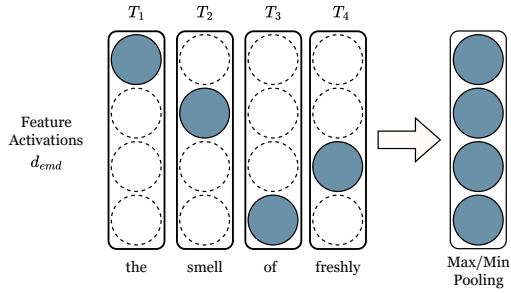


Figure 5.3: Max and Min Pooling Visually Explained

Figure 5.5: Visualization of the different pooling strategies used in our experiments.

- (5) Concat-Mean-Max-Min Pooling: Concatenates the mean, max, and min pooling, generating a representation of size $3 \times d_{\text{embed}}$ that encapsulates multiple aspects of the token embeddings.
- (6) Attention Mean Pooling (Fig. 5.4): Given a token representation $\mathbf{x}_{\text{rep}}[i, j]$,

where i denotes the token and j represents a specific embedding dimension, this pooling method constructs a representation by combining token embeddings with a corresponding importance score. The importance score for each token i is computed by applying the softmax function to the mean of the token’s embedding values. This assigns higher weights to tokens with larger average values. These scores are multiplied element-wise to each dimension of the token embeddings, $x_{\text{rep}}[i, :]$, to adjust their contribution based on importance. The final pooled representation, x_{pooled} , is then computed as a weighted sum of these adjusted embeddings, emphasizing the most relevant tokens activations. Mathematically:

$$x_{\text{pooled}} = \sum_{i=1}^T \text{softmax} \left(\frac{1}{J} \sum_{j=1}^J x_{\text{rep}}[i, j] \right) \cdot x_{\text{rep}}[i, :].$$

These approaches offer diverse strategies for selecting representations, enabling a more nuanced understanding of how sentiment information is encoded. The choice of tokens should ideally reveal whether sentiment is distributed across the entire prompt or concentrated in specific tokens. For instance, the sentences “My name is XYZ and I’m happy” and “I’m happy because my name is XYZ” both convey positive sentiment. However, relying solely on the last token could lead to a suboptimal representation, as it may not adequately capture the sentiment expressed earlier in the sequence.

Choice of Classifier. With the hidden representations selected, we can train our probing classifier C_w on the sentiment labels. In our experimental setup we include twelve distinct classifiers, grouped into five categories based on their underlying modeling approach, as summarized in Table 5.1.

These classifiers were chosen to balance simplicity, interpretability, and the capacity to model complex patterns. Linear models, such as Logistic Regression and Linear SVM, serve as baselines for probing tasks due to their simplicity and their ability to identifying and leveraging linear relationships. Non-linear and neural network models, on the other hand, are included for their ability to capture intricate relationships within the hidden space. BiLSTM and CNN, in particular, were selected for their proven ability to generate effective hidden representations (Ghafoor et al., 2023).

All models, except for BiLSTM and CNN, are implemented using the scikit-

Table 5.1: Five different clusters of probing classifiers used in this study based on associated model type.

Type	Models
Linear	Logistic Regression, Linear SVM
Distance-based	K-Nearest Neighbors
Tree-based	Decision Tree, Random Forest, XGBoost, LightGBM
Neural Network	MLP, BiLSTM, CNN
Other	Non-linear SVM, Gaussian Naive Bayes

learn library (Pedregosa et al., 2011). While, BiLSTM and CNN were implemented using PyTorch. For each classifier, we employed the Optuna framework (Akiba et al., 2019) for hyperparameter optimization, focusing on key parameters such as regularization strengths, tree depths, and kernel types. Optimization was performed over five trials (Paparella et al., 2023), balancing computational efficiency with sufficient exploration of the hyperparameter space. To ensure reproducibility, we configured random seed initialization and enforced deterministic behavior for CUDA operations.

Datasets Details. Our experiments utilize three benchmark datasets for sentiment classification: IMDB (Maas et al., 2011), SST-2 (Socher et al., 2013), and Rotten Tomatoes (Pang and Lee, 2005), all of which contain movie reviews for binary polarity tasks (e.g., positive or negative sentiment). Additionally, we extended the setup to a more complex evaluation, conducting fine-grained sentiment classification using the Emotion dataset (Saravia et al., 2018), which categorizes sentiments into six nuanced classes: joy, sadness, anger, fear, love, and surprise.

We only preprocessed the IMDB and Emotion datasets because their original sizes, 50K for IMDB and 20K for Emotion, made them impractical for the large number of experiments. The IMDB and Emotion datasets were preprocessed due to their original sizes—50K for IMDB and 20K for Emotion—which made them impractical for the extensive number of experiments. To address this, reduced versions of these datasets were created while preserving their statistical properties. A structured preprocessing workflow was used to prepare the datasets for the experiments, ensuring statistical consistency and adherence to specific constraints, such as sentence length and label distribution. The key steps are outlined below:

1. **Initialize Random Seed:** To achieve reproducible datasets, we initialized

Table 5.2: Summary of the number of samples and train-test splits for the four sentiment and emotion datasets used in this study.

Dataset	Train	Test	Labels	Max Len
IMDB	7,000	7,000	pos/neg	132
SST-2	6,920	1,821	pos/neg	56
Rotten Tomatoes	8,530	1,066	pos/neg	59
Emotion	6,000	2,000	six labels	64

a random seed at the beginning of the workflow, ensuring that the sampling process could be replicated exactly in future runs.

2. Calculate Dataset Statistics:

- (a) Compute the label distribution to maintain the proportion of samples for each label in the original dataset.
- (b) Compute the average sentence length across all samples to preserve similar statistics.

3. Generate the Dataset:

For each label in the dataset, we followed a sampling process as follows:

- (a) Filter samples by sentence length for the given label.
- (b) Random sample to select the required number of samples for each label.
- (c) Combine the selected samples for all labels into a single dataset. To ensure randomness, the combined dataset was shuffled before saving.

4. Validate Reduced Dataset:

The label distribution and average sentence length were recalculated for the reduced dataset. If the average sentence length exceeded the specified constraint, an error was raised, indicating the need for further adjustments.

This comprehensive preprocessing workflow ensured that the reduced datasets retained the statistical properties of the original datasets while adhering to predefined constraints. Table 5.2 provides details on the number of samples and the train-test splits used in the experiments.

5.1.2 Experimental Results

We conduct our experiments on three Llama models, specifically testing the instruction-tuned variants of Llama 3.2-1B, Llama 3.2-3B, and Llama 3.1-8B, all

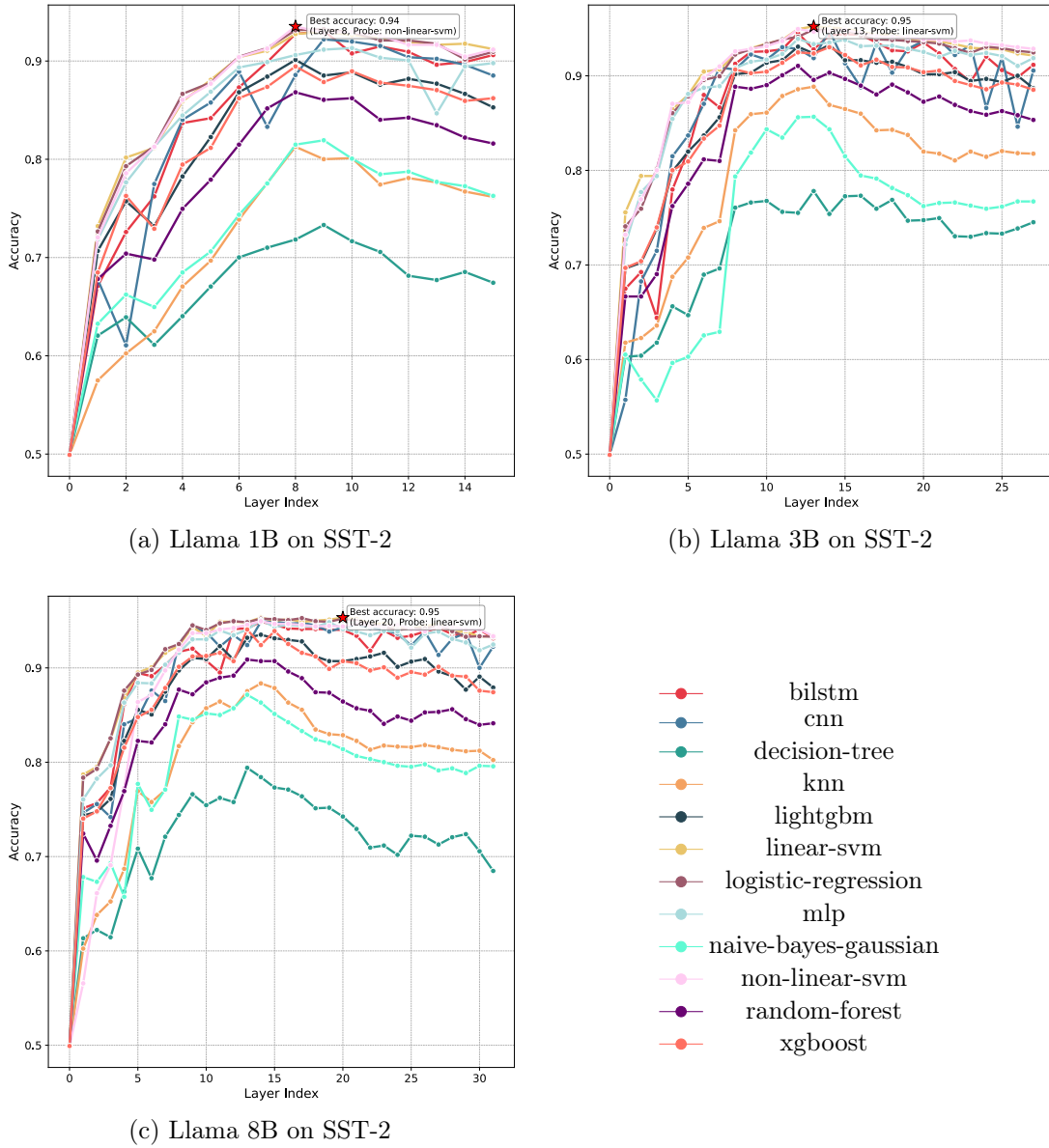


Figure 5.6: Layer-wise probing accuracy using the Last-Token approach on the SST-2 dataset, for each of the twelve probing classifiers reported in Table 5.1. Figures show the best performing combination of layer and probing classifier.

sourced from Hugging Face². These models were selected as they represent the most recent publicly available instruction-tuned releases in the Llama family at the time of our experiments. For every probing datasets we train the classifier C_w on the training set, and evaluate its detection performance by measuring the accuracy on the test set.

²<https://huggingface.co/meta-llama>

Table 5.3: Probing results across different pooling methods, reporting the top-3 best-performing combinations of layer/pooling method/prober for each of the datasets used in this study.

Model	Dataset	Layer	Prober	Pooling	Accuracy
Llama 3.2 1B (Instruct)	SST2	10	Non-linear SVM	Attn	0.9450
		10	Non-linear SVM	Mean	0.9450
		8	Non-linear SVM	Last-Token	0.9352
	IMDB	8	Logistic Reg.	Mean	0.9400
		8	Logistic Reg.	Attn	0.9396
		8	Linear SVM	Last-Token	0.9009
	Rotten	8	Linear SVM	Concat	0.8939
		8	Logistic Reg.	Concat	0.8939
		8	Non-linear SVM	Last-Token	0.8789
	Emotion	1	Linear SVM	Concat	0.7880
		10	LightGBM	Max	0.7880
		7	Linear SVM	Last-Token	0.6830
Llama 3.2 3B (Instruct)	SST2	5	Non-linear SVM	Concat	0.9594
		12	Non-linear SVM	Concat	0.9577
		13	Linear SVM	Last-Token	0.9522
	IMDB	4	Logistic Reg.	Attn	0.9523
		14	Logistic Reg.	Mean	0.9522
		13	Non-linear SVM	Last-Token	0.9177
	Rotten	3	Logistic Reg.	Concat	0.9090
		3	CNN	Mean	0.9071
		13	Linear SVM	Last-Token	0.8949
	Emotion	1	MLP	Concat	0.8220
		0	LightGBM	Concat	0.8195
		4	Linear SVM	Last-Token	0.6940
Llama 3.1 8B (Instruct)	SST2	3	Logistic Reg.	Concat	0.9605
		14	Logistic Reg.	Concat	0.9599
		14	Linear SVM	Last-Token	0.9533
	IMDB	4	Non-linear SVM	Concat	0.9579
		16	Non-linear SVM	Mean	0.9576
		11	Linear SVM	Last-Token	0.9273
	Rotten	9	MLP	Mean	0.9203
		14	CNN	Mean	0.9203
		14	Logistic Regression	Last-Token	0.9183
	Emotion	0	Linear SVM	Concat	0.8685
		0	LightGBM	Max	0.8655
		4	Linear SVM	Last-Token	0.6885

Sentiment detection results. In our first experiments, we evaluate the detection performance of the different classifier C_w . To extract the rep_θ , following prior

work, we focus on the residual stream using the Last-Token approach. We report detection accuracies for each layer in LLM_θ and visualize the results in Fig. 5.6.

For sentiment detection in binary polarity tasks (i.e., SST-2, IMDB, and Rotten Tomatoes), non-linear SVM, linear SVM, and logistic regression consistently outperform other probing techniques, achieving approximately 90% accuracy in the middle layers across all model sizes. For fine-grained emotion datasets, linear SVM demonstrates the best performance, reaching around 70% accuracy in the initial layers across all model sizes. This indicates that Llama models have linear representations for binary sentiment (positive/negative) and fine-grained emotions (joy, sadness, anger, fear, love, and surprise). It is interesting to discover that, although sentiment is not easily discernible during direct interaction with Llama, sentiment and emotion concepts can be linearly detected within certain layers.

Furthermore, since prior work often use the last-token representations, we also investigate the effect of different token representation methods. Specifically, we compare the last-token approach with five alternatives: mean, max, min, concatenation, and attention, as previously described.

The results, summarized in Table 5.3, which shows the top-3 performers by dataset, reveal that concatenating the mean, max, and min of rep_θ is often the most effective method for detecting sentiment concepts. Notably, mean and attention pooling also perform strongly, achieving results comparable to concatenation and consistently outperforming last-token pooling. This indicates, that the last-token representation is not always the optimal choice for capturing sentiment concepts.

Furthermore, we discover that combining the max pooling representation with LightGBM yields particularly strong performance. Further investigation reveals that tree-based models, including Random Forest, Decision Tree, and XGBoost, also perform better with max pooling. We hypothesize that the synergy between max pooling and tree-based models arises from max pooling’s ability to emphasize dominant features, which, when combined with tree-based models’ strength in exploiting high-contrast, threshold-based partitions, leads to an optimal combination. This alignment makes max pooling particularly effective for tree-based models, highlighting how the choice of classifier should also influence the choice of representation, and vice versa.

5.1.3 SENTRILLAMA for Efficient Downstream Tasks

Sentiment analysis has been a fundamental task since the inception of NLP. Traditionally, there have been two primary approaches to accomplish this task: (1) training a model from scratch or (2) fine-tuning an existing model to meet specific requirements. Recently, a third approach has gained prominence: leveraging well-designed prompts in conjunction with state-of-the-art large language models (Deng et al., 2023; Xing, 2025; Ahmed et al., 2024).

We introduce SENTRYLLAMA, a specialized Llama model that leverages layers up to $L_{\leq i}$ for sentiment tasks, where i denotes the most representative layer for the task. Based on our earlier analysis, we identified the optimal layer L_i , which retains the most salient features for downstream tasks. By attaching a lightweight, task-specific classification head—such as a LinearSVM—SENTRYLLAMA efficiently repurposes the expressive power of the Llama model while discarding unnecessary layers. This approach significantly reduces computational requirements for inference, making it efficient and task-specific without compromising performance.

Specifically, the total model parameters are calculated as the sum of the input embedding, N -layers, and LM head parameters. The proposed SENTRYLLAMA approach reduces complexity by retaining only the input embedding and the layers up to the most representative layer (i) identified in previous experiments, while replacing the LM head with a lightweight classification head (e.g., linear SVM), reducing parameter usage and adapting the architecture for downstream tasks.

To benchmark the effectiveness of SENTRYLLAMA, we compare its performance against fine-tuned DeBERTaV3-large³ and RoBERTa-large⁴ across all datasets. Additionally, we evaluate SENTRYLLAMA against Llama models under zero-shot, few-shot, and Chain-of-Thought prompting scenarios using carefully designed templates inspired by prior work (Deng et al., 2023), which are exemplified in figures 5.7, 5.8, 5.9, 5.10, 5.11, and 5.12.

Furthermore, to understand the effect of instruction fine-tuning on the sentiment task, we also include in the comparison the non-instructed version of Llama3.2 (1B). Table 5.4 presents a comparative analysis of the accuracy of SENTRYLLAMA, DeBERTa, RoBERTa, and a prompt-based method.

For the sentiment downstream task, SENTRYLLAMA 8B outperforms all other approaches, achieving approximately 96% accuracy on SST-2, 96% on IMDB,

³Microsoft/deberta-v3-large

⁴FacebookAI/roberta-large

Zero-Shot Prompting for Binary Sentiment	
System:	"You are an assistant trained to perform strict sentiment and emotion classification. You MUST only respond with the numeric label corresponding to the classification. Do not provide any explanations, reasoning, or any text other than the required numeric value."
User:	"Classify the sentiment of the following text: '{text}'"
Assistant:	"If the sentiment is positive, respond with '1'. If the sentiment is negative, respond with '0'. No other text, explanation, or formatting."

Figure 5.7: Example of a zero-shot prompting template used for binary sentiment classification. The system is strictly instructed to return only numeric sentiment labels.

92% on Rotten Tomatoes, and 87% on the Emotion dataset. Only DeBERTa surpasses this performance on the Emotion dataset, with an accuracy of 88%. The prompt-based approach reveals the effectiveness of few-shot and Chain-of-Thought (CoT), achieving comparable results but falling short on the Emotion dataset. These results align with expectations, revealing several insights: (1) increasing the base model size improves performance, (2) the non-instruction version of the 1B model exhibits minimal differences compared to the instructed ones, and (3) DeBERTa and RoBERTa continue to lead the leaderboard in sentiment analysis, outperforming prompt-based methods. However, it is worth noting that the actual size of SENTRILLAMA differs from the original Llama model. We represent the effective sizes in Table 5.5.

Using SENTRILLAMA reduces the size of the models. For the 1B model (both the standard and instruction-finetuned versions) on SST-2, IMDB, and Rotten Tomatoes, approximately 19% of the parameters are removed, while for the Emotion dataset, the reduction is up to 61.6%. For the 3B model on SST-2, IMDB, and Rotten Tomatoes, the reduction is approximately 36.6%, and for the Emotion dataset, it is 80%. For the 8B model on SST-2, IMDB, and Rotten Tomatoes, approximately 53.7% of the parameters are removed, while for the Emotion dataset, the reduction is 90.7%. This result indicates that the amount of parameters removed depends heavily on the specific task and that it is possible to reuse the highly relevant representations of Llama to improve downstream tasks, going beyond prompting and resulting in a competitive model.

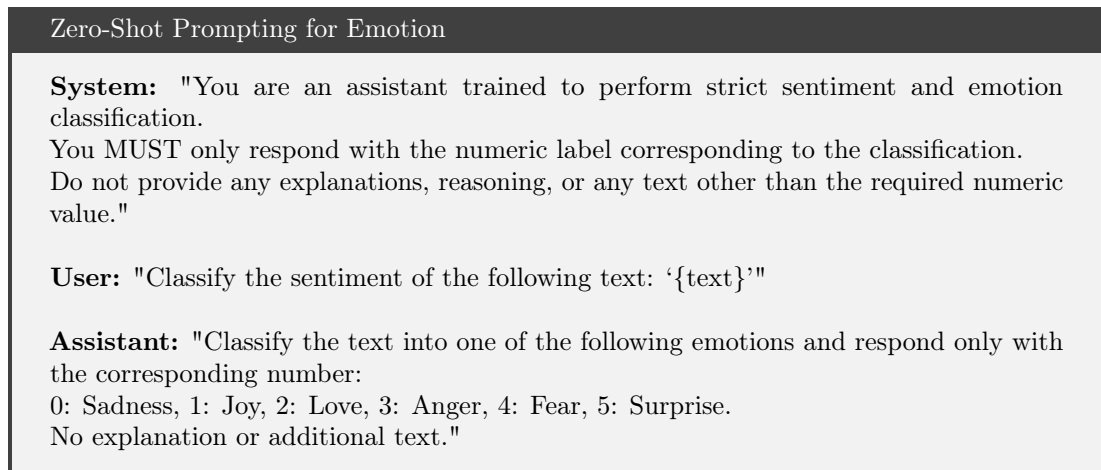


Figure 5.8: Example of a zero-shot prompting template used for emotion classification. The system is strictly instructed to return only numeric sentiment labels.

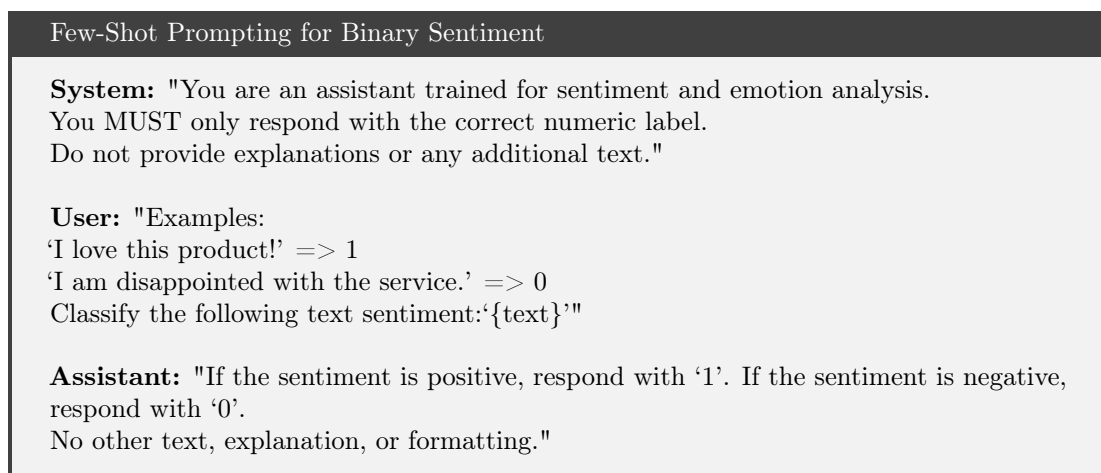


Figure 5.9: Example of a few-shot prompting template used for binary sentiment classification. The system is strictly instructed to return only numeric sentiment labels.

Furthermore, we compare the computational efficiency of SENTRILLAMA with DeBERTa, RoBERTa, and a prompt-based method. In Table 5.6, we present the results for SST-2.

Comparing GPU usage, SENTRILLAMA models demonstrate greater memory efficiency than their Instruct-Llama counterparts. SENTRILLAMA 3.2 (1B) uses 37.5% less GPU memory than Instruct-Llama 3.2 (1B), while SENTRILLAMA 3.1 (8B) reduces memory usage by 79.2% compared to Instruct-Llama 3.1 (8B). How-

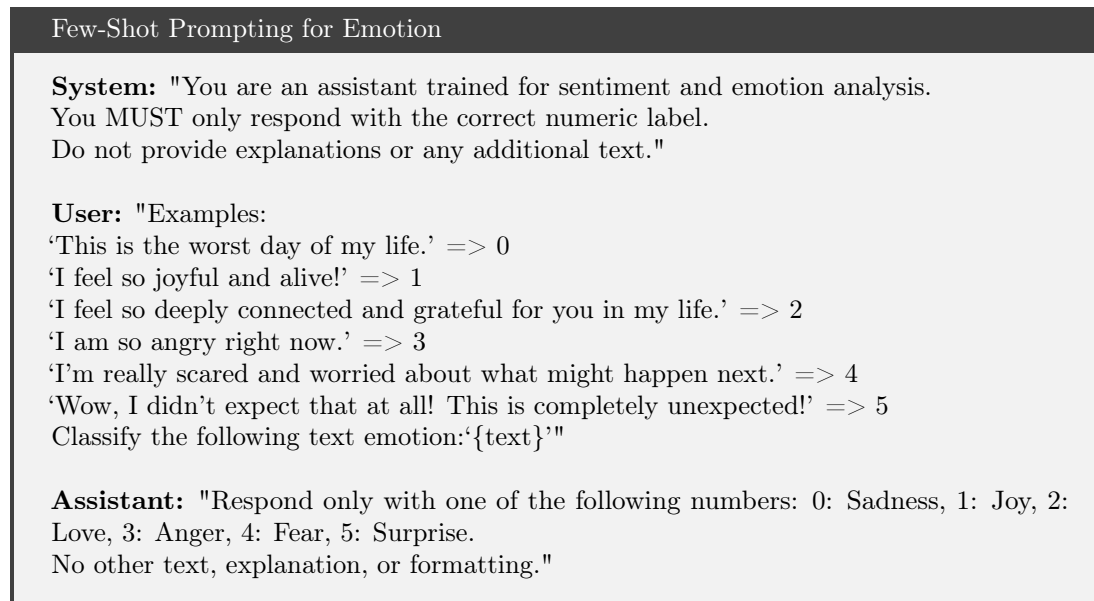


Figure 5.10: Example of a few-shot prompting template used for emotion classification. The system is strictly instructed to return only numeric sentiment labels.

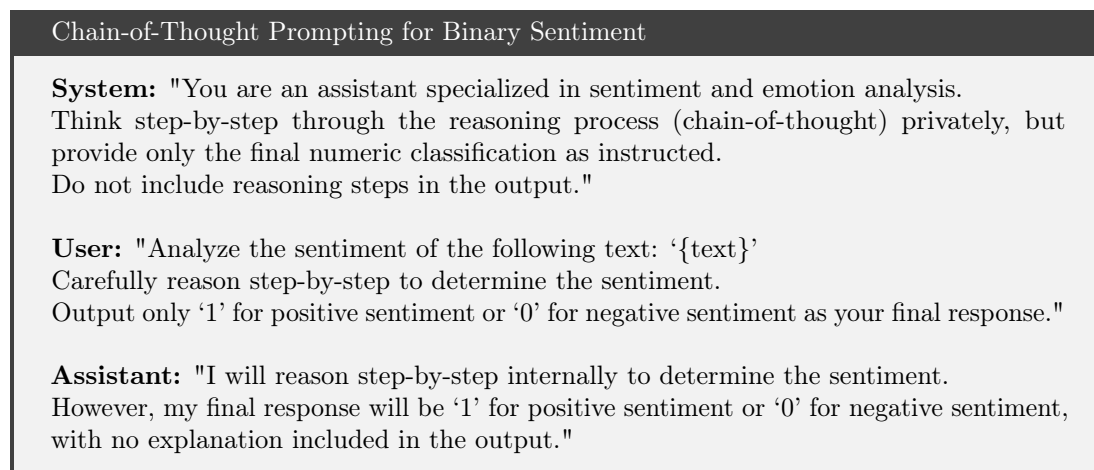


Figure 5.11: Example of a chain-of-thought prompting template used for binary sentiment classification. The system is strictly instructed to return only numeric sentiment labels.

ever, DeBERTa and RoBERTa remain the most memory-efficient, with RoBERTa requiring only 692 MB.

In terms of inference speed, SENTRILLAMA models exhibit a significantly lower average time per sample, indicating they are faster than their Instruct-Llama counterparts. Specifically, SENTRILLAMA 3.2 (1B) is 45.5% faster than Instruct-Llama 3.2 (1B), SENTRILLAMA 3.2 (3B) is 71.9% faster than Instruct-

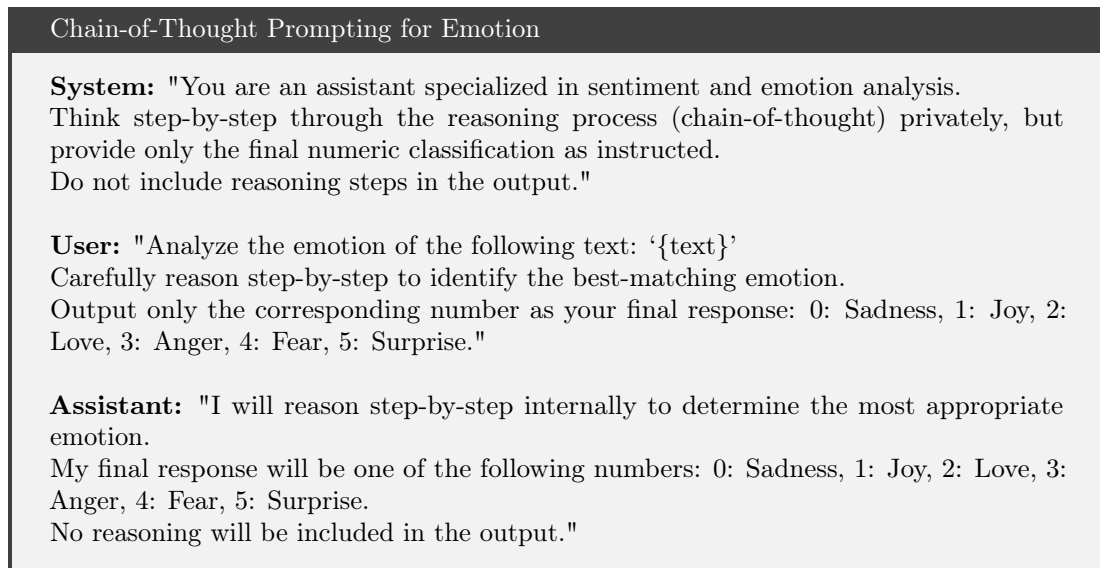


Figure 5.12: Example of a chain-of-thought prompting template used for emotion classification. The system is strictly instructed to return only numeric sentiment labels.

Llama 3.2 (3B), and SENTRILLAMA 3.1 (8B) is 85.9% faster than Instruct-Llama 3.1 (8B). Meanwhile, DeBERTa has a 75.9% longer inference time compared to SENTRILLAMA 3.2 (8B), while RoBERTa is 36.4% slower.

Finally, SENTRILLAMA models achieve significantly higher throughput (samples per second) than Instruct-Llama, DeBERTa, and RoBERTa.

Overall, this analysis reveals that SENTRILLAMA models are more efficient in terms of GPU memory usage and inference speed, consistently outperforming Instruct-Llama models across all aspects. Smaller models, such as SENTRILLAMA 3.2 (1B) and SENTRILLAMA 3.2 (3B), offer the best trade-off between memory consumption and speed. Meanwhile, RoBERTa Large provides competitive throughput and latency compared to larger Llama-based models, whereas DeBERTa struggles with slower inference.

5.1.4 Discussion

Our results show that our method is still far from providing a smaller Llama model that performs as well as similarly sized models like DeBERTa or RoBERTa. The primary limitation stems from the need to retain all preceding layers up to the most representative one. This limitation introduces an element of randomness: if circumstances align favorably and the most representative layer is among the early

Table 5.4: Comparison of SENTRILLAMA against DeBERTa, RoBERTa, and prompt-based method. The best-performing model is highlighted in **bold**, while the runner-up is underlined.

Model	SST2	IMDB	Rotten Tomatoes	Emotion
Instruct-Llama 3.2 (1B)				
Zero-shot	0.7210	0.6898	0.6923	0.2140
Few-shot	0.6485	0.5994	0.5994	0.2885
Chain-of-Thought	0.4992	0.5000	0.5000	0.3475
Instruct-Llama 3.2 (3B)				
Zero-shot	0.7759	0.8397	0.7279	0.3750
Few-shot	0.7606	0.8528	0.7176	0.3045
Chain-of-Thought	0.9154	0.9306	0.8743	0.4645
Instruct-Llama 3.1 (8B)				
Zero-shot	0.9341	0.9461	0.9024	0.4455
Few-shot	0.9330	0.9411	0.8968	0.3340
Chain-of-Thought	0.9165	0.9363	0.8771	0.5605
SENTRI Llama 3.2 (1B)	0.9308	0.9445	0.8912	0.8015
SENTRI Llama 3.2 (1B) Instruct	0.9450	0.9400	0.8940	0.7880
SENTRI Llama 3.2 (3B) Instruct	0.9594	0.9523	0.9090	0.8220
SENTRI Llama 3.1 (8B) Instruct	0.9605	0.9579	0.9203	<u>0.8685</u>
DeBERTa V3 Large (418M)	<u>0.9599</u>	<u>0.9534</u>	0.8671	0.8765
RoBERTa Large (355M)	0.9038	0.9430	0.8808	0.8416

layers (as observed with the Emotion dataset), the final model’s size becomes highly competitive with DeBERTa and RoBERTa. However, if the most representative layer is in the middle layers, the number of parameters—although still lower than the full model—can exceed those of the baseline models.

While we have automated the discovery of the most representative layer and pooling methods (see Table 5.3), the dependence on preceding layers results in a suboptimal outcome. A more robust solution could involve leveraging only the input embeddings and the most representative layer, potentially complemented by layer compression or pruning techniques to reduce computational overhead while preserving performance.

Is this method generalizable for different downstream tasks beyond sentiment analysis? While we cannot yet provide a definitive answer on the method’s generalizability, prior studies have shown that LLMs capture a broad range of semantic concepts, such as truthfulness, honesty, and factual knowledge (Burns et al., 2023; Azaria and Mitchell, 2023; Li et al., 2023b). Our findings suggest that sentiment and emotion are similarly well-represented in these models. We

Table 5.5: Dataset-wise comparison of the different SENTRILLAMA sizes resulting from model compression (retaining only the input embedding and the layers up to the most representative layer).

Model	SST2	IMDB	Rotten Tomatoes	Emotion
SENTRI LLAMA 3.2 (1B)	810M	811M	811M	384M
SENTRI LLAMA 3.2 (1B) Instruct	932M	810M	810M	384M
SENTRI LLAMA 3.2 (3B) Instruct	2B	1.9B	1.8B	595M
SENTRI LLAMA 3.1 (8B) Instruct	3.5B	3.8B	3.8B	743M

Table 5.6: Computational efficiency comparison of different SENTRILLAMA versions with DeBERTa, RoBERTa, and a prompt-based method. The reference dataset is SST-2.

Model	Peak GPU Usage	Avg. Time per Sample	Throughput (Samples/sec)
Instruct-Llama 3.2 (1B)	2.4 GB	11.17 ms	90
Instruct-Llama 3.2 (3B)	6.2 GB	18.19 ms	55
Instruct-Llama 3.1 (8B)	15.4 GB	37.73 ms	48
SENTRI LLAMA 3.2 (1B)	1.5 GB	6.08 ms	164
SENTRI LLAMA 3.2 (1B) Instruct	1.7 GB	7.98 ms	125
SENTRI LLAMA 3.2 (3B) Instruct	1.7 GB	5.09 ms	196
SENTRI LLAMA 3.1 (8B) Instruct	3.2GB	5.31 ms	182
DeBERTa V3 Large (418M)	845 MB	22.03 ms	45
RoBERTa Large (355M)	692 MB	8.35 ms	120

hypothesize that this approach could extend to tasks requiring nuanced linguistic analysis, such as sarcasm detection or intent recognition, assuming the availability of labeled datasets.

However, significant challenges remain. Extending this method to more complex tasks may require additional adaptations. Future research should systematically evaluate this approach across diverse tasks and datasets to uncover its broader potential and limitations.

5.1.5 Limitations

This work has two primary limitations. First, the study does not evaluate the proposed approach on domain-specific datasets, such as those from financial, healthcare, or social media domains, nor does it address multilingual sentiment analysis. This gap restricts the generalizability of the findings to a broader range of real-world applications. Second, while the study introduces a method for

identifying the most representative layer, there is scope for exploring alternative methodologies within this selection mechanism.

5.1.6 Summary

In this work, we present a systematic analysis of hidden representations in Llama for sentiment analysis. Our findings demonstrate that for positive/negative sentiment, the middle layers are the most representative across dimensions, whereas for emotion analysis, the most representative layers are found in the initial ones. Additionally, we show that while the current standard for representing a sentence involves using the last token, alternative methods, such as concatenating the max, min, and mean of the representations, yield better results.

Finally, we introduce a novel approach called SENTRILLAMA, which utilizes only a subset of the model’s parameters. This approach achieves results comparable to state-of-the-art models and outperforms prompting-based methods. We hope that the introduction of this layer selection approach for Llama will inspire the development of more robust techniques, facilitating the reuse of these large models for downstream tasks and extending their usability beyond text generation. Furthermore, we have made all the code publicly available on GitHub ⁵ to enable systematic and efficient probing of LLMs.

⁵<https://github.com/sisinflab/LLM-SentimentProber.git>

5.2 *Do We Really Need Specialization?* Evaluating Generalist Text Embeddings for Zero-Shot Recommendation and Search ⁶

Recommendation and Search systems are essential for helping users navigate large information spaces, from retrieving relevant documents to providing personalized product suggestions. Traditional methods often rely on ID-based embeddings or keyword-based matching, which may struggle to capture the nuanced semantics of user intent and item content, especially in cold-start scenarios. Recent advances in Natural Language Processing, especially the rise of pretrained language models (PLMs), have created new opportunities for tackling these challenges. Dense retrieval models (Khattab and Zaharia, 2020; Hou et al., 2024) based on PLMs, like BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020), move beyond shallow matching by capturing rich representations of queries, items, and users. In recommendation, particularly in sequential settings, they enhance item embeddings by modeling content and user behavior dynamics (Hou et al., 2022; Yuan et al., 2023) while incorporating their internal domain knowledge (Di Palma et al., 2025a); in search, they improve query-item alignment. Despite their benefits, PLMs are not directly optimized for recommendation and search tasks, often limiting generalization (Hou et al., 2024). Recent work has addressed this through task-specific fine-tuning. One notable example is BLAIR (Hou et al., 2024), a RoBERTa-based (Liu et al., 2019) embedding model, fine-tuned on large-scale user reviews and item metadata to better align query and item embeddings, improving efficacy in sequential recommendation and product search. While domain and task adaptation methods such as BLAIR highlight their benefits of specialized models, recent advances in text embedding techniques have led to the emergence of *Generalist Text Embedding Models (GTEs)* (Lee et al., 2025; Li et al., 2023c). Trained on diverse corpora and tasks, GTEs learn broad, transferable semantics, offering a training-free alternative to fine-tuning at scale. Thus, they may offer a strong alternative as embedding models for recommendation and search, achieving competitive performance without specific adaptation.

This motivates our research question: **How well do GTEs perform in**

⁶Based on our paper, “Do We Really Need Specialization? Evaluating Generalist Text Embeddings for Zero-Shot Recommendation and Search”, in *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 575–580 (2025).

recommendation and search tasks in a zero-shot setting, compared to traditional and fine-tuned models? To address this question, we evaluate state-of-the-art GTEs on two key tasks: *sequential recommendation* and *product search*, focusing on models included in the *Massive Text Embedding Benchmark (MTEB)* (Muennighoff et al., 2023). Our results show that GTEs outperform both traditional and fine-tuned models, with recent open-source models (e.g., NVEEmbed-v2 (Lee et al., 2025) and GTE-Qwen2 (Li et al., 2023c)) even surpassing popular closed-source alternatives such as OpenAI’s.

Moreover, our study takes a first step towards addressing a largely open question: *What underlying characteristics of textual embeddings influence their effectiveness for recommendation and search?* We focus on factors related to embedding geometry and dimensionality, with the aim to uncover the key characteristics that govern embedding effectiveness. In particular, we analyze *space utilization*: we examine how effectively models distribute variance across embedding dimensions, addressing concerns about *dimensional collapse* (Ethayarajh, 2019). We use principal component analysis (PCA) to estimate the *effective dimensionality*, which provides a quantitative measure of the phenomenon. This analysis reveals that GTEs tend to achieve more uniform space utilization. Building on this, we demonstrate that PCA provides a pathway for effective compression: retaining only the most informative components substantially reduces GTE dimensionality without sacrificing accuracy, improving scalability. We further find that this compression also benefits fine-tuned models as it removes noisy components, reducing the performance gap compared to GTEs.

Thesis Note. It is important to clarify the nature of GTEs. Unlike the "raw" PLMs analyzed in previous chapters, GTEs are themselves derived from PLMs but have undergone an additional, large-scale fine-tuning stage (e.g., contrastive learning on diverse text pairs) by their original creators to specialize them for the general modality of embedding generation. This process, however, is neither domain- nor task-specific. We treat these GTEs as off-the-shelf, generalist components. Our investigation aligns with the core frozen paradigm by using these models as-is, without any parameter updates or domain adaptation, to evaluate their zero-shot effectiveness. This upholds our commitment to sustainability by leveraging a single, pre-trained (in the broad sense) model for multiple applications.

5.2.1 Background and Methodology

This section outlines our methodology and formalizes two core tasks: sequential recommendation and product search.

Task Overview

Sequential Recommendation (SR). Sequential recommendation focuses on modeling a user’s evolving interests based on their interaction history, with the goal of predicting the next item of interest. Let a user session be represented as a sequence $\mathcal{S}_t = [i_1, i_2, \dots, i_t]$, $i_j \in \mathcal{I}$, where $\mathcal{I}$ is the set of all items, and i_j denotes the item interacted with at instant j . The recommendation task can be framed as learning a function $f_{\text{seq}}(\mathcal{S}_t) \rightarrow i_{t+1}$.

Recent approaches employ neural sequence models to encode the interaction sequence. Each item i_j is mapped to an embedding $\mathbf{e}_{i_j} \in \mathbb{R}^d$ using an item encoder, and a sequence encoder ϕ_{seq} transforms the embedded sequence into a contextual representation $\mathbf{h}_t = \phi_{\text{seq}}([\mathbf{e}_{i_1}, \mathbf{e}_{i_2}, \dots, \mathbf{e}_{i_t}])$, where $\mathbf{h}_t \in \mathbb{R}^d$ captures the user’s dynamic preference at time t . This representation $\mathbf{h}_t$ is then used to score or rank the candidate items via a similarity measure, $\text{Sim}(\cdot, \cdot)$, as $f_{\text{seq}}(\mathcal{S}_t) = \arg \max_{i \in \mathcal{I}} \text{Sim}(\mathbf{h}_t, \mathbf{e}_i)$.

Product Search (PS). The goal of product search techniques is to retrieve the most relevant items from a large catalog $\mathcal{I} = \{i_1, i_2, \dots, i_N\}$ in response to a user query $q \in \mathcal{Q}$ (the space of possible queries), often expressed in natural language. Recent approaches significantly improve upon traditional lexical matching (e.g., BM25) by leveraging PLMs to map queries and items into a shared high-dimensional embedding space. A pre-trained text embedding model generates dense vector representations $\mathbf{e}_q$ for queries and $\mathbf{e}_i$ for items, typically using their textual attributes (like titles, descriptions, or metadata). The core task is to learn a scoring function $f(q, i) \rightarrow \mathbb{R}$ that reflects the relevance of item i to query q . This score is commonly computed as the similarity between their respective embeddings: $f(q, i) = \text{Sim}(\mathbf{e}_q, \mathbf{e}_i)$ where $\text{Sim}(\cdot, \cdot)$ often represents cosine similarity or dot product. Items are then ranked according to this relevance score.

Embedding Space Utilization

Text embedding models, which map text to dense representations, often suffer from *dimensional collapse* (Ethayarajh, 2019), where variance concentrates in a

small subset of dimensions. We employ *effective dimensionality* (Cai et al., 2021) to quantitatively assess this effect, given that PCA is a conventional (Rudman et al., 2022), interpretable technique which facilitates both isotropy analysis and the practical removal of components for compressibility. Let $\mathbf{E}_{\mathcal{I}}^c \in \mathbb{R}^{N \times d}$ denote the set of mean-centered item embeddings, extracted from a pre-trained encoder using item metadata. Applying PCA yields singular values $\{\sigma_i\}_{i=0}^{d-1}$ for the covariance matrix of $\mathbf{E}_{\mathcal{I}}^c$. The **explained variance ratio** for the top k components is defined as $r_k = \sum_{i=0}^{k-1} \sigma_i / \sum_{i=0}^{d-1} \sigma_i$. The ε -**effective dimension** is then $d(\varepsilon) := \arg \min_k r_k \geq \varepsilon$, representing the minimum number of components needed to retain at least an ε fraction of the total variance. Following prior work (Cai et al., 2021), we robustly estimate principal components by applying PCA to a random subset $\hat{\mathbf{E}}_{\mathcal{I}}^c \in \mathbb{R}^{n \times d}$, where $n \gg d$ (Rudman et al., 2022). Projecting query embeddings onto the resulting subspace ensures representational consistency, reduces dimensionality, and preserves the most discriminative features. This analysis (i) quantifies **space utilization** (Rudman et al., 2022), which impacts downstream performance (Machina and Mercer, 2024), and (ii) establishes a **lower bound for compression** critical for scaling large GTEs while maintaining informativeness.

5.2.2 Experiments and Discussion

We evaluate different GTEs on sequential recommendation (SR) and product search (PS), as defined in Section 5.2.1, and analyze the properties of their embeddings in relation to performance. Our experiments address two core research questions:

RQ1: *How well do GTEs perform in recommendation and search tasks in a zero-shot setting, compared to traditional and fine-tuned models?*

RQ2: *What underlying characteristics of textual embeddings influence their effectiveness for recommendation and search?*

Experimental Setup

Datasets. To ensure fair comparison with prior work, we adopt the experimental configuration of Hou et al. (2024) for both SR and PS. We use the Amazon Reviews 2023 dataset (Hou et al., 2024), accessed via the HuggingFace⁷ `datasets` API,

⁷<https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023>

following the original data splits and preprocessing. Item metadata is constructed by concatenating the *title*, *features*, *categories*, and *description* fields.

For SR, we focus on the *All Beauty* (Beauty), *Video Games* (Games), and *Baby Products* (Baby) categories, with user-item interactions ranging from 104,766 to 3,583,323, reflecting varying data sparsity. To ensure temporal consistency, the interactions were split chronologically based on an absolute timestamp strategy, with 80%, 10%, and 10% designated for training, validation, and test, respectively. For PS, we use the ESCI (Reddy et al., 2022) and Amazon-C4 (Hou et al., 2024) datasets. ESCI provides real-world `<query,item>` pairs with graded relevance labels. Following Hou et al. (2024), we select only *exact* matches from the English *ESCI-small* split for higher-quality supervision, containing 27,643 queries and 1,367,729 items. Amazon-C4 comprises 21,223 queries and 1,058,417 items, with synthetic queries generated from 5-star reviews using ChatGPT (Blog, 2022). Queries in both datasets are linked to Amazon Reviews 2023 via product IDs. Although both datasets are sourced from the same e-commerce catalog, they exhibit notable differences: ESCI has keyword queries, Amazon-C4 long-form ones, allowing us to test GTE generalization across formats. We report results on two representative categories: *Office Products* (Office) and *Sports and Outdoors* (Sports). According to Hou et al. (2024), each `<query,item>` pair is ranked against 50 randomly sampled items per domain.

Models

For SR, we consider the baseline models GRU4Rec (Hidasi et al., 2016), SASRec (Kang and McAuley, 2018), and UniSRec (Hou et al., 2022). Both GRU4Rec and SASRec are designed to operate with item ID embeddings. To incorporate textual information, we follow the strategy proposed by Hou et al. (2024), using item metadata as textual inputs. This results in two variants for each model: an **ID**-based version using standard embeddings and a **Text**-based version with text-derived representations. For the text-based variants, we test several pre-trained text embedding models: BLAIR (Hou et al., 2024) (base and large), OpenAI’s `text-embedding-3-large`⁸ (`t-emb-3`) and two open-source GTEs: KALM (Ni et al., 2022a) and NVEmbed-v2 (Lee et al., 2025). KALM matches BLAIR in parameter count and embedding size for fair comparison, while NVEmbed-v2 offers

⁸API version: 2024-02-01

a more powerful GTE with higher dimensionality and an increased parameter size.

For PS, we expand our evaluation to include a range of text encoders: RoBERTa_{BASE} (Liu et al., 2019), ColBERTv2 (Santhanam et al., 2022), mGTE (Zhang et al., 2024d), INSTRUCTOR_{XL} (Su et al., 2023), Sentence-T5_{XXL} (Ni et al., 2022a), Jasper (Zhang and FulongWang, 2024), and GTE-Qwen2 (Li et al., 2023c). These models represent a mix of encoder- and decoder-based transformers, covering various parameter sizes and embedding dimensions for a comprehensive comparison.

Evaluation Protocol

For SR, we employ the RecBole (Zhao et al., 2021; Zhao et al., 2022) framework, adopting the optimized hyperparameter settings from (Hou et al., 2024), as they are already well-tuned for the Amazon Reviews 2023 dataset and ensure fair comparison. Early stopping is applied based on nDCG@10 with a patience of 10. For PS, models are implemented using Hugging Face Transformers, with a maximum input sequence length of 512. Retrieval is performed using cosine similarity, which empirically outperforms Euclidean and dot product measures. The metrics employed are Recall and nDCG. We analyze embedding space utilization by computing the ε -effective dimension using PCA on a random sample of 100,000 item embeddings (Rudman et al., 2022).

Model Comparison (RQ1)

Table 5.7 presents the results for the SR task. Textual embeddings consistently outperform ID-based models, highlighting the importance of semantic item information. Among the embedding models, NVEmbed-v2 and KALM, both of the GTE family, deliver the most substantial gains across all architectures. NVEmbed-v2 achieves the best results on the Games and Baby categories, while KALM slightly outperforms NVEmbed-v2 on Beauty. The improvements offered by leading GTE models are statistically significant compared to the best BLAIR baseline for most text-based architectures and datasets. The significance was evaluated by comparing the best GTE with the leading BLAIR variant for each model-dataset pair, based on nDCG@10. Furthermore, GTEs surpass the closed-source model **t-emb-3** on two datasets. These results showcase the benefits of recent, high-capacity embedding methods that do not depend on domain-specific data.

Table 5.7: Sequential recommendation performance: Recall@k (R_k) and nDCG@k (N_k). Results are reported as percentages. Best scores are bolded, second-best underlined. ID stands for ID-based, T stands for Text-based.

Model	Beauty				Games				Baby			
	R_{10}	N_{10}	R_{50}	N_{50}	R_{10}	N_{10}	R_{50}	N_{50}	R_{10}	N_{10}	R_{50}	N_{50}
GRU4Rec^{ID}	0.31	0.15	1.11	0.32	2.11	1.14	5.28	1.82	1.10	0.56	3.55	1.09
SASRec^{ID}	0.31	0.18	0.84	0.30	2.32	1.15	5.55	1.85	1.23	0.58	3.69	1.11
GRU4Rec^T												
BLAIR _B	0.63	0.30	1.67	0.53	2.16	1.14	5.43	1.84	1.29	0.65	4.05	1.24
BLAIR _L	0.84	0.42	2.09	0.69	2.35	1.29	5.90	2.06	1.26	0.64	3.96	1.22
t-emb-3	0.57	0.27	1.78	0.55	3.01	1.64	7.39	<u>2.59</u>	0.44	0.21	1.67	0.47
NVEmb [†]	0.75	0.36	1.74	0.57	2.98*	<u>1.65*</u>	7.27*	2.57*	<u>1.63*</u>	<u>0.84*</u>	4.92*	<u>1.54*</u>
KALM	0.69	0.36	1.76	0.59	2.67	1.46	6.52	2.30	1.45	0.73	4.51	1.38
SASRec^T												
BLAIR _B	0.69	0.35	2.16	0.66	1.90	1.01	5.01	1.68	1.24	0.63	3.84	1.19
BLAIR _L	1.21	0.50	2.55*	0.77	2.18	1.17	5.36	1.86	1.33	0.68	4.04	1.26
t-emb-3	0.84	0.42	2.20	0.71	2.77	1.51	6.92	2.41	0.40	0.20	1.43	0.42
NVEmb [†]	0.77	0.43	2.05	0.70	2.78*	1.54*	6.87*	2.42*	1.50*	0.75*	4.51*	1.40*
KALM	0.88	0.45	1.80	0.65	2.40	1.30	5.83	2.04	1.44	0.73	4.38	1.35
UniSRec^T												
BLAIR _B	2.60	1.39	4.69	1.85	2.47	1.30	6.08	2.07	1.52	0.77	4.34	1.37
BLAIR _L	2.51	1.38	4.54	1.83	2.53	1.37	6.24	2.17	1.53	0.78	4.36	1.39
t-emb-3	3.60*	1.86*	5.73*	2.31*	<u>3.05</u>	<u>1.64</u>	<u>7.46</u>	<u>2.59</u>	0.44	0.24	1.57	0.48
NVEmb [†]	2.93	1.58	4.77	2.00	3.11*	1.65*	7.50*	2.59*	1.72*	0.89*	4.87*	1.57*
KALM	<u>3.45</u>	<u>1.84</u>	<u>5.55</u>	<u>2.29</u>	2.75	1.47	6.68	2.32	1.55	0.78	4.39	1.39

([†]) NVEmb is adopted as abbreviation of NVEmbed-v2.

(*) Statistically significant improvement ($p < 0.001$) between the best GTE model and the best BLAIR-based model.

Table 5.8 shows the PS results on the ESCI and Amazon-C4 datasets. GTE-based models consistently surpass traditional models (RoBERTa and ColBERTv2) and BLAIR. Notably, top-performing GTEs achieve statistically significant gains over the best-performing BLAIR baseline, validating the superiority of the GTE approach. Scores are generally higher on ESCI due to its simpler queries: GTE-Qwen2 improves from 0.1792 on Amazon-C4 to 0.2806 on ESCI, and Jasper from 0.1710 to 0.2662. An exception is BLAIR, which performs well on Amazon-C4 but drops sharply on ESCI, suggesting that fine-tuning hinders generalization. RoBERTa performs worse across the board, reflecting its limitations in retrieval tasks. Among the top performers are high-capacity GTEs, with NVEmbed-v2 leading on Amazon-C4 and GTE-Qwen2 on ESCI.

Overall, our findings indicate that **GTEs achieve strong performance in both tasks without requiring task-specific fine-tuning, outperforming traditional and fine-tuned solutions, as well as closed-source models.**

Table 5.8: Comparison of text encoders for product search (nDCG@100) on the two datasets (results are percentages). “All” averages scores for all domains. We also report two domains, “Office” and “Sports”.

	Model	#Par	Dim	ESCI			Amazon-C4		
				All	Office	Sports	All	Office	Sports
	BM25	—	—	1.10	1.35	1.57	0.00	0.00	0.00
<i>Encoders</i>	RoBERTa_B	123M	768	0.08	0.03	0.25	0.25	0.33	0.37
	ColBERTv2	110M	768	1.71	1.80	1.44	0.88	1.60	0.51
	BLAIR_B	123M	768	11.76	12.56	14.26	14.90	18.02	22.24
	BLAIR_L	354M	1024	12.12	12.81	13.89	17.18	20.07	<u>25.68</u>
	mGTE	305M	768	22.17	22.32	24.12	13.11	15.33	17.38
	INSTRUCTOR_{XL}	1.5B	768	20.53	20.13	21.32	12.22	15.79	16.38
	Sentence-T5_{XXL}	11B	768	21.75	22.13	22.98	13.88	17.77	19.22
<i>Decoders</i>	KALM	0.5B	896	23.12	23.76	24.39	15.61	18.55	22.01
	Jasper	2B	1024	26.62	26.07	27.11	17.10	20.23	23.16
	GTE-Qwen2	7B	3584	28.06*	28.09*	29.09*	<u>17.92</u>	21.41*	24.03
	NVEmbed-v2	11B	4096	<u>27.59</u>	<u>27.24</u>	<u>28.45</u>	19.02*	<u>21.32</u>	25.91*
	t-emb-3	—	3072	25.76	24.67	26.96	17.53	20.86	25.23

(*) Statistically significant improvement ($p < 0.001$) between the best GTE model and the best BLAIR-based model.

Model Analysis (RQ2)

To understand the superior performance of GTEs, we analyze model capacity, embedding structure, and architectural design. Table 5.8 shows that **performance does not scale linearly with model capacity** (no. of parameters) **and dimensionality**: KALM (0.5B) outperforms larger models such as INSTRUCTOR_{XL} (1.5B) and Sentence-T5_{XXL} (11B), while Jasper (1024 dimensions) outperforms t-emb-3 (3072 dimensions) on ESCI. In the SR task, KALM is found to outperform high-dimensional GTEs under specific model-dataset combinations (e.g., GRU4Rec-Beauty).

To assess space utilization, we compute the ε -effective-dimensionality (Section 5.2.1). Figure 5.13 shows how the nDCG varies with the number of retained components in the PS task (points correspond to ε values of 0.80, 0.95, and 1.00, left to right). An nDCG increase with fewer components suggests noisy directions in the feature space (indicating that the removed, low-variance dimensions contained mostly noise), while a drop indicates that information is uniformly distributed. We observe that for large GTEs (e.g., NVEmbed-v2 and

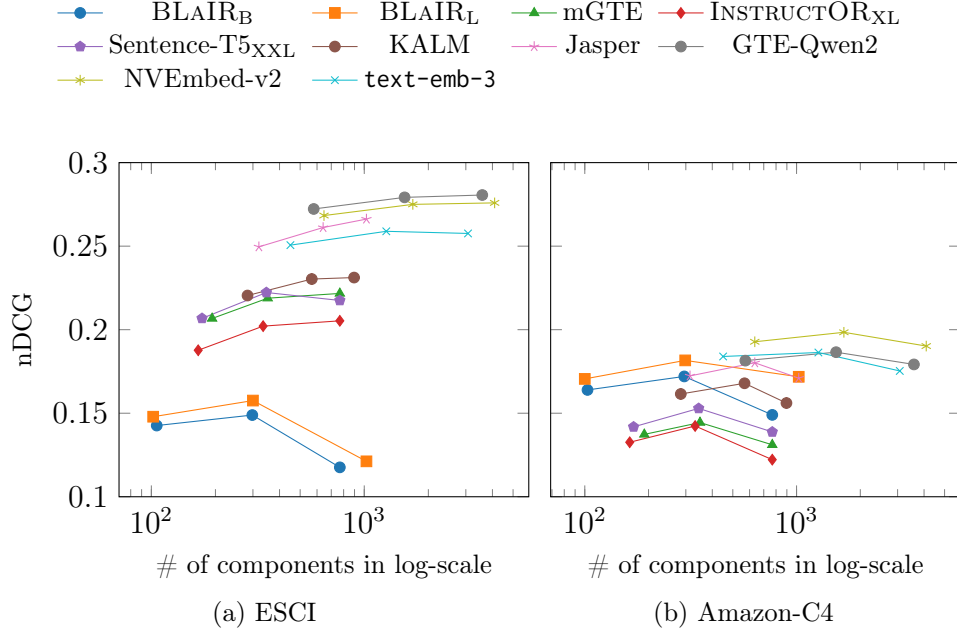


Figure 5.13: nDCG scores for various models on the two PS datasets (a) ESCI and (b) Amazon-C4 versus the number of components (log-scale). Points shown correspond to component counts yielding 80%, 95%, and 100% explained variance.

GTE-Qwen2), the number of dimensions needed to preserve 80% of the total variance is close to the full dimensionality of BLAIR_B, suggesting more efficient use of representational space. BLAIR exhibits a significant drop in components with $\varepsilon = 0.80$, hinting at high anisotropy (Cai et al., 2021). For GTEs, this phenomenon is less pronounced. While BLAIR benefits from PCA showing a marked inverted-U trend, many GTEs show a plateau or slight improvement at 95% variance: for large GTEs, moderate compression substantially reduces dimensionality without compromising performance. Conversely, on Amazon-C4, all models benefit from compression, resulting in a performance gain. This trend is particularly pronounced for BLAIR, indicating presence of noise in low-variance components.

Architecturally, we note that decoder-style models (Jasper, GTE-Qwen2, NVEmbed-v2) outperform encoder-based ones, with generative, autoregressive, or retrieval-oriented training yielding semantically rich, task-aligned representations (BehnamGhader et al., 2024; Lee et al., 2025).

Overall, we show that **GTEs, particularly decoder-based ones, tend to spread useful information more evenly across embedding dimensions**, possibly explaining their strong performance. In contrast, we find that **model ca-**

capacity and dimensionality often show little correlation with downstream performance.

5.2.3 Summary

This work investigates the zero-shot capabilities of GTEs in sequential recommendation and product search. Despite the potential advantages of domain-specific fine-tuning, we find that GTEs can outperform fine-tuned models (e.g., BLAIR), classical architectures (e.g., ColBERT), and closed-source solutions (e.g., OpenAI) without task-specific adaptation. We identify more uniform space utilization as a potential key factor for their success and show that emphasizing the most informative dimensions via PCA can boost the performance of fine-tuned models and effectively compress dimensionality for large GTEs. These results position GTEs as strong alternatives when fine-tuning is impractical. Future work could further improve embedding properties, especially isotropy and disentanglement, as explored in state-of-the-art models like UniSRec, and devise methods to adapt GTEs to specific domains while preserving their generalization capability. Lastly, recent findings suggest that LLMs can memorize recommendation datasets (Di Palma et al., 2025c), raising the possibility that GTEs, trained on large-scale data, may also exhibit training leakage; future work could explore how such memorization affects embedding distributions. To ensure reproducibility, we publicly release the code on GitHub ⁹.

⁹<https://github.com/sisinflab/GTE-Zero-Shot-Recsys>

5.3 Related Work

5.3.1 Sentiment Information in PLMs

Sentiment Analysis Meets LLMs

Sentiment analysis, the task of classifying and extracting subjective information from text, has evolved from lexicon-based approaches using dictionaries of positive and negative words (Hatzivassiloglou and McKeown, 1997; Wiebe, 2000) to supervised learning methods employing n-gram models and machine learning (Pang et al., 2002; Chaovalit and Zhou, 2005). These early methods, limited by their reliance on handcrafted features, paved the way for feature engineering techniques and eventually the deep learning revolution that reshaped sentiment classification.

To improve feature identification, researchers incorporated advanced techniques such as syntactic dependencies (Yao et al., 2010), part-of-speech tagging (Søgaard, 2011), and negation handling (Morante and Blanco, 2012). The advent of deep learning further transformed sentiment analysis, with models such as Recurrent Neural Networks (RNNs) (Socher et al., 2013) and Convolutional Neural Networks (CNNs) (Kim, 2014), achieving significant advancements by learning representations directly from raw text.

However, the introduction of Transformer architectures (Vaswani et al., 2017) marked a paradigm shift. Models like BERT (Devlin et al., 2019) leverage pretraining on vast datasets to extract contextualized representations, and they have been widely used as encoding backbones for downstream tasks. Conversely, GPT (Radford et al., 2019) employs the Transformer-decoder block and frames language modeling as an autoregressive task. Finally, inspired by the transfer learning paradigm, Raffel et al. (2020) introduced T5, an encoder-decoder model for text-to-text tasks, trained on vast amounts of data to capture general language patterns and fine-tuned for a wide range of specific applications.

The introduction of GPT-3 (Brown et al., 2020) marked a turning point by introducing few-shot learning and demonstrating the effectiveness of prompt-based techniques, significantly reducing the reliance on large labeled datasets for sentiment analysis and other downstream tasks.

Modern models like GPT-4 (OpenAI, 2023) and Llama (Touvron et al., 2023a) achieve even greater flexibility through instruction-following capabilities, excelling in zero-shot settings (Qin et al., 2023) and outperforming fine-tuned models in sentiment-related tasks (Hasan et al., 2024).

Nowadays, the latest models used for sentiment analysis include DeBERTaV3 (He et al., 2023) and RoBERTa-large (Liu et al., 2019), which have demonstrated excellent performance when fine-tuned. Additionally, models like GPT and Llama have shown effectiveness in this domain, leveraging prompting techniques or instruction fine-tuning (Stigall et al., 2024; Krugmann and Hartmann, 2024; Li et al., 2023d; Wei et al., 2023a), even in complex aspect-based scenarios (Bai et al., 2024; Zheng et al., 2025).

Probing LLMs for Sentiment Analysis

Various studies have assessed the capabilities of LLMs for sentiment analysis tasks. For instance, Fatouros et al. (2023) analyzed the performance of ChatGPT-3.5 in financial sentiment analysis, demonstrating performance that exceeds FinBERT (Araci, 2019). Similarly, Araujo et al. (2024) investigated ChatGPT’s effectiveness in Portuguese sentiment analysis, highlighting its potential value in dataset annotation. Lyu et al. (2024) investigates the application of causal inference to sentiment analysis and introduces causal prompts to enhance LLM performance in sentiment prediction tasks. Furthermore, Zhang et al. (2024b) conducted a systematic evaluation across various sentiment tasks using ChatGPT and different T5 model sizes. Their findings reveal that while LLMs excel in simple (e.g. binary or trinary) zero-shot sentiment classification tasks, they struggle with complex ones (e.g. aspect-based).

While previous studies have primarily evaluated LLMs as *text-to-text models*, focusing on zero- and few-shot learning capabilities, our work takes a different approach. We investigate the hidden representations within the intermediate layers of transformer architectures to identify where sentiment concepts are encoded and how these insights can inform the development of more efficient and accurate models.

Similar studies have investigated the representation of semantic concepts within LLMs (Anelli et al., 2022), such as Burns et al. (2023) on truthfulness, Azaria and Mitchell (2023) on honesty, Roy et al. (2023) on hate speech, and Li et al. (2023b) on factual knowledge. However, the investigation of sentiment within the hidden representations of these models remains comparatively underexplored.

5.3.2 Language Models as Text Embedders for Recommendation and Search

Generalist Text Embedding Models

GTEs (Li et al., 2023c) are text embedding models trained on large-scale corpora and diverse tasks to produce rich, transferable representations for a variety of downstream applications. Several encoder-based GTEs build on the transformer encoder (Devlin et al., 2019) architecture. For instance, mGTE (Zhang et al., 2024d) extends BERT (Devlin et al., 2019) with RoPE (Su et al., 2024) and unpadding (Portes et al., 2023) for long-context support. INSTRUCTOR (Su et al., 2023) enhances GTR (Ni et al., 2022b) via instruction tuning for task-aware embeddings, while Sentence-T5 (Ni et al., 2022a) derives embeddings from the T5 (Raffel et al., 2020) encoder block. Decoder-only LLMs have also been adapted for embedding tasks. GTE-Qwen2 (Li et al., 2023c) repurposes Qwen2 (Yang et al., 2024) via multi-stage training to generate universal embeddings. Jasper (Zhang and FulongWang, 2024) applies Matrioshka Representation Learning (Kusupati et al., 2022) to distill embeddings from a larger teacher model. NVEmbed-v2 (Lee et al., 2025), based on Mistral-7B (Jiang et al., 2023), removes causal masking and introduces latent attention for improved token-aware pooling. KALM (Hu et al., 2025), being fine-tuned on a small model (i.e., Qwen2-0.5B), shows that even compact models can yield high-quality embeddings when trained on large-scale, multi-task datasets.

Text Encoders for Sequential Recommendation and Product Search

Sequential recommendation models predict a user’s next interaction based on past behavior. Transformer-based models (e.g., SASRec (Kang and McAuley, 2018) and BERT4Rec (Sun et al., 2019)) address the limitations of recurrent models such as GRU4Rec (Hidasi et al., 2016), employing self-attention for long-range dependencies and parallelization. Yet, many approaches rely solely on item IDs, ignoring side information like metadata and images. UniSRec (Hou et al., 2022) demonstrates the effectiveness of incorporating multimodal signals, combining ID and semantic features (Yuan et al., 2023).

Unlike recommendation, product search requires explicit query-item matching. Traditional methods (e.g., BM25) have been supplanted by deep models that learn dense query and product embeddings. Transformer-based approaches

like BERT (Devlin et al., 2019) and ColBERT (Khattab and Zaharia, 2020) enhance semantic matching but face limitations due to a certain scarcity of public datasets (Merra et al., 2023; Merlo et al., 2025). To address these limitations, BLAIR (Hou et al., 2024) introduces a contrastive learning-based text encoder using reviews as pseudo-queries. Still, most text embedding solutions in recommendation and search rely on closed-source models (Harte et al., 2023; Boz et al., 2024) or domain-specific fine-tuning (e.g., BLAIR) leaving GTEs largely unexplored. Despite the growing popularity of GTEs, there is still no common consensus on which model to use for search and recommendation. Our work fills this gap through a systematic evaluation.

Embedding Quality Evaluation

Embedding models are commonly evaluated implicitly through downstream task performance. MTEB (Muennighoff et al., 2023) provides a comprehensive suite for assessing the performance of GTEs across such tasks. Beyond downstream performance, it is also crucial to investigate the intrinsic properties of embedding models, as these can significantly influence their effectiveness and suitability in practical applications. Factors such as embedding dimension and model capacity affect not only accuracy but also the scalability and sustainability of retrieval systems (Kusupati et al., 2022). Another key aspect is space utilization: a frequent issue in contextual pre-trained language models is *dimensional collapse* (Ethayarajh, 2019; Gao et al., 2021), where embeddings lie in low-rank subspaces, limiting expressiveness and degrading performance (Machina and Mercer, 2024) in tasks reliant on distance metrics (Ni et al., 2022a). Consequently, post-processing methods (Li et al., 2020; Zhang et al., 2024a) have been proposed to improve embedding utility. In this work, we investigate whether GTEs inherently possess properties that enable efficient space utilization, potentially eliminating the need for ad-hoc adaptation.

5.4 Overview: Choice of Representation

This section provides a consolidated overview of the representation extraction strategies adopted in the works presented in this chapter, with a particular focus on the analyses described in Section 5.1 and Section 5.2. Its role is primarily expository: rather than introducing new methodological components, it summarizes where and how internal signals are accessed from frozen PLMs. This overview is intended to support readers who may be less familiar with probing-based or representation-centric analyses of language models.

We begin by describing the representation extraction process used for sentiment and emotion analysis in Section 5.1, illustrated in Figure 5.14. Unlike approaches that rely on a single representation (most commonly the final-layer hidden state of the last token) we perform a comprehensive analysis that is both layer-wise and token-wise. This design choice reflects the observation that sentiment is often expressed locally within a sentence and may not be adequately captured by a single token representation, especially when sentiment-bearing words occur early or mid-sequence.

Specifically, we analyze residual stream representations from Llama models, treating each token-level activation as a potential carrier of sentiment information. To obtain fixed-size vectors suitable for classification, we compare multiple pooling strategies over token representations, including last-token, mean, max, min, concatenation-based, and attention-based pooling. The resulting representations are then evaluated using a diverse set of probing classifiers. This setup allows us to isolate how sentiment information is distributed across layers, tokens, and representation choices, while ensuring that any predictive signal originates from the model’s internal representations rather than from task-specific parameter updates.

In contrast to the probing-based analyses of sentiment and emotion described above, the work on Generalist Text Embedding Models (GTEs) presented later in this chapter adopts a fundamentally different perspective on representation extraction. Rather than inspecting intermediate activations or token-level representations within a language model, GTEs are explicitly designed to produce a single, fixed-size embedding for each input text. These embeddings are intended to serve as general-purpose semantic representations that can be reused across tasks without task-specific adaptation.

As depicted in Figure 5.15, representation extraction consists of encoding

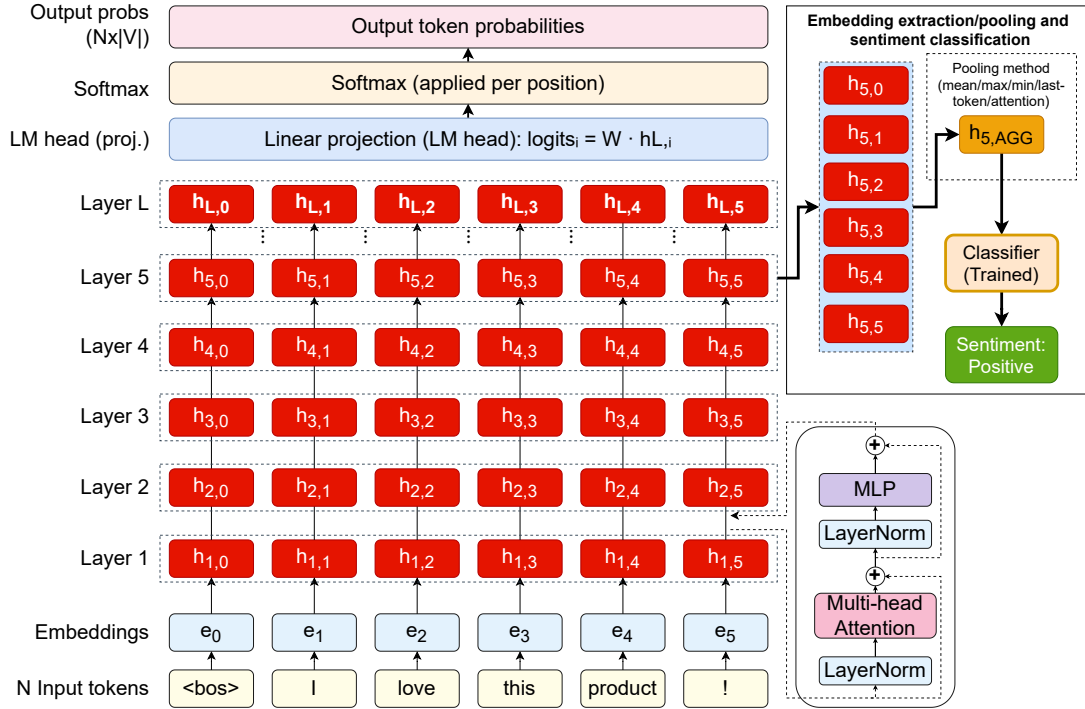


Figure 5.14: Overview of the representations extraction process adopted in Section 5.1. In this work, we perform a comprehensive, layer- and token-wise analysis of how sentiment information is encoded at the intermediate processing layers of a Llama model. Therefore, all intermediate, token-wise hidden states are taken into consideration. Aside from typical last-token pooling, we experiment with various pooling strategies, detailed in Section 5.1.1).

queries and items independently using a frozen GTE model and obtaining a sequence-level embedding through pooling over last-layer token representations. The resulting vectors are used directly for downstream tasks such as zero-shot recommendation and product search, without probing internal layers or analyzing token-wise contributions. As a result, the representational focus shifts from understanding how specific linguistic properties emerge within the transformer stack to evaluating the quality, geometry, and transferability of the embedding space itself.

Despite these methodological differences, the GTE analysis remains aligned with the overarching frozen paradigm adopted throughout this thesis. GTEs are treated as off-the-shelf components, used as-is without parameter updates, enabling a fair assessment of how broadly trained embedding models compare to specialized or fine-tuned alternatives. By examining both probing-based representations and

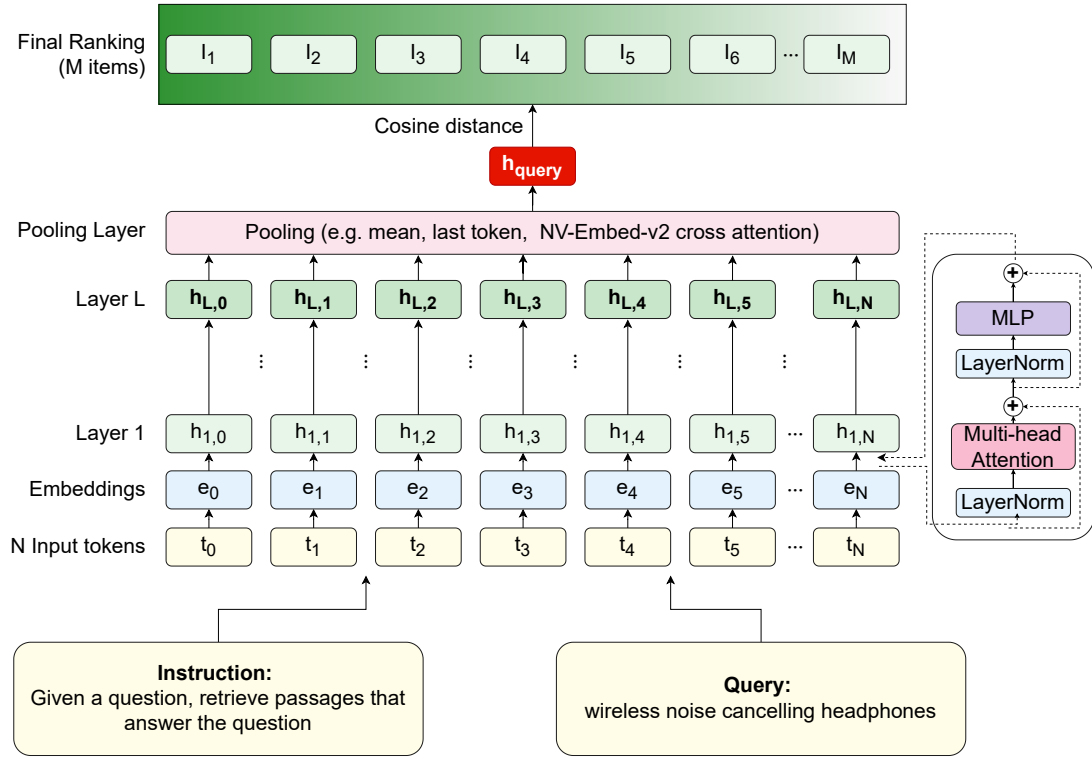


Figure 5.15: Overview of the representation extraction process adopted in Section 5.2. Although the figure only depicts the process for queries, queries and items are independently encoded using the same frozen Generalist Text Embedding (GTE) model, producing fixed-size dense vectors. Representations are obtained by aggregating final-layer token hidden states via a pooling mechanism, whose specific form depends on the underlying GTE architecture (e.g., NV-Embed-v2 employs a trainable latent attention pooling layer).

task-agnostic embeddings within the same chapter, this section provides a unified view of how different representational choices—ranging from internal activations to pooled text embeddings—support diverse downstream applications.

This page intentionally left blank

Chapter 6

Conclusion

“Even if it’s over now, I had a good time learning.
But I think the rules are about to change.”

— Chert, *Outer Wilds*

This thesis has examined the hidden representations of PLMs, guided by the hypothesis that substantial latent knowledge is encoded within their frozen parameters. This knowledge can be accessed and leveraged without resorting to the costly and unsustainable paradigm of fine-tuning. Through studies of semantics, factuality, and latent task capabilities, this work demonstrates that a model’s internal signals can be decoded to support complex tasks, pointing toward a more interpretable and sustainable future for AI. At the same time, our analyses highlight key challenges that persist in this line of research. This final chapter summarizes the main contributions of this work, outlines directions for future investigation, and reflects on its broader implications.

6.1 Summary of Contributions

On Semantic Structure and Type Knowledge. Frozen PLMs encode structured type knowledge and semantic containment relationships (Section 3.1). This implicit type understanding can be decoded into ontology-aligned structures and effectively applied to tasks such as entity typing and ontology completion, bridging symbolic and neural representations of meaning. By leveraging latent type information, PLM-based predictors can improve relational reasoning in incomplete or sparse graphs (Section 3.2). This provides evidence that pretrained models

encode generalizable inductive biases that can support knowledge completion and link prediction without additional fine-tuning.

On Factuality and Internal Self-Evaluation. Probing frozen models reveals that their hidden representations encode, to some degree, measurable signals related to the factual accuracy of generated text (Section 4.2). This property enables *factuality self-evaluation* (i.e., the ability of a model to assess the truthfulness of its own outputs) offering a promising direction toward more trustworthy and self-monitoring systems. However, our analysis also indicates that current factuality probing methods face significant challenges when applied to realistic, naturally occurring data, as opposed to controlled synthetic settings.

On Latent Capabilities for Downstream Tasks. Even without adaptation, frozen PLMs preserve competitive performance on sentiment and emotion classification tasks (Section 5.1). Furthermore, this work showed that models can be compressed efficiently while maintaining accuracy, highlighting opportunities for lightweight and sustainable deployment. The thesis further demonstrated that general-purpose text embedding models can perform competitively on recommendation and search tasks without task- or domain-specific fine-tuning (Section 5.2). This calls into question the assumption that domain specialization is always necessary, at least for these application contexts.

Together, these findings suggest that pretrained models are more than surface-level pattern matchers, acting in part as repositories of structured, interpretable, and actionable knowledge. They encode both conceptual and relational information in ways that, if properly elicited, can replace or complement parameter adaptation.

6.2 *The Journey Ahead: Current Limitations and Future Challenges*

This dissertation has charted a course through the hidden representations of PLMs, yet this exploration has primarily served to reveal the extent of the territory that remains uncharted. The insights gained are not an endpoint, but a compass, pointing toward promising directions the journey could take next. Building on these insights, this section revisits the limitations of the present work and outlines

a set of potential future research directions, organized along the three core pillars of the thesis.

6.2.1 On Explicit Semantics and Structured Knowledge

The works presented in Section 3.1 and Section 3.2 are limited by their reliance on frozen PLMs and on the knowledge they acquired during pretraining. While these models encode substantial world and ontological knowledge through large-scale self-supervised learning, this knowledge is bounded by the scope, quality, and biases of the pretraining data, as well as by the models’ representational and reasoning capacities. As a result, our analyses can only probe knowledge already embedded in the models and cannot evaluate inner reasoning mechanisms beyond their pretraining exposure. Moreover, PLMs trained on internet-scale corpora may encode societal biases and toxic patterns. Although our probing experiments do not explicitly address these issues, they may influence model behavior, making bias and toxicity mitigation an important direction for future work.

The limitations of the work in Section 3.2 mainly stem from methodological choices. The use of a fixed set of manually designed prompts, while effective, may not be optimal, and techniques such as adaptive prompt selection or prompt tuning could yield further improvements. The work in Section 3.1 also relies on fixed, templated prompts, but explores multiple template variants, resulting in a more robust analysis. In addition, several hyperparameters were selected empirically without a full optimization procedure. While this approach produced good results, it does not guarantee optimality. More systematic hyperparameter tuning and prompt optimization could further improve performance, but were computationally prohibitive at the scale of our experiments.

The following paragraphs outline future research directions motivated by each of the two studies.

Probing PLMs for Generalizable Relational Patterns. Future research could further integrate KGs and PLMs, extending the approach discussed in Section 3.1 to a broader range of semantic relations. A particularly interesting direction would be to investigate whether models can learn to infer a compact, generalizable relational vocabulary from latent representations and use it for direct verbalization of different relationship types. Additionally, strategies must be developed to address the long-tail problem, especially for sparse relationships

whose limited occurrences inhibit the training of effective verbalizers.

Generalizability of Learned Containment Patterns to Long-Tail Types.

A significant challenge for the verbalizer-based approach detailed in Section 3.1 is its performance on long-tail types. The effectiveness of the learned verbalizers, whether based on direct mapping or non-linear transformation, is contingent upon the availability of sufficient training examples. For types that are sparsely represented in the training corpus, the model may fail to learn a robust and generalizable mapping, leading to poor performance and a bias toward more frequent, data-rich types. Future work must therefore explore strategies to mitigate this data sparsity.

6.2.2 On Factuality and Internal Self-Evaluation

The work presented in Section 4.2 presents some limitations. While efforts were made to ensure the consistency of our constructed datasets, more extensive human annotation would improve robustness and mitigate potential biases. Moreover, the analysis is limited to a small set of models; extending it to larger and more diverse architectures could reveal different behaviors, particularly in self-evaluation settings. In addition, all models considered in this study are pretrained language models rather than instruction-tuned variants. Given the growing prevalence of instruction tuning and its known impact on model behavior, a natural extension of this work is to investigate how our findings translate to instruction-tuned models. Finally, our reliance on perplexity as a proxy for plausibility has inherent limitations. A more rigorous assessment of perplexity-based approaches, as well as the exploration of alternative, theoretically grounded plausibility measures, remains an important direction for future work.

The following paragraphs discuss aspects that merit consideration in future work.

Domain-Specific Factuality Self-Assessment. Building upon the findings discussed in Chapter 4, future work could explore the application of factuality self-evaluation techniques in specialized domains (e.g., medicine), where mitigating hallucinations and ensuring factual reliability are of critical importance. Such studies would also provide insights into how domain-specific knowledge interacts with general pretrained representations. Regarding the work discussed in Section 4.2,

we also acknowledge that our method for constructing true-false probing datasets, which relies on an LLM-as-judge, is a key limitation. More robust methods, such as those involving human annotation, are necessary to ensure reliability.

Are We Looking in The Right Place? Our procedure builds upon the SAPLMA (Azaria and Mitchell, 2023) probe, which probes the last token hidden state at various hidden layers for factuality signals. However, this approach of probing a single, static state may be too simplistic to fully capture the dynamics of sequential text generation. A key difficulty is that an early hallucination, even if uncertain, can propagate, causing subsequent claims to also be incorrect. This suggests that assessing only the final state is insufficient. This challenge has spurred research into more recent probing strategies (Vazhentsev et al., 2024; Sriramanan et al., 2024; Chuang et al., 2024) that can track and propagate uncertainty throughout the entire generation process, rather than assessing a single state. To broaden the scope of our findings, a natural development is to extend our evaluation to more recent probing methods.

6.2.3 On Latent Capabilities and Sustainable Applications

The work presented in Section 5.1 is subject to two main limitations. First, the proposed approach is evaluated only on general-domain, monolingual datasets, limiting its applicability to domain-specific and multilingual sentiment analysis settings. Extending the evaluation to specialized domains as well as to multilingual scenarios, would improve the generalizability of the findings. Second, although the study proposes a method for identifying the most representative layer, alternative selection strategies could be explored to further strengthen this component.

As shown in Section 5.2, generalist text embedding (GTE) models demonstrate strong zero-shot performance in search and recommendation tasks. However, the analysis does not fully explain why certain GTEs outperform others. While we provide empirical evidence suggesting that specific representational properties (e.g. isotropy) are associated with better retrieval performance, these observations remain largely descriptive. The underlying mechanisms that connect these properties to downstream effectiveness are not explicitly characterized, and causal relationships are not established. Consequently, identifying and rigorously validating the representational factors that drive the superior performance of some GTEs remains an open challenge and an important direction for future research.

Motivated by these limitations, the following paragraphs outline several directions for future research.

Generalizing Sentiment Encoding Mechanisms in LLMs. A key avenue for future research is to investigate whether the findings from Section 5.1 generalize across diverse languages and domains. This line of inquiry could employ causal probing to uncover the precise, domain-independent mechanisms that PLMs utilize for encoding sentiment and emotion, thereby validating the fundamental nature of our observations.

Probing-Guided Model Compression. In Section 5.1, we demonstrated that a specialized, weight-efficient model can be obtained by pruning layers subsequent to the most informative one, as identified by our probing analysis. A natural extension is to apply this framework to other human-interpretable concepts beyond sentiment and emotion. This direction is particularly promising for supporting *agentic LLM* (Li et al., 2023a) developments. Such agents must often decompose complex tasks into simpler sub-tasks (e.g., checking sentiment, classifying intent). Our method provides a pathway to create the lightweight, efficient "specialist" models needed for this delegation, enabling agents to operate far more efficiently than systems relying on a single, monolithic model or multiple large and inefficient delegates.

Exploring Post-Hoc Techniques to Enhance Representational Power. As established in Section 5.2, generalist text embedding (GTE) models exhibit significant potential for zero-shot search and recommendation tasks. However, their representational quality can be further enhanced. A documented limitation of LLM representations is their high anisotropy (Ethayarajh, 2019), a phenomenon where embeddings occupy a narrow cone within the vector space. Our own results corroborate that isotropy is a beneficial property for retrieval efficacy. Consequently, a promising research avenue involves applying post-hoc techniques, such as whitening, to improve embedding isotropy. This approach is particularly advantageous as it offers an additional method to enhance performance without necessitating costly model retraining. The primary challenge, however, lies in calibrating the optimal trade-off between improved isotropy and downstream task performance. As prior work (Rajaei and Pilehvar, 2021) has demonstrated, gains in isotropy do not invariably translate to improved representational utility.

6.3 Final Perspective

The thesis has contributed methodologically, empirically, and conceptually to the study of pretrained models:

- *Methodologically*, it introduced and applied probing procedures that interpret model internals through forward passes, without weight modification.
- *Empirically*, it provided evidence that frozen models possess useful latent competencies across semantics, factuality, and user-ended tasks.
- *Conceptually*, it contributed to the discussion of model usage around interpretability and sustainability, suggesting that many capabilities attributed to fine-tuned models may already be embedded in pretrained ones.

These contributions support a vision of interpretable and sustainable AI: interpretable, by mapping hidden signals to explicit, verifiable semantics; and sustainable, by minimizing the need for costly retraining while maintaining strong task performance.

In closing, this thesis underscores that progress in large language modeling need not rely solely on ever-larger models or repeated fine-tuning. By revealing the depth of knowledge already encoded within pretrained networks, we can build systems that are more interpretable, efficient, and sustainable. Understanding *what is already there*, before changing parameters, offers both scientific insight and practical impact, marking a helpful step toward transparent and responsible AI development.

This page intentionally left blank

Bibliography

- Rabbia Ahmed, Sadaf Abdul-Rauf, and Seemab Latif (2024). “Leveraging Large Language Models and Prompt Settings for Context-Aware Financial Sentiment Analysis”. In: *5th International Conference on Advancements in Computational Sciences, ICACS 2024, Lahore, Pakistan, February 19-20, 2024*. IEEE, pp. 1–9. DOI: 10.1109/ICACS60934.2024.10473283. URL: <https://doi.org/10.1109/ICACS60934.2024.10473283>.
- Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019). “Optuna: A Next-generation Hyperparameter Optimization Framework”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. Ed. by Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis. ACM, pp. 2623–2631. DOI: 10.1145/3292500.3330701. URL: <https://doi.org/10.1145/3292500.3330701>.
- Kamla Al-Mannai, Hanan Alshikhabobakr, Sabih Bin Wasi, Rukhsar Neyaz, Houda Bouamor, and Behrang Mohit (2014). “CMUQ-Hybrid: Sentiment Classification By Feature Engineering and Parameter Tuning”. In: *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*. Ed. by Preslav Nakov, and Torsten Zesch. The Association for Computer Linguistics, pp. 181–185. DOI: 10.3115/V1/S14-2028. URL: <https://doi.org/10.3115/v1/s14-2028>.
- Guillaume Alain, and Yoshua Bengio (2017). “Understanding intermediate layers using linear classifier probes”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net. URL: <https://openreview.net/forum?id=HJ4-rAVt1>.
- Vito Walter Anelli, Giovanni Maria Biancofiore, Alessandro De Bellis, Tommaso Di Noia, and Eugenio Di Sciascio (2022). “Interpretability of BERT Latent Space

- through Knowledge Graphs”. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*. Ed. by Mohammad Al Hasan, and Li Xiong. ACM, pp. 3806–3810. DOI: 10.1145/3511808.3557617. URL: <https://doi.org/10.1145/3511808.3557617>.
- Dogu Araci (2019). “FinBERT: Financial Sentiment Analysis with Pre-trained Language Models”. In: *CoRR* abs/1908.10063. arXiv: 1908.10063. URL: <http://arxiv.org/abs/1908.10063>.
- Gladson de Araujo, Tiago de Melo, and Carlos Maurício S. Figueiredo (2024). “Is ChatGPT an effective solver of sentiment analysis tasks in Portuguese? A Preliminary Study”. In: *Proceedings of the 16th International Conference on Computational Processing of Portuguese, PROPOR 2024, Santiago de Compostela, Galicia/Spain, March 12-15, 2024, Volume 1*. Ed. by Pablo Gamallo, Daniela Barreiro Claro, António J. S. Teixeira, Livy Real, Marcos García, Hugo Gonçalo Oliveira, and Raquel Amaro. Association for Computational Linguistics, pp. 13–21. URL: <https://aclanthology.org/2024.propor-1.2>.
- A. Arditì, and O. B. Obeso (2023). *Refusal Mechanisms: Initial Experiments with Llama-2-7b-chat*. <https://www.lesswrong.com/posts/pYcEhoAoPfhHgJ8YC>. Accessed: 2024-11-18.
- Matteo Attimonelli, Alessandro De Bellis, Claudio Pomo, Dietmar Jannach, Eugenio Di Sciascio, and Tommaso Di Noia (2025). “Do We Really Need Specialization? Evaluating Generalist Text Embeddings for Zero-Shot Recommendation and Search”. In: *Proceedings of the Nineteenth ACM Conference on Recommender Systems. RecSys ’25*. Association for Computing Machinery, pp. 575–580. ISBN: 9798400713644. DOI: 10.1145/3705328.3748040. URL: <https://doi.org/10.1145/3705328.3748040>.
- Amos Azaria, and Tom M. Mitchell (2023). “The Internal State of an LLM Knows When It’s Lying”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Association for Computational Linguistics, pp. 967–976. DOI: 10.18653/v1/2023.FINDINGS-EMNLP.68. URL: <https://doi.org/10.18653/v1/2023.findings-emnlp.68>.
- Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). “Layer Normalization”. In: *CoRR* abs/1607.06450.

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio (2015). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio, and Yann LeCun. URL: <http://arxiv.org/abs/1409.0473>.
- Yinhao Bai, Zhixin Han, Yuhua Zhao, Hang Gao, Zhuowei Zhang, Xunzhi Wang, and Mengting Hu (Jan. 2024). “Is Compound Aspect-Based Sentiment Analysis Addressed by LLMs?” In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 7836–7861. DOI: 10.18653/v1/2024.findings-emnlp.460. URL: <https://aclanthology.org/2024.findings-emnlp.460>.
- Klaudia Balazy, Lukasz Struski, Marek Smieja, and Jacek Tabor (2023). “r-softmax: Generalized Softmax with Controllable Sparsity Rate”. In: *Computational Science - ICCS 2023 - 23rd International Conference, Prague, Czech Republic, July 3-5, 2023, Proceedings, Part II*. Ed. by Jiri Mikyska, Clélia de Mulatier, Maciej Paszynski, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M. A. Sloot. Vol. 14074. Lecture Notes in Computer Science. Springer, pp. 137–145. DOI: 10.1007/978-3-031-36021-3_11. URL: https://doi.org/10.1007/978-3-031-36021-3_11.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy (2024). “LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders”. In: *CoRR* abs/2404.05961. DOI: 10.48550/ARXIV.2404.05961. arXiv: 2404.05961. URL: <https://doi.org/10.48550/arXiv.2404.05961>.
- Yonatan Belinkov (2022a). “Probing Classifiers: Promises, Shortcomings, and Advances”. In: *Comput. Linguistics* 48.1, pp. 207–219. DOI: 10.1162/COLI_A_00422. URL: https://doi.org/10.1162/coli_a_00422.
- Yonatan Belinkov (2022b). “Probing Classifiers: Promises, Shortcomings, and Advances”. In: *Comput. Linguistics* 48.1, pp. 207–219. DOI: 10.1162/COLI_A_00422. URL: https://doi.org/10.1162/coli_a_00422.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin (2003). “A Neural Probabilistic Language Model”. In: *J. Mach. Learn. Res.* 3, pp. 1137–1155. URL: <https://jmlr.org/papers/v3/bengio03a.html>.

- Russa Biswas, Radina Sofronova, Mehwish Alam, and Harald Sack (2021). “Contextual Language Models for Knowledge Graph Completion”. In: *Machine Learning with Symbolic Methods and Knowledge Graphs co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2021), Virtual, September 17, 2021*. Ed. by Mehwish Alam, Mehdi Ali, Paul Groth, Pascal Hitzler, Jens Lehmann, Heiko Paulheim, Achim Rettinger, Harald Sack, Afshin Sadeghi, and Volker Tresp. Vol. 2997. CEUR Workshop Proceedings. CEUR-WS.org. URL: <https://ceur-ws.org/Vol-2997/paper3.pdf>.
- OpenAI Blog (2022). *Introducing ChatGPT*. <https://openai.com/index/chatgpt/>. [Accessed 18-04-2025].
- Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda B. Viégas, and Martin Wattenberg (2021). “An Interpretability Illusion for BERT”. In: *CoRR* abs/2104.07143. arXiv: 2104.07143. URL: <https://arxiv.org/abs/2104.07143>.
- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. (2021). “On the Opportunities and Risks of Foundation Models”. In: *CoRR* abs/2108.07258. arXiv: 2108.07258. URL: <https://arxiv.org/abs/2108.07258>.
- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko (2013). “Translating Embeddings for Modeling Multi-relational Data”. In: *NIPS*, pp. 2787–2795.
- Monali Bordoloi, and Saroj K. Biswas (2023). “Sentiment analysis: A survey on design framework, applications and future scopes”. In: *Artif. Intell. Rev.*

- 56.11, pp. 12505–12560. DOI: 10.1007/S10462-023-10442-2. URL: <https://doi.org/10.1007/s10462-023-10442-2>.
- Artun Boz, Wouter Zorgdrager, Zoe Kotti, Jesse Harte, Panos Louridas, Dietmar Jannach, and Marios Fragkoulis (2024). “Improving Sequential Recommendations with LLMs”. In: *CoRR* abs/2402.01339. DOI: 10.48550/ARXIV.2402.01339. arXiv: 2402.01339. URL: <https://doi.org/10.48550/arXiv.2402.01339>.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. URL: <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html>.
- Lennart Bürger, Fred A. Hamprecht, and Boaz Nadler (2024). “Truth is Universal: Robust Detection of Lies in LLMs”. In: *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*. Ed. by Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang. URL: http://papers.nips.cc/paper%5C_files/paper/2024/hash/f9f54762cbb4fe4dbffdd4f792c31221-Abstract-Conference.html.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt (2023). “Discovering Latent Knowledge in Language Models Without Supervision”. In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. URL: <https://openreview.net/forum?id=ETKGuby0hcs>.
- Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church (2021). “Isotropy in the Contextual Embedding Space: Clusters and Manifolds”. In: *9th Inter-*

national Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net. URL: <https://openreview.net/forum?id=xYGN0860WDH>.

Nitay Calderon, Roi Reichart, and Rotem Dror (2025). “The Alternative Annotator Test for LLM-as-a-Judge: How to Statistically Justify Replacing Human Annotators with LLMs”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics, pp. 16051–16081. URL: <https://aclanthology.org/2025.acl-long.782/>.

José Camacho-Collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa Anke, Fangyu Liu, and Eugenio Martínez Cámara (2022). “TweetNLP: Cutting-Edge Natural Language Processing for Social Media”. In: *Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022 - System Demonstrations, Abu Dhabi, UAE, December 7-11, 2022*. Ed. by Wanxiang Che, and Ekaterina Shutova. Association for Computational Linguistics, pp. 38–49. DOI: 10.18653/V1/2022.EMNLP-DEMOS.5. URL: <https://doi.org/10.18653/v1/2022.emnlp-demos.5>.

Marc E. Canby, Adam Davies, Chirag Rastogi, and Julia Hockenmaier (2024). “Measuring the Reliability of Causal Probing Methods: Tradeoffs, Limitations, and the Plight of Nullifying Interventions”. In: *CoRR* abs/2408.15510. DOI: 10.48550/ARXIV.2408.15510. arXiv: 2408.15510. URL: <https://doi.org/10.48550/arXiv.2408.15510>.

Pimwadee Chaovalit, and Lina Zhou (2005). “Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches”. In: *38th Hawaii International Conference on System Sciences (HICSS-38 2005), CD-ROM / Abstracts Proceedings, 3-6 January 2005, Big Island, HI, USA*. IEEE Computer Society. DOI: 10.1109/HICSS.2005.445. URL: <https://doi.org/10.1109/HICSS.2005.445>.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye (2024a). “INSIDE: LLMs’ Internal States Retain the Power of Hallucination Detection”. In: *The Twelfth International Conference on Learning Rep-*

- resentations, *ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. URL: <https://openreview.net/forum?id=Zj12nzlQbz>.
- Jiajun Chen, Huarui He, Feng Wu, and Jie Wang (2021). “Topology-Aware Correlations Between Relations for Inductive Link Prediction in Knowledge Graphs”. In: *AAAI*. AAAI Press, pp. 6271–6278.
- Nuo Chen, Ning Wu, Shining Liang, Ming Gong, Linjun Shou, Dongmei Zhang, and Jia Li (2023). “Is Bigger and Deeper Always Better? Probing LLaMA Across Scales and Layers”. In: *CoRR* abs/2312.04333. DOI: 10.48550/ARXIV.2312.04333. arXiv: 2312.04333. URL: <https://doi.org/10.48550/arXiv.2312.04333>.
- Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen (2022). “KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction”. In: *WWW ’22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*. Ed. by Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini. ACM, pp. 2778–2788. DOI: 10.1145/3485447.3511998. URL: <https://doi.org/10.1145/3485447.3511998>.
- Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou (2024b). “Teaching Large Language Models to Self-Debug”. In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. URL: <https://openreview.net/forum?id=KuPixIqPiq>.
- Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass (2024). “Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps”. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Association for Computational Linguistics, pp. 1419–1436. DOI: 10.18653/V1/2024.EMNLP-MAIN.84. URL: <https://doi.org/10.18653/v1/2024.emnlp-main.84>.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning (2020). “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators”. In: *8th International Conference on Learning Representations*,

- ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: <https://openreview.net/forum?id=r1xMH1BtvB>.
- Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso (2023). “Towards Automated Circuit Discovery for Mechanistic Interpretability”. In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Ed. by Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Abstract-Conference.html.
- Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loic Barrault, and Marco Baroni (2018). “What you can cram into a single vector: Probing sentence embeddings for linguistic properties”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Ed. by Iryna Gurevych, and Yusuke Miyao. Association for Computational Linguistics, pp. 2126–2136. DOI: 10.18653/V1/P18-1198. URL: <https://aclanthology.org/P18-1198/>.
- Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E. Ho (2024). “Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models”. In: *CoRR* abs/2401.01301. DOI: 10.48550/ARXIV.2401.01301. arXiv: 2401.01301. URL: <https://doi.org/10.48550/arXiv.2401.01301>.
- Daniel Daza, Michael Cochez, and Paul Groth (2021). “Inductive Entity Representations from Text via Link Prediction”. In: *WWW. ACM / IW3C2*, pp. 798–808.
- Alessandro De Bellis (2023). “Structuring the Unstructured: an LLM-guided Transition”. In: *Proceedings of the Doctoral Consortium at ISWC 2023 co-located with 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 7, 2023*. Ed. by Claudia d’Amato, and Jeff Z. Pan. Vol. 3678. CEUR Workshop Proceedings. CEUR-WS.org. URL: <https://ceur-ws.org/Vol-3678/paper12.pdf>.
- Alessandro De Bellis, Vito Walter Anelli, Tommaso Di Noia, and Eugenio Di Sciascio (2024). “PRONTO: Prompt-Based Detection of Semantic Containment Patterns in MLMs”. In: *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024*,

- Proceedings, Part II*. Ed. by Gianluca Demartini, Katja Hose, Maribel Acosta, Matteo Palmonari, Gong Cheng, Hala Skaf-Molli, Nicolas Ferranti, Daniel Hernández, and Aidan Hogan. Vol. 15232. Lecture Notes in Computer Science. Springer, pp. 227–246. DOI: 10.1007/978-3-031-77850-6_13. URL: https://doi.org/10.1007/978-3-031-77850-6_13.
- Alessandro De Bellis, Giovanni Maria Biancofiore, Vito Walter Anelli, Fedelucio Narducci, Tommaso Di Noia, Azzurra Ragone, and Eugenio Di Sciascio (2023). “Semantic Interpretation of BERT Embeddings with Knowledge Graphs”. In: *Proceedings of the 31st Symposium of Advanced Database Systems, Galzingano Terme, Italy, July 2nd to 5th, 2023*. Ed. by Diego Calvanese, Claudia Diamantini, Guglielmo Faggioli, Nicola Ferro, Stefano Marchesin, Gianmaria Silvello, and Letizia Tanca. Vol. 3478. CEUR Workshop Proceedings. CEUR-WS.org, pp. 181–191. URL: <https://ceur-ws.org/Vol-3478/paper69.pdf>.
- Alessandro De Bellis, Salvatore Bui, Giovanni Servedio, Vito Walter Anelli, Tommaso Di Noia, and Eugenio Di Sciascio (Nov. 2025). “Type-Less yet Type-Aware Inductive Link Prediction with Pretrained Language Models”. In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Ed. by Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng. Suzhou, China: Association for Computational Linguistics, pp. 27181–27197. ISBN: 979-8-89176-332-6. URL: <https://aclanthology.org/2025.emnlp-main.1383/>.
- Xiang Deng, Vasilisa Bashlovkina, Feng Han, Simon Baumgartner, and Michael Bendersky (2023). “LLMs to the Moon? Reddit Market Sentiment Analysis with Large Language Models”. In: *Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*. Ed. by Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben. ACM, pp. 1014–1019. DOI: 10.1145/3543873.3587605. URL: <https://doi.org/10.1145/3543873.3587605>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and

- Thamar Solorio. Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/V1/N19-1423. URL: <https://doi.org/10.18653/v1/n19-1423>.
- Dario Di Palma (2023). “Retrieval-augmented Recommender System: Enhancing Recommender Systems with Large Language Models”. In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, pp. 1369–1373. DOI: 10.1145/3604915.3608889. URL: <https://doi.org/10.1145/3604915.3608889>.
- Dario Di Palma, Giovanni Maria Biancofiore, Vito Walter Anelli, Fedelucio Narducci, and Tommaso Di Noia (2025a). “Content-Based or Collaborative? Insights from Inter-List Similarity Analysis of ChatGPT Recommendations”. In: *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization, UMAP Adjunct 2025, New York City, NY, USA, June 16-19, 2025*. ACM, pp. 28–33. DOI: 10.1145/3708319.3733680. URL: <https://doi.org/10.1145/3708319.3733680>.
- Dario Di Palma, Alessandro De Bellis, Giovanni Servedio, Vito Walter Anelli, Fedelucio Narducci, and Tommaso Di Noia (2025b). “LLaMAs Have Feelings Too: Unveiling Sentiment and Emotion Representations in LLaMA Models Through Probing”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics, pp. 6124–6142. URL: <https://aclanthology.org/2025.acl-long.306/>.
- Dario Di Palma, Felice Antonio Merra, Maurizio Sfilio, Vito Walter Anelli, Fedelucio Narducci, and Tommaso Di Noia (2025c). “Do LLMs Memorize Recommendation Datasets? A Preliminary Study on MovieLens-1M”. In: *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*. Ed. by Nicola Ferro, Maria Maistro, Gabriella Pasi, Omar Alonso, Andrew Trotman, and Suzan Verberne. ACM, pp. 2582–2586. DOI: 10.1145/3726302.3730178. URL: <https://doi.org/10.1145/3726302.3730178>.
- Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Xiaobin Wang, Pengjun Xie, Haitao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim (2022). “Prompt-learning for

- Fine-grained Entity Typing”. In: *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Association for Computational Linguistics, pp. 6888–6901. DOI: 10.18653/v1/2022.FINDINGS-EMNLP.512. URL: <https://doi.org/10.18653/v1/2022.findings-emnlp.512>.
- Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu (2021). “Few-NERD: A Few-shot Named Entity Recognition Dataset”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Association for Computational Linguistics, pp. 3198–3213. DOI: 10.18653/v1/2021.ACL-LONG.248. URL: <https://doi.org/10.18653/v1/2021.acl-long.248>.
- Michael Duan, Anshuman Suri, Niloofar Miresghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi (2024). “Do Membership Inference Attacks Work on Large Language Models?” In: *CoRR* abs/2402.07841. DOI: 10.48550/ARXIV.2402.07841. arXiv: 2402.07841. URL: <https://doi.org/10.48550/arXiv.2402.07841>.
- Abhimanyu Dubey et al. (2024a). “The Llama 3 Herd of Models”. In: *CoRR* abs/2407.21783. DOI: 10.48550/ARXIV.2407.21783. arXiv: 2407.21783. URL: <https://doi.org/10.48550/arXiv.2407.21783>.
- Abhimanyu Dubey et al. (2024b). “The Llama 3 Herd of Models”. In: *CoRR* abs/2407.21783.
- Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger B. Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah (2022). “Toy Models of Superposition”. In: *CoRR* abs/2209.10652. DOI: 10.48550/ARXIV.2209.10652. arXiv: 2209.10652. URL: <https://doi.org/10.48550/arXiv.2209.10652>.
- Kawin Ethayarajh (2019). “How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language*

- Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, pp. 55–65. DOI: 10.18653/V1/D19-1006. URL: <https://doi.org/10.18653/v1/D19-1006>.
- Georgios Fatouros, John Soldatos, Kalliopi Kouroumali, Georgios Makridis, and Dimosthenis Kyriazis (2023). “Transforming Sentiment Analysis in the Financial Domain with ChatGPT”. In: *CoRR* abs/2308.07935. DOI: 10.48550/ARXIV.2308.07935. arXiv: 2308.07935. URL: <https://doi.org/10.48550/arXiv.2308.07935>.
- Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov (2024). “Don’t Hallucinate, Abstain: Identifying LLM Knowledge Gaps via Multi-LLM Collaboration”. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Association for Computational Linguistics, pp. 14664–14690. DOI: 10.18653/V1/2024.ACL-LONG.786. URL: <https://doi.org/10.18653/v1/2024.acl-long.786>.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen (2021). “SimCSE: Simple Contrastive Learning of Sentence Embeddings”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Association for Computational Linguistics, pp. 6894–6910. DOI: 10.18653/V1/2021.EMNLP-MAIN.552. URL: <https://doi.org/10.18653/v1/2021.emnlp-main.552>.
- Andrés García-Silva, Cristian Berrio, and José Manuel Gómez-Pérez (2023). “Textual Entailment for Effective Triple Validation in Object Prediction”. In: *The Semantic Web - ISWC 2023 - 22nd International Semantic Web Conference, Athens, Greece, November 6-10, 2023, Proceedings, Part I*. Ed. by Terry R. Payne, Valentina Presutti, Guilin Qi, María Poveda-Villalón, Giorgos Stoilos, Laura Hollink, Zoi Kaoudi, Gong Cheng, and Juanzi Li. Vol. 14265. Lecture Notes in Computer Science. Springer, pp. 80–100. DOI: 10.1007/978-3-031-47240-4_5. URL: https://doi.org/10.1007/978-3-031-47240-4_5.
- Zorik Gekhman, Eyal Ben-David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart (2025). “Inside-Out: Hidden

- Factual Knowledge in LLMs”. In: *CoRR* abs/2503.15299. DOI: 10.48550/ARXIV.2503.15299. arXiv: 2503.15299. URL: <https://doi.org/10.48550/arXiv.2503.15299>.
- Aryo Pradipta Gema, Dominik Grabarczyk, Wolf De Wulf, Piyush Borole, Javier Antonio Alfaro, Pasquale Minervini, Antonio Vergari, and Ajitha Rajan (2023). “Knowledge Graph Embeddings in the Biomedical Domain: Are They Useful? A Look at Link Prediction, Rule Learning, and Downstream Polypharmacy Tasks”. In: *CoRR* abs/2305.19979.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy (Nov. 2021). “Transformer Feed-Forward Layers Are Key-Value Memories”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 5484–5495. DOI: 10.18653/v1/2021.emnlp-main.446. URL: <https://aclanthology.org/2021.emnlp-main.446/>.
- Yusra Ghafoor, Shi Jinping, Fernando H. Calderon, Yen-Hao Huang, Kuan-Ta Chen, and Yi-Shin Chen (2023). “TERMS: textual emotion recognition in multidimensional space”. In: *Appl. Intell.* 53.3, pp. 2673–2693. DOI: 10.1007/S10489-022-03567-4. URL: <https://doi.org/10.1007/s10489-022-03567-4>.
- Tanya Goyal, Junyi Jessy Li, and Greg Durrett (2022). “News Summarization and Evaluation in the Era of GPT-3”. In: *CoRR* abs/2209.12356. DOI: 10.48550/ARXIV.2209.12356. arXiv: 2209.12356. URL: <https://doi.org/10.48550/arXiv.2209.12356>.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo (2024). “A Survey on LLM-as-a-Judge”. In: *CoRR* abs/2411.15594.
- Wes Gurnee, and Max Tegmark (2024). “Language Models Represent Space and Time”. In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. URL: <https://openreview.net/forum?id=jE8xbmvFin>.
- William L. Hamilton, Zhitao Ying, and Jure Leskovec (2017). “Inductive Representation Learning on Large Graphs”. In: *NIPS*, pp. 1024–1034.
- Shibo Hao, Bowen Tan, Kaiwen Tang, Bin Ni, Xiyan Shao, Hengzhe Zhang, Eric P. Xing, and Zhiting Hu (2023). “BertNet: Harvesting Knowledge Graphs with

- Arbitrary Relations from Pretrained Language Models”. In: *ACL (Findings)*. Association for Computational Linguistics, pp. 5000–5015.
- Momchil Hardalov, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nakov (Nov. 2020). “EXAMS: A Multi-subject High School Examinations Dataset for Cross-lingual and Multilingual Question Answering”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 5427–5444. DOI: 10.18653/v1/2020.emnlp-main.438. URL: <https://aclanthology.org/2020.emnlp-main.438/>.
- Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis (2023). “Leveraging Large Language Models for Sequential Recommendation”. In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, pp. 1096–1102. DOI: 10.1145/3604915.3610639. URL: <https://doi.org/10.1145/3604915.3610639>.
- Md. Arif Hasan, Shudipta Das, Afijat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori (2024). “Zero- and Few-Shot Prompting with LLMs: A Comparative Study with Fine-tuned Models for Bangla Sentiment Analysis”. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*. Ed. by Nicoletta Calzolari, Min-Yen Kan, Véronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue. ELRA and ICCL, pp. 17808–17818. URL: <https://aclanthology.org/2024.lrec-main.1549>.
- Vasileios Hatzivassiloglou, and Kathleen R. McKeown (1997). “Predicting the Semantic Orientation of Adjectives”. In: *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, 7-12 July 1997, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain*. Ed. by Philip R. Cohen, and Wolfgang Wahlster. Morgan Kaufmann Publishers / ACL, pp. 174–181. DOI: 10.3115/976909.979640. URL: <https://aclanthology.org/P97-1023/>.

- Pengcheng He, Jianfeng Gao, and Weizhu Chen (2023). “DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing”. In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. URL: <https://openreview.net/forum?id=sE7-XhLxHA>.
- Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk (2016). “Session-based Recommendations with Recurrent Neural Networks”. In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio, and Yann LeCun. URL: <http://arxiv.org/abs/1511.06939>.
- Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d’Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann (2021). *Knowledge Graphs*. Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool Publishers.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi (2020). “The Curious Case of Neural Text Degeneration”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: <https://openreview.net/forum?id=rygGQyrFvH>.
- Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian J. McAuley (2024). “Bridging Language and Items for Retrieval and Recommendation”. In: *CoRR* abs/2403.03952. DOI: 10.48550/ARXIV.2403.03952. arXiv: 2403.03952. URL: <https://doi.org/10.48550/arXiv.2403.03952>.
- Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen (2022). “Towards Universal Sequence Representation Learning for Recommender Systems”. In: *KDD ’22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*. Ed. by Aidong Zhang, and Huzefa Rangwala. ACM, pp. 585–593. DOI: 10.1145/3534678.3539381. URL: <https://doi.org/10.1145/3534678.3539381>.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly (2019).

- “Parameter-Efficient Transfer Learning for NLP”. In: *ICML*. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 2790–2799.
- Xinshuo Hu, Zifei Shan, Xinping Zhao, Zetian Sun, Zhenyu Liu, Dongfang Li, Shaolin Ye, Xinyuan Wei, Qian Chen, Baotian Hu, Haofen Wang, Jun Yu, and Min Zhang (2025). “KaLM-Embedding: Superior Training Data Brings A Stronger Embedding Model”. In: *CoRR* abs/2501.01028. DOI: 10.48550/ARXIV.2501.01028. arXiv: 2501.01028. URL: <https://doi.org/10.48550/arXiv.2501.01028>.
- Jiaxin Huang, Yu Meng, and Jiawei Han (2022). “Few-Shot Fine-Grained Entity Typing with Automatic Label Interpretation and Instance Generation”. In: *KDD ’22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*. Ed. by Aidong Zhang, and Huzefa Rangwala. ACM, pp. 605–614. DOI: 10.1145/3534678.3539443. URL: <https://doi.org/10.1145/3534678.3539443>.
- Devansh Jain, and Luis Espinosa Anke (2022). “Distilling Hypernymy Relations from Language Models: On the Effectiveness of Zero-Shot Taxonomy Induction”. In: *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2022, Seattle, WA, USA, July 14-15, 2022*. Ed. by Vivi Nastase, Ellie Pavlick, Mohammad Taher Pilehvar, José Camacho-Collados, and Alessandro Raganato. Association for Computational Linguistics, pp. 151–156. DOI: 10.18653/V1/2022.STARSEM-1.13. URL: <https://doi.org/10.18653/v1/2022.starsem-1.13>.
- Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung (2023). “Towards Mitigating LLM Hallucination via Self Reflection”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Association for Computational Linguistics, pp. 1827–1843. DOI: 10.18653/V1/2023.FINDINGS-EMNLP.123. URL: <https://doi.org/10.18653/v1/2023.findings-emnlp.123>.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed (2023). “Mistral 7B”. In: *CoRR* abs/2310.06825.

- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig (2020). “How Can We Know What Language Models Know”. In: *Trans. Assoc. Comput. Linguistics* 8, pp. 423–438. DOI: 10.1162/TACL\A_00324. URL: https://doi.org/10.1162/tacl%5C_a%5C_00324.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer (2017). “TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. Ed. by Regina Barzilay, and Min-Yen Kan. Association for Computational Linguistics, pp. 1601–1611. DOI: 10.18653/V1/P17-1147. URL: <https://doi.org/10.18653/v1/P17-1147>.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan (2022). “Language Models (Mostly) Know What They Know”. In: *CoRR* abs/2207.05221. DOI: 10.48550/ARXIV.2207.05221. arXiv: 2207.05221. URL: <https://doi.org/10.48550/arXiv.2207.05221>.
- Wang-Cheng Kang, and Julian J. McAuley (2018). “Self-Attentive Sequential Recommendation”. In: *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, pp. 197–206. DOI: 10.1109/ICDM.2018.00035. URL: <https://doi.org/10.1109/ICDM.2018.00035>.
- Nora Kassner, Philipp Dufter, and Hinrich Schütze (2021). “Multilingual LAMA: Investigating Knowledge in Multilingual Pretrained Language Models”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*. Ed. by Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty. Association for Computational Linguistics, pp. 3250–3258. DOI: 10.18653/V1/2021.EACL-MAIN.284. URL: <https://doi.org/10.18653/v1/2021.eacl-main.284>.
- Michael Katz, Harsha Kokel, Kavitha Srinivas, and Shirin Sohrabi (2024). “Thought of Search: Planning with Language Models Through The Lens of Efficiency”.

- In: *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*. Ed. by Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang. URL: http://papers.nips.cc/paper%5C_files/paper/2024/hash/fa080fe0f218871faec1d8ba20e491d5-Abstract-Conference.html.
- Khairullah Khan, Baharum Baharudin, Aurangzeb Khan, and Ashraf Ullah (2014). “Mining opinion components from unstructured reviews: A review”. In: *J. King Saud Univ. Comput. Inf. Sci.* 26.3, pp. 258–275. DOI: 10.1016/J.JKSUCI.2014.03.009. URL: <https://doi.org/10.1016/j.jksuci.2014.03.009>.
- Omar Khattab, and Matei Zaharia (2020). “ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT”. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. Ed. by Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu. ACM, pp. 39–48. DOI: 10.1145/3397271.3401075. URL: <https://doi.org/10.1145/3397271.3401075>.
- Yookyung Kho, Jaehee Kim, and Pilsung Kang (2023). “Boosting Prompt-Based Self-Training With Mapping-Free Automatic Verbalizer for Multi-Class Classification”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Association for Computational Linguistics, pp. 13786–13800. DOI: 10.18653/V1/2023.FINDINGS-EMNLP.921. URL: <https://doi.org/10.18653/v1/2023.findings-emnlp.921>.
- Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh (2023). “ProPILE: Probing Privacy Leakage in Large Language Models”. In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Ed. by Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/420678bb4c8251ab30e765bc27c3b047-Abstract-Conference.html.
- Su Young Kim, Hyeon-Jin Park, Kyuyong Shin, and Kyung-Min Kim (2022). “Ask Me What You Need: Product Retrieval using Knowledge from GPT-3”. In:

- CoRR* abs/2207.02516. DOI: 10.48550/ARXIV.2207.02516. arXiv: 2207.02516. URL: <https://doi.org/10.48550/arXiv.2207.02516>.
- Yoon Kim (2014). “Convolutional Neural Networks for Sentence Classification”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. ACL, pp. 1746–1751. DOI: 10.3115/V1/D14-1181. URL: <https://doi.org/10.3115/v1/d14-1181>.
- Filip Klubicka, and John D. Kelleher (2022). “Probing with Noise: Unpicking the Warp and Weft of Embeddings”. In: *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 8, 2022*. Ed. by Jasmijn Bastings, Yonatan Belinkov, Yanai Elazar, Dieuwke Hupkes, Naomi Saphra, and Sarah Wiegrefe. Association for Computational Linguistics, pp. 404–417. DOI: 10.18653/V1/2022.BLACKBOXNLP-1.34. URL: <https://doi.org/10.18653/v1/2022.blackboxnlp-1.34>.
- Jan Ole Krugmann, and Jochen Hartmann (2024). “Sentiment Analysis in the Age of Generative AI”. In: *Customer Needs and Solutions* 11.1, p. 3.
- Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadi (2022). “Matryoshka Representation Learning”. In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh. URL: http://papers.nips.cc/paper%5C_files/paper/2022/hash/c32319f4868da7613d78af9993100e42-Abstract-Conference.html.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020). “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: <https://openreview.net/forum?id=H1eA7AEtvS>.
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping (2025). “NV-Embed: Improved Techniques

- for Training LLMs as Generalist Embedding Models”. In: *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. URL: <https://openreview.net/forum?id=lgsyLSsDRe>.
- Benjamin A. Levinstein, and Daniel A. Herrmann (2023). “Still No Lie Detector for Language Models: Probing Empirical and Conceptual Roadblocks”. In: *CoRR* abs/2307.00175. DOI: 10.48550/ARXIV.2307.00175. arXiv: 2307.00175. URL: <https://doi.org/10.48550/arXiv.2307.00175>.
- Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li (2020). “On the Sentence Embeddings from Pre-trained Language Models”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics, pp. 9119–9130. DOI: 10.18653/V1/2020.EMNLP-MAIN.733. URL: <https://doi.org/10.18653/v1/2020.emnlp-main.733>.
- Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem (2023a). “CAMEL: Communicative Agents for "Mind" Exploration of Large Scale Language Model Society”. In: *CoRR* abs/2303.17760. DOI: 10.48550/ARXIV.2303.17760. arXiv: 2303.17760. URL: <https://doi.org/10.48550/arXiv.2303.17760>.
- Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao (Nov. 2024). “Revisiting Catastrophic Forgetting in Large Language Model Tuning”. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 4297–4308. DOI: 10.18653/v1/2024.findings-emnlp.249. URL: <https://aclanthology.org/2024.findings-emnlp.249/>.
- Kenneth Li, Oam Patel, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg (2023b). “Inference-Time Intervention: Eliciting Truthful Answers from a Language Model”. In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Ed. by Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

- Muzhi Li, Cehao Yang, Chengjin Xu, Zixing Song, Xuhui Jiang, Jian Guo, Ho-fung Leung, and Irwin King (2025). “Context-aware Inductive Knowledge Graph Completion with Latent Type Constraints and Subgraph Reasoning”. In: *AAAI*. AAAI Press, pp. 12102–12111.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang (2023c). “Towards General Text Embeddings with Multi-stage Contrastive Learning”. In: *CoRR* abs/2308.03281. DOI: 10.48550/ARXIV.2308.03281. arXiv: 2308.03281. URL: <https://doi.org/10.48550/arXiv.2308.03281>.
- Zongxia Li, Paiheng Xu, Fuxiao Liu, and Hyemi Song (2023d). “Towards Understanding In-Context Learning with Contrastive Demonstrations and Saliency Maps”. In: *CoRR* abs/2307.05052. DOI: 10.48550/ARXIV.2307.05052. arXiv: 2307.05052. URL: <https://doi.org/10.48550/arXiv.2307.05052>.
- Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang (Aug. 2024). “Learning to Trust Your Feelings: Leveraging Self-awareness in LLMs for Hallucination Mitigation”. In: *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*. Ed. by Wenhao Yu, Weijia Shi, Michihiro Yasunaga, Meng Jiang, Chenguang Zhu, Hannaneh Hajishirzi, Luke Zettlemoyer, and Zhihan Zhang. Bangkok, Thailand: Association for Computational Linguistics, pp. 44–58. DOI: 10.18653/v1/2024.knowledgenlp-1.4. URL: <https://aclanthology.org/2024.knowledgenlp-1.4/>.
- Shuying Lin, Huosheng Xie, Liang-Chih Yu, and K. Robert Lai (2017a). “SentiNLP at IJCNLP-2017 Task 4: Customer Feedback Analysis Using a Bi-LSTM-CNN Model”. In: *Proceedings of the IJCNLP 2017, Shared Tasks, Taipei, Taiwan, November 27 - December 1, 2017, Shared Tasks*. Ed. by Chao-Hong Liu, Preslav Nakov, and Nianwen Xue. Asian Federation of Natural Language Processing, pp. 149–154. URL: <https://aclanthology.org/I17-4025/>.
- Stephanie Lin, Jacob Hilton, and Owain Evans (2022). “TruthfulQA: Measuring How Models Mimic Human Falsehoods”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, pp. 3214–3252. DOI: 10.18653/v1/2022.acl-long.229. URL: <https://doi.org/10.18653/v1/2022.acl-long.229>.

- Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár (2017b). “Focal Loss for Dense Object Detection”. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, pp. 2999–3007. DOI: 10.1109/ICCV.2017.324. URL: <https://doi.org/10.1109/ICCV.2017.324>.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu (2015). “Learning Entity and Relation Embeddings for Knowledge Graph Completion”. In: *AAAI*. AAAI Press, pp. 2181–2187.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *CoRR* abs/1907.11692. arXiv: 1907.11692. URL: <http://arxiv.org/abs/1907.11692>.
- Zhiheng Lyu, Zhijing Jin, Fernando Gonzalez Adauro, Rada Mihalcea, Bernhard Schölkopf, and Mrinmaya Sachan (2024). “Do LLMs Think Fast and Slow? A Causal Study on Sentiment Analysis”. In: *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Association for Computational Linguistics, pp. 9353–9372. DOI: 10.18653/V1/2024.FINDINGS-EMNLP.547. URL: <https://doi.org/10.18653/v1/2024.findings-emnlp.547>.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts (2011). “Learning Word Vectors for Sentiment Analysis”. In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. The Association for Computer Linguistics, pp. 142–150. URL: <https://aclanthology.org/P11-1015/>.
- Anemily Machina, and Robert E. Mercer (2024). “Anisotropy is Not Inherent to Transformers”. In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*. Ed. by Kevin Duh, Helena Gómez-Adorno, and Steven Bethard. Association for Computational Linguistics, pp. 4892–4907. DOI: 10.18653/V1/2024.NAACL-LONG.274. URL: <https://doi.org/10.18653/v1/2024.naacl-long.274>.

- Sijie Mai, Shuangjia Zheng, Yuedong Yang, and Haifeng Hu (2021). “Communicative Message Passing for Inductive Relation Reasoning”. In: *AAAI*. AAAI Press, pp. 4294–4302.
- Alex Mallen, and Nora Belrose (2023). “Eliciting Latent Knowledge from Quirky Language Models”. In: *CoRR* abs/2312.01037. DOI: 10.48550/ARXIV.2312.01037. arXiv: 2312.01037. URL: <https://doi.org/10.48550/arXiv.2312.01037>.
- Francesco Manigrasso, Stefan F. Schouten, Lia Morra, and Peter Bloem (2024). “Probing LLMs for Logical Reasoning”. In: *Neural-Symbolic Learning and Reasoning - 18th International Conference, NeSy 2024, Barcelona, Spain, September 9-12, 2024, Proceedings, Part I*. Ed. by Tarek R. Besold, Artur d’Avila Garcez, Ernesto Jiménez-Ruiz, Roberto Confalonieri, Pranava Madhyastha, and Benedikt Wagner. Vol. 14979. Lecture Notes in Computer Science. Springer, pp. 257–278. DOI: 10.1007/978-3-031-71167-1_14. URL: https://doi.org/10.1007/978-3-031-71167-1_14.
- Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy (2020). “Emergent linguistic structure in artificial neural networks trained by self-supervision”. In: *Proc. Natl. Acad. Sci. USA* 117.48, pp. 30046–30054. DOI: 10.1073/PNAS.1907367117. URL: <https://doi.org/10.1073/pnas.1907367117>.
- Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller (2025). “Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models”. In: *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. URL: <https://openreview.net/forum?id=I4e82CIDxv>.
- Samuel Marks, and Max Tegmark (2023). “The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets”. In: *CoRR* abs/2310.06824. DOI: 10.48550/ARXIV.2310.06824. arXiv: 2310.06824. URL: <https://doi.org/10.48550/arXiv.2310.06824>.
- Christian Meilicke, Manuel Fink, Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, and Heiner Stuckenschmidt (2018). “Fine-Grained Evaluation of Rule- and Embedding-Based Systems for Knowledge Graph Completion”. In: *ISWC (1)*. Vol. 11136. Lecture Notes in Computer Science. Springer, pp. 3–20.

- Pablo Mendes, Max Jakob, and Christian Bizer (May 2012). “DBpedia: A Multilingual Cross-domain Knowledge Base”. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Istanbul, Turkey: European Language Resources Association (ELRA), pp. 1813–1817. URL: <https://aclanthology.org/L12-1323/>.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov (2022). “Locating and Editing Factual Associations in GPT”. In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh. URL: http://papers.nips.cc/paper%5C_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.
- Simone Merlo, Guglielmo Faggioli, and Nicola Ferro (2025). “A Reproducibility Study for Joint Information Retrieval and Recommendation in Product Search”. In: *Advances in Information Retrieval - 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6-10, 2025, Proceedings, Part IV*. Ed. by Claudia Hauff, Craig Macdonald, Dietmar Jannach, Gabriella Kazai, Franco Maria Nardini, Fabio Pinelli, Fabrizio Silvestri, and Nicola Tonellotto. Vol. 15575. Lecture Notes in Computer Science. Springer, pp. 130–145. DOI: 10.1007/978-3-031-88717-8_10. URL: https://doi.org/10.1007/978-3-031-88717-8_10.
- Felice Antonio Merra, Omar Zaidan, and Fabricio de Sousa Nascimento (2023). “Improving the Relevance of Product Search for Queries with Negations”. In: *Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*. Ed. by Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben. ACM, pp. 86–89. DOI: 10.1145/3543873.3587319. URL: <https://doi.org/10.1145/3543873.3587319>.
- Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). “Efficient Estimation of Word Representations in Vector Space”. In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona,*

- USA, May 2-4, 2013, Workshop Track Proceedings. Ed. by Yoshua Bengio, and Yann LeCun. URL: <http://arxiv.org/abs/1301.3781>.
- Tomás Mikolov, Wen-tau Yih, and Geoffrey Zweig (2013b). “Linguistic Regularities in Continuous Space Word Representations”. In: *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*. Ed. by Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff. The Association for Computational Linguistics, pp. 746–751. URL: <https://aclanthology.org/N13-1090/>.
- Roser Morante, and Eduardo Blanco (2012). “*SEM 2012 Shared Task: Resolving the Scope and Focus of Negation”. In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics, *SEM 2012, June 7-8, 2012, Montréal, Canada*. Ed. by Eneko Agirre, Johan Bos, and Mona T. Diab. Association for Computational Linguistics, pp. 265–274. URL: <https://aclanthology.org/S12-1035/>.
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers (2023). “MTEB: Massive Text Embedding Benchmark”. In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*. Ed. by Andreas Vlachos, and Isabelle Augenstein. Association for Computational Linguistics, pp. 2006–2029. DOI: 10.18653/V1/2023.EACL-MAIN.148. URL: <https://doi.org/10.18653/v1/2023.eacl-main.148>.
- Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang (2022a). “Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models”. In: *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, pp. 1864–1874. DOI: 10.18653/V1/2022.FINDINGS-ACL.146. URL: <https://doi.org/10.18653/v1/2022.findings-acl.146>.
- Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang (2022b). “Large Dual Encoders Are Generalizable Retrievers”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Association for Computa-

- tional Linguistics, pp. 9844–9855. DOI: 10.18653/V1/2022.EMNLP-MAIN.669. URL: <https://doi.org/10.18653/v1/2022.emnlp-main.669>.
- Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov (2025). “Arithmetic Without Algorithms: Language Models Solve Math with a Bag of Heuristics”. In: *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. URL: <https://openreview.net/forum?id=09YTt26r2P>.
- Jingcheng Niu, Xingdi Yuan, Tong Wang, Hamidreza Saghir, and Amir H. Abdi (2025). “Llama See, Llama Do: A Mechanistic Perspective on Contextual Entrainment and Distraction in LLMs”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics, pp. 16218–16239. URL: <https://aclanthology.org/2025.acl-long.791/>.
- nostalgebraist (2020). *Interpreting GPT: The Logit Lens*. <https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru>.
- Chris Olah (2022). *Mechanistic Interpretability, Variables, and the Importance of Interpretable Bases — transformer-circuits.pub*. <https://www.transformer-circuits.pub/2022/mech-interp-essay>. [Accessed 07-10-2025].
- OpenAI (2023). “GPT-4 Technical Report”. In: *CoRR* abs/2303.08774.
- Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov (2025). “LLMs Know More Than They Show: On the Intrinsic Representation of LLM Hallucinations”. In: *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. URL: <https://openreview.net/forum?id=KRnsX5Em3W>.
- Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeljanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux (2023). “Large Language Models and Knowledge Graphs: Opportunities and Challenges”. In: *TGDK* 1.1, 2:1–2:38. DOI: 10.4230/TGDK.1.1.2. URL: <https://doi.org/10.4230/TGDK.1.1.2>.
- Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu (2024). “Unifying Large Language Models and Knowledge Graphs: A Roadmap”.

- In: *IEEE Trans. Knowl. Data Eng.* 36.7, pp. 3580–3599. DOI: 10.1109/TKDE.2024.3352100. URL: <https://doi.org/10.1109/TKDE.2024.3352100>.
- Bo Pang, and Lillian Lee (2005). “Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales”. In: *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. Ed. by Kevin Knight, Hwee Tou Ng, and Kemal Oflazer. The Association for Computer Linguistics, pp. 115–124. DOI: 10.3115/1219840.1219855. URL: <https://aclanthology.org/P05-1015/>.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan (2002). “Thumbs up? Sentiment Classification using Machine Learning Techniques”. In: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, Philadelphia, PA, USA, July 6-7, 2002*, pp. 79–86. DOI: 10.3115/1118693.1118704. URL: <https://aclanthology.org/W02-1011/>.
- Yannis Papanikolaou, and Andrea Pierleoni (2020). “DARE: Data Augmented Relation Extraction with GPT-2”. In: *CoRR* abs/2004.13845. arXiv: 2004.13845. URL: <https://arxiv.org/abs/2004.13845>.
- Vincenzo Paparella, Dario Di Palma, Vito Walter Anelli, and Tommaso Di Noia (2023). “Broadening the Scope: Evaluating the Potential of Recommender Systems beyond prioritizing Accuracy”. In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, pp. 1139–1145. DOI: 10.1145/3604915.3610649. URL: <https://doi.org/10.1145/3604915.3610649>.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay (2011). “Scikit-learn: Machine Learning in Python”. In: *J. Mach. Learn. Res.* 12, pp. 2825–2830. DOI: 10.5555/1953048.2078195. URL: <https://dl.acm.org/doi/10.5555/1953048.2078195>.
- Hao Peng, Xiaozhi Wang, Shengding Hu, Hailong Jin, Lei Hou, Juanzi Li, Zhiyuan Liu, and Qun Liu (2022). “COPEN: Probing Conceptual Knowledge in Pre-trained Language Models”. In: *Proceedings of the 2022 Conference on Empirical*

- Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Association for Computational Linguistics, pp. 5015–5035. DOI: 10.18653/V1/2022.EMNLP-MAIN.335. URL: <https://doi.org/10.18653/v1/2022.emnlp-main.335>.
- Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith (2019). “Knowledge Enhanced Contextual Word Representations”. In: *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, pp. 43–54.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. Ed. by Marilyn A. Walker, Heng Ji, and Amanda Stent. Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/V1/N18-1202. URL: <https://doi.org/10.18653/v1/n18-1202>.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller (2019). “Language Models as Knowledge Bases?” In: *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, pp. 2463–2473.
- Duy Khoa Pham, and Bao Quoc Vo (2024). “Towards Reliable Medical Question Answering: Techniques and Challenges in Mitigating Hallucinations in Language Models”. In: *CoRR* abs/2408.13808. DOI: 10.48550/ARXIV.2408.13808. arXiv: 2408.13808. URL: <https://doi.org/10.48550/arXiv.2408.13808>.
- Paulo Pirozelli, Marcos M. José, Paulo de Tarso P. Filho, Anarosa A. F. Brandão, and Fábio G. Cozman (2024). “Assessing Logical Reasoning Capabilities of Encoder-Only Transformer Models”. In: *Neural-Symbolic Learning and Reasoning - 18th International Conference, NeSy 2024, Barcelona, Spain, September 9-12, 2024, Proceedings, Part I*. Ed. by Tarek R. Besold, Artur d’Avila Garcez, Ernesto Jiménez-Ruiz, Roberto Confalonieri, Pranava Madhyastha, and Benedikt Wagner. Vol. 14979. Lecture Notes in Computer Science. Springer, pp. 29–46. DOI: 10.1007/978-3-031-71167-1_2. URL: https://doi.org/10.1007/978-3-031-71167-1%5C_2.

- Nina Pörner, Ulli Waltinger, and Hinrich Schütze (2020). “E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Vol. EMNLP 2020. Findings of ACL. Association for Computational Linguistics, pp. 803–818. DOI: 10.18653/V1/2020.FINDINGS-EMNLP.71. URL: <https://doi.org/10.18653/v1/2020.findings-emnlp.71>.
- Jacob P. Portes, Alexander Trott, Sam Havens, Daniel King, Abhinav Venigalla, Moin Nadeem, Nikhil Sardana, Daya Khudia, and Jonathan Frankle (2023). “MosaicBERT: A Bidirectional Encoder Optimized for Fast Pretraining”. In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Ed. by Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/095a6917768712b7ccc61acbeecad1d8-Abstract-Conference.html.
- Jay Pujara, Eriq Augustine, and Lise Getoor (2017). “Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short”. In: *EMNLP*. Association for Computational Linguistics, pp. 1751–1756.
- Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang (2023). “Is ChatGPT a General-Purpose Natural Language Processing Task Solver?” In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Association for Computational Linguistics, pp. 1339–1384. DOI: 10.18653/V1/2023.EMNLP-MAIN.85. URL: <https://doi.org/10.18653/v1/2023.emnlp-main.85>.
- Guanghui Qin, and Jason Eisner (June 2021). “Learning How to Ask: Querying LMs with Mixtures of Soft Prompts”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Online: Association for Computational Linguistics, pp. 5203–5212. DOI: 10.18653/v1/2021.naacl-main.410. URL: <https://aclanthology.org/2021.naacl-main.410>.

- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). “Language Models are Unsupervised Multitask Learners”. In.
- 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”. In: *J. Mach. Learn. Res.* 21, 140:1–140:67. URL: <https://jmlr.org/papers/v21/20-074.html>.
- Sara Rajaei, and Mohammad Taher Pilehvar (2021). “How Does Fine-tuning Affect the Geometry of Embedding Space: A Case Study on Isotropy”. In: *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Association for Computational Linguistics, pp. 3042–3049. DOI: 10.18653/V1/2021.FINDINGS-EMNLP.261. URL: <https://doi.org/10.18653/v1/2021.findings-emnlp.261>.
- Pranav Rajpurkar, Robin Jia, and Percy Liang (2018). “Know What You Don’t Know: Unanswerable Questions for SQuAD”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*. Ed. by Iryna Gurevych, and Yusuke Miyao. Association for Computational Linguistics, pp. 784–789. DOI: 10.18653/V1/P18-2124. URL: <https://aclanthology.org/P18-2124/>.
- Abhilasha Ravichander, Shruti Ghela, David Wadden, and Yejin Choi (2025). “HALoGEN: Fantastic LLM Hallucinations and Where to Find Them”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics, pp. 1402–1425. URL: <https://aclanthology.org/2025.acl-long.71/>.
- Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian (2022). “Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search”. In: *CoRR* abs/2206.06588. DOI: 10.48550/ARXIV.2206.06588. arXiv: 2206.06588. URL: <https://doi.org/10.48550/arXiv.2206.06588>.

- N. Rimsky (2023). *Reducing Sycophancy and Improving Honesty via Activation Steering*. <https://www.lesswrong.com/posts/zt6hRsDE84HeBK7E>.
- Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo (2021). “Knowledge Graph Embedding for Link Prediction: A Comparative Analysis”. In: *ACM Trans. Knowl. Discov. Data* 15.2, 14:1–14:49.
- Sarthak Roy, Ashish Harshavardhan, Animesh Mukherjee, and Punyajoy Saha (2023). “Probing LLMs for hate speech detection: strengths and vulnerabilities”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Association for Computational Linguistics, pp. 6116–6128. DOI: 10.18653/V1/2023.FINDINGS-EMNLP.407. URL: <https://doi.org/10.18653/v1/2023.findings-emnlp.407>.
- William Rudman, Nate Gillman, Taylor Rayne, and Carsten Eickhoff (2022). “IsoScore: Measuring the Uniformity of Embedding Space Utilization”. In: *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, pp. 3325–3339. DOI: 10.18653/V1/2022.FINDINGS-ACL.262. URL: <https://doi.org/10.18653/v1/2022.findings-acl.262>.
- Dimitri von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann (2024). “A Language Model’s Guide Through Latent Space”. In: *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. URL: <https://openreview.net/forum?id=c0LoolDFw4>.
- Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia (2022). “ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction”. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz. Association for Computational Linguistics, pp. 3715–3734. DOI: 10.18653/V1/2022.NAACL-MAIN.272. URL: <https://doi.org/10.18653/v1/2022.naacl-main.272>.
- Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen (2018). “CARER: Contextualized Affect Representations for Emotion

- Recognition”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii. Association for Computational Linguistics, pp. 3687–3697. DOI: 10.18653/V1/D18-1404. URL: <https://doi.org/10.18653/v1/d18-1404>.
- Timo Schick, and Hinrich Schütze (2021). “Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*. Ed. by Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty. Association for Computational Linguistics, pp. 255–269. DOI: 10.18653/V1/2021.EACL-MAIN.20. URL: <https://doi.org/10.18653/v1/2021.eacl-main.20>.
- Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling (2018). “Modeling Relational Data with Graph Convolutional Networks”. In: *ESWC*. Vol. 10843. Lecture Notes in Computer Science. Springer, pp. 593–607.
- Giovanni Servedio, Alessandro De Bellis, Dario Di Palma, Vito Walter Anelli, and Tommaso Di Noia (2025). “Are the Hidden States Hiding Something? Testing the Limits of Factuality-Encoding Capabilities in LLMs”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics, pp. 6089–6104. URL: <https://aclanthology.org/2025.acl-long.304/>.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts (2013). “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, pp. 1631–1642. DOI: 10.18653/V1/D13-1170. URL: <https://doi.org/10.18653/v1/d13-1170>.
- Anders Søgaard (2011). “Semi-supervised condensed nearest neighbor for part-of-speech tagging”. In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Confer-*

- ence, 19-24 June, 2011, Portland, Oregon, USA - Short Papers. The Association for Computer Linguistics, pp. 48–52. URL: <https://aclanthology.org/P11-2009/>.
- Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi (2024). “LLM-Check: Investigating Detection of Hallucinations in Large Language Models”. In: *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*. Ed. by Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang. URL: http://papers.nips.cc/paper%5C_files/paper/2024/hash/3c1e1fdf305195cd620c118aaa9717ad-Abstract-Conference.html.
- William Stigall, Md Abdullah Al Hafiz Khan, Dinesh Chowdary Attota, Francis Nweke, and Yong Pei (2024). “Large Language Models Performance Comparison of Emotion and Sentiment Classification”. In: *Proceedings of the 2024 ACM Southeast Conference, ACM SE 2024, Marietta, GA, USA, April 18-20, 2024*. ACM, pp. 60–68. DOI: 10.1145/3603287.3651183. URL: <https://doi.org/10.1145/3603287.3651183>.
- Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu (2023). “One Embedder, Any Task: Instruction-Finetuned Text Embeddings”. In: *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*. Ed. by Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki. Association for Computational Linguistics, pp. 1102–1121. DOI: 10.18653/V1/2023.FINDINGS-ACL.71. URL: <https://doi.org/10.18653/v1/2023.findings-acl.71>.
- Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu (2024). “RoFormer: Enhanced transformer with Rotary Position Embedding”. In: *Neurocomputing* 568, p. 127063. DOI: 10.1016/J.NEUCOM.2023.127063. URL: <https://doi.org/10.1016/j.neucom.2023.127063>.
- Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang (2019). “BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer”. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. Ed. by Wenwu Zhu, Dacheng Tao,

- Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu. ACM, pp. 1441–1450. DOI: 10.1145/3357384.3357895. URL: <https://doi.org/10.1145/3357384.3357895>.
- Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong (2024). “Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs?” In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*. Ed. by Kevin Duh, Helena Gómez-Adorno, and Steven Bethard. Association for Computational Linguistics, pp. 311–325. DOI: 10.18653/V1/2024.NAACL-LONG.18. URL: <https://doi.org/10.18653/v1/2024.naacl-long.18>.
- Mujeen Sung, Jinhyuk Lee, Sean Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang (Nov. 2021). “Can Language Models be Biomedical Knowledge Bases?” In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 4723–4734. DOI: 10.18653/v1/2021.emnlp-main.388. URL: <https://aclanthology.org/2021.emnlp-main.388/>.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick (2019). “What do you learn from context? Probing for sentence structure in contextualized word representations”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: <https://openreview.net/forum?id=SJzSgnRcKX>.
- Komal K. Teru, Etienne G. Denis, and William L. Hamilton (2020). “Inductive Relation Prediction by Subgraph Reasoning”. In: *ICML*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 9448–9457.
- Kristina Toutanova, and Danqi Chen (2015). “Observed versus latent features for knowledge base and text inference”. In: *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, CVSC 2015, Beijing, China, July 26-31, 2015*. Ed. by Alexandre Allauzen, Edward Grefenstette, Karl Moritz Hermann, Hugo Larochelle, and Scott Wen-tau Yih. Association

- for Computational Linguistics, pp. 57–66. DOI: 10.18653/V1/W15-4007. URL: <https://doi.org/10.18653/v1/W15-4007>.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample (2023a). “LLaMA: Open and Efficient Foundation Language Models”. In: *CoRR* abs/2302.13971. DOI: 10.48550/ARXIV.2302.13971. arXiv: 2302.13971. URL: <https://doi.org/10.48550/arXiv.2302.13971>.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom (2023b). “Llama 2: Open Foundation and Fine-Tuned Chat Models”. In: *CoRR* abs/2307.09288. DOI: 10.48550/ARXIV.2307.09288. arXiv: 2307.09288. URL: <https://doi.org/10.48550/arXiv.2307.09288>.
- Shriyash Upadhyay, Etan Ginsberg, and Chris Callison-Burch (2023). “Improving Mathematics Tutoring With A Code Scratchpad”. In: *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2023, Toronto, Canada, 13 July 2023*. Ed. by Ekaterina Kochmar, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Nitin Madnani, Anais Tack, Victoria Yaneva, Zheng Yuan, and Torsten Zesch. Association for Computational Linguistics, pp. 20–28. DOI: 10.18653/V1/2023.BEA-1.2. URL: <https://doi.org/10.18653/v1/2023.bea-1.2>.

- Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar (2020). “Composition-based Multi-Relational Graph Convolutional Networks”. In: *ICLR*. OpenReview.net.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). “Attention is All you Need”. In: *NIPS*, pp. 5998–6008.
- Artem Vazhentsev, Ekaterina Fadeeva, Rui Xing, Alexander Panchenko, Preslav Nakov, Timothy Baldwin, Maxim Panov, and Artem Shelmanov (2024). “Unconditional Truthfulness: Learning Conditional Dependency for Uncertainty Quantification of Large Language Models”. In: *CoRR* abs/2408.10692. DOI: 10.48550/ARXIV.2408.10692. arXiv: 2408.10692. URL: <https://doi.org/10.48550/arXiv.2408.10692>.
- Denny Vrandečić, and Markus Krötzsch (Sept. 2014). “Wikidata: a free collaborative knowledgebase”. In: *Commun. ACM* 57.10, pp. 78–85. ISSN: 0001-0782. DOI: 10.1145/2629489. URL: <https://doi.org/10.1145/2629489>.
- Ivan Vulic, Edoardo Maria Ponti, Robert Litschko, Goran Glavas, and Anna Korhonen (2020). “Probing Pretrained Language Models for Lexical Semantics”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics, pp. 7222–7240. DOI: 10.18653/V1/2020.EMNLP-MAIN.586. URL: <https://doi.org/10.18653/v1/2020.emnlp-main.586>.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi (2020). “Fact or Fiction: Verifying Scientific Claims”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics, pp. 7534–7550. DOI: 10.18653/V1/2020.EMNLP-MAIN.609. URL: <https://doi.org/10.18653/v1/2020.emnlp-main.609>.
- Hetong Wang, Pasquale Minervini, and Edoardo M. Ponti (2024a). “Probing the Emergence of Cross-lingual Alignment during LLM Training”. In: *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Association for Computational Linguistics, pp. 12159–

12173. DOI: 10.18653/V1/2024.FINDINGS-ACL.724. URL: <https://doi.org/10.18653/v1/2024.findings-acl.724>.
- Ling Wang, Jicang Lu, Gang Zhou, Hangyu Pan, Taojie Zhu, Ningbo Huang, Peng He, and Carmen De Maio (Jan. 2022). “Representation Learning Method with Semantic Propagation on Text-Augmented Knowledge Graphs”. In: *Intell. Neuroscience* 2022. ISSN: 1687-5265. DOI: 10.1155/2022/1438047. URL: <https://doi.org/10.1155/2022/1438047>.
- Shuyao Wang, Yongduo Sui, Chao Wang, and Hui Xiong (2024b). “Unleashing the Power of Knowledge Graph for Recommendation via Invariant Learning”. In: *WWW*. ACM, pp. 3745–3755.
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang (2021). “KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation”. In: *Trans. Assoc. Comput. Linguistics* 9, pp. 176–194.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen (2014). “Knowledge Graph Embedding by Translating on Hyperplanes”. In: *AAAI*. AAAI Press, pp. 1112–1119.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus (2022). “Emergent Abilities of Large Language Models”. In: *Trans. Mach. Learn. Res.* 2022. URL: <https://openreview.net/forum?id=yzkSU5zdwD>.
- Jerry W. Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma (2023a). “Larger language models do in-context learning differently”. In: *CoRR* abs/2303.03846. DOI: 10.48550/ARXIV.2303.03846. arXiv: 2303.03846. URL: <https://doi.org/10.48550/arXiv.2303.03846>.
- Yanbin Wei, Qiushi Huang, Yu Zhang, and James T. Kwok (2023b). “KICGPT: Large Language Model with Knowledge in Context for Knowledge Graph Completion”. In: *EMNLP (Findings)*. Association for Computational Linguistics, pp. 8667–8683.
- Jiaxin Wen, Pei Ke, Hao Sun, Zhixin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang (2023). “Unveiling the Implicit Toxicity in Large Language Models”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural*

- Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Association for Computational Linguistics, pp. 1322–1338. DOI: 10.18653/V1/2023.EMNLP-MAIN.84. URL: <https://doi.org/10.18653/v1/2023.emnlp-main.84>.
- Janyce Wiebe (2000). “Learning Subjective Adjectives from Corpora”. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*. Ed. by Henry A. Kautz, and Bruce W. Porter. AAAI Press / The MIT Press, pp. 735–740. URL: <http://www.aaai.org/Library/AAAI/2000/aaai00-113.php>.
- Jiawei Wu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun (2016). “Knowledge Representation via Joint Learning of Sequential Text and Knowledge Graphs”. In: *CoRR* abs/1609.07075. arXiv: 1609.07075. URL: <http://arxiv.org/abs/1609.07075>.
- Weiqi Wu, Chengyue Jiang, Yong Jiang, Pengjun Xie, and Kewei Tu (July 2023). “Do PLMs Know and Understand Ontological Knowledge?” In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 3080–3101. DOI: 10.18653/v1/2023.acl-long.173. URL: <https://aclanthology.org/2023.acl-long.173>.
- Qingrong Xia, Bo Zhang, Rui Wang, Zhenghua Li, Yue Zhang, Fei Huang, Luo Si, and Min Zhang (2021). “A Unified Span-Based Approach for Opinion Mining with Syntactic Constituents”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Association for Computational Linguistics, pp. 1795–1804. DOI: 10.18653/V1/2021.NAACL-MAIN.144. URL: <https://doi.org/10.18653/v1/2021.naacl-main.144>.
- Frank Xing (2025). “Designing Heterogeneous LLM Agents for Financial Sentiment Analysis”. In: *ACM Trans. Manag. Inf. Syst.* 16.1, pp. 1–24. DOI: 10.1145/3688399. URL: <https://doi.org/10.1145/3688399>.

- Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka (2018). “Representation Learning on Graphs with Jumping Knowledge Networks”. In: *ICML*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 5449–5458.
- Yadollah Yaghoobzadeh, Katharina Kann, Timothy J. Hazen, Eneko Agirre, and Hinrich Schütze (2019). “Probing for Semantic Classes: Diagnosing the Meaning Content of Word Embeddings”. In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Ed. by Anna Korhonen, David R. Traum, and Lluís Màrquez. Association for Computational Linguistics, pp. 5740–5753. DOI: 10.18653/V1/P19-1574. URL: <https://doi.org/10.18653/v1/p19-1574>.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan (2024). “Qwen2 Technical Report”. In: *CoRR* abs/2407.10671.
- Fan Yang, Zhilin Yang, and William W. Cohen (2017). “Differentiable Learning of Logical Rules for Knowledge Base Reasoning”. In: *NIPS*, pp. 2319–2328.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le (2019). “XLNet: Generalized Autoregressive Pretraining for Language Understanding”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, pp. 5754–5764. URL: <https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html>.

- Liang Yao, Chengsheng Mao, and Yuan Luo (2019). “KG-BERT: BERT for Knowledge Graph Completion”. In: *CoRR* abs/1909.03193.
- Lin Yao, Chengjie Sun, Lu Li, Zhixin Hao, and Xiaolong Wang (2010). “Hybrid learning of Syntactic and Semantic Dependencies”. In: *Comput. Inf. Sci.* 3.4, pp. 187–199. DOI: 10.5539/CIS.V3N4P187. URL: <https://doi.org/10.5539/cis.v3n4p187>.
- Paul Youssef, Osman Alperen Koras, Meijie Li, Jörg Schlötterer, and Christin Seifert (2023). “Give Me the Facts! A Survey on Factual Knowledge Probing in Pre-trained Language Models”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Association for Computational Linguistics, pp. 15588–15605. DOI: 10.18653/V1/2023.FINDINGS-EMNLP.1043. URL: <https://doi.org/10.18653/v1/2023.findings-emnlp.1043>.
- Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro (2024). “RankRAG: Unifying Context Ranking with Retrieval-Augmented Generation in LLMs”. In: *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*. Ed. by Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang. URL: http://papers.nips.cc/paper%5C_files/paper/2024/hash/db93ccb6cf392f352570dd5af0a223d3-Abstract-Conference.html.
- Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni (2023). “Where to Go Next for Recommender Systems? ID-vs. Modality-based Recommender Models Revisited”. In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*. Ed. by Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete. ACM, pp. 2639–2649. DOI: 10.1145/3539618.3591932. URL: <https://doi.org/10.1145/3539618.3591932>.
- Hanwen Zha, Zhiyu Chen, and Xifeng Yan (2022). “Inductive Relation Prediction by BERT”. In: *AAAI*. AAAI Press, pp. 5923–5931.
- Dun Zhang, and FulongWang (2024). “Jasper and Stella: distillation of SOTA embedding models”. In: *CoRR* abs/2412.19048. DOI: 10.48550/ARXIV.2412.

19048. arXiv: 2412.19048. URL: <https://doi.org/10.48550/arXiv.2412.19048>.
- Li Zhang, Haimeng Fan, Chengxia Peng, Guozheng Rao, and Qing Cong (2020a). “Sentiment Analysis Methods for HPV Vaccines Related Tweets Based on Transfer Learning”. In: *Healthcare* 8.3. ISSN: 2227-9032. DOI: 10.3390/healthcare8030307. URL: <https://www.mdpi.com/2227-9032/8/3/307>.
- Lingzi Zhang, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen (2024a). “Dual-View Whitening on Pre-trained Text Embeddings for Sequential Recommendation”. In: *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*. Ed. by Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan. AAAI Press, pp. 9332–9340. DOI: 10.1609/AAAI.V38I8.28786. URL: <https://doi.org/10.1609/aaai.v38i8.28786>.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer (2022). “OPT: Open Pre-trained Transformer Language Models”. In: *CoRR* abs/2205.01068. DOI: 10.48550/ARXIV.2205.01068. arXiv: 2205.01068. URL: <https://doi.org/10.48550/arXiv.2205.01068>.
- Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing (2024b). “Sentiment Analysis in the Era of Large Language Models: A Reality Check”. In: *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*. Ed. by Kevin Duh, Helena Gómez-Adorno, and Steven Bethard. Association for Computational Linguistics, pp. 3881–3906. DOI: 10.18653/V1/2024.FINDINGS-NAACL.246. URL: <https://doi.org/10.18653/v1/2024.findings-naacl.246>.
- Xiaokang Zhang, Zijun Yao, Jing Zhang, Kaifeng Yun, Jifan Yu, Juanzi Li, and Jie Tang (2024c). “Transferable and Efficient Non-Factual Content Detection via Probe Training with Offline Consistency Checking”. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Association for Computational

Linguistics, pp. 12348–12364. DOI: 10.18653/V1/2024.ACL-LONG.668. URL: <https://doi.org/10.18653/v1/2024.acl-long.668>.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang (2024d). “mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval”. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track, Miami, Florida, USA, November 12-16, 2024*. Ed. by Franck Dernoncourt, Daniel Preotiuc-Pietro, and Anastasia Shimorina. Association for Computational Linguistics, pp. 1393–1412. DOI: 10.18653/V1/2024.EMNLP-INDUSTRY.103. URL: <https://doi.org/10.18653/v1/2024.emnlp-industry.103>.

Zhiyuan Zhang, Xiaoqian Liu, Yi Zhang, Qi Su, Xu Sun, and Bin He (2020b). “Pretrain-KGE: Learning Knowledge Representation from Pretrained Language Models”. In: *EMNLP (Findings)*. Vol. EMNLP 2020. Findings of ACL. Association for Computational Linguistics, pp. 259–266.

Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, Yushuo Chen, Lanling Xu, Gaowei Zhang, Zhen Tian, Changxin Tian, Shanlei Mu, Xinyan Fan, Xu Chen, and Ji-Rong Wen (2022). “RecBole 2.0: Towards a More Up-to-Date Recommendation Library”. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*. Ed. by Mohammad Al Hasan, and Li Xiong. ACM, pp. 4722–4726. DOI: 10.1145/3511808.3557680. URL: <https://doi.org/10.1145/3511808.3557680>.

Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen (2021). “RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms”. In: *CIKM ’21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. Ed. by Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong. ACM, pp. 4653–4664. DOI: 10.1145/3459637.3482016. URL: <https://doi.org/10.1145/3459637.3482016>.

- Kaiyuan Zheng, Qinghua Zhao, and Lei Li (2025). “Reassessing the Role of Chain-of-Thought in Sentiment Analysis: Insights and Limitations”. In: *Advanced Intelligent Computing Technology and Applications - 21st International Conference, ICIC 2025, Ningbo, China, July 26-29, 2025, Proceedings, Part XXIV*. Ed. by De-Shuang Huang, Qinhu Zhang, Chuanlei Zhang, and Wei Chen. Vol. 15865. Lecture Notes in Computer Science. Springer, pp. 89–100. DOI: 10.1007/978-981-95-0020-8_8. URL: https://doi.org/10.1007/978-981-95-0020-8%5C_8.
- Wentao Zhou, Jun Zhao, Tao Gui, Qi Zhang, and Xuanjing Huang (2023). “Inductive Relation Inference of Knowledge Graph Enhanced by Ontology Information”. In: *EMNLP (Findings)*. Association for Computational Linguistics, pp. 6491–6502.
- Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang (2024). “LLMs for knowledge graph construction and reasoning: recent capabilities and future opportunities”. In: *World Wide Web (WWW)* 27.5, p. 58.
- Andy Zou, Long Phan, Sarah Li Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks (2023). “Representation Engineering: A Top-Down Approach to AI Transparency”. In: *CoRR* abs/2310.01405. DOI: 10.48550/ARXIV.2310.01405. arXiv: 2310.01405. URL: <https://doi.org/10.48550/arXiv.2310.01405>.