UNIVERSITY OF TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
DEPARTMENT OF INDUSTRIAL ENGINEERING
FONDAZIONE BRUNO KESSLER

**Doctorate Program in Industrial Innovation**

# From Text to Neural Embeddings: Structuring Conceptual Knowledge in Large Language Models

Marco Bronzini

Advisor
Prof. Andrea Passerini
Università di Trento

Co-Advisor
Dott. Carlo Nicolini
Ipazia S.p.A.

XXXVIII | October 2026

Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani | UNIVERSITÀ DI TRENTO

# Acknowledgment

To myself, who has never given up on me while crashing into unexpected obstacles.

Instead of viewing these moments as overwhelming challenges, I have learned to perceive them as essential milestones along a longer and unknown journey: a shift that has helped me appreciate the serendipity of life. A PhD has the potential to accelerate personal growth dramatically, but my own growth would not have been possible without my questioning mindset and the constructive feedback of all my de-facto supervisors. Thus, I also want to express my deep gratitude to Andrea Passerini, Carlo Nicolini, Bruno Lepri, and Jacopo Staiano; each of you has played a meaningful role in making me a stronger person.

*Un ringraziamento immenso ad Ornella, mia madre.*

*Se sono arrivato fin qui, è anche merito tuo. Mi hai reso ciò che sono. Ora mi sento fiero di me stesso, ed è anche grazie a te. Mi hai sempre sostenuto in tutto, nei momenti difficili cosi come in quelli importanti. Ma, soprattutto, con il tuo esempio mi hai mostrato la forza di superare ogni difficoltà.*

*Per fare una famiglia bastano due persone, e noi siamo una famiglia.*

*"E quindi uscimmo a riveder le stelle"*

Dante Alighieri. Inferno (Canto XXXIV), Divina Commedia.

# Abstract

The broad adoption of Large Language Models (LLMs) has transformed the landscape of Natural Language Processing (NLP) since the early 2020s, placing this PhD thesis within one of the fastest-growing areas of recent years. This research project first investigates the capabilities of language models for the NLP task of Information Extraction via prompting, and subsequently deepens our understanding of their internal mechanisms through neural probing. The first part of this PhD thesis advances the extraction of structured information from verbose textual documents through prompt engineering. It leverages different prompting techniques to extend the traditional, syntax-driven task of Information Extraction. In particular, the semantic understanding and prompting flexibility of LLMs are exploited to extract triples centred on Environmental, Social, and Governance aspects from companies' non-financial reports. The resulting structured information is then represented as graphs, enabling descriptive, similarity-based, correlation, and interpretability analyses that uncover non-trivial insights into companies' disclosures. Subsequently, the research questions of this PhD thesis move beyond the use of LLMs as black-box tools and instead focus on the limited understanding of their internal representations. Specifically, the first probing-oriented work introduces a novel methodology for investigating the knowledge-resolution process in LLMs by extracting factual structures from their latent space and visualizing their layer-wise dynamics within graph representations. Conversely, the final part of this thesis investigates the semantics encoded in the vector space of LLMs from a different perspective by addressing research questions related to the output-oriented interpretability of the semantic concepts represented within LLMs. It introduces a new probing paradigm, combining ideas from neural probing and symbolic representation, to generalise the extraction of structured knowledge from neural embeddings. By exploiting the properties of Vector Symbolic Architectures and hypervector algebra, this probing method provides non-trivial insights into the latent space of LLMs, ranging from concept-based patterns across input types to the investigation of representational changes before and after text generation. Following the research developments in the community, the novel methodologies proposed in this PhD thesis advance both the application of LLMs to traditional tasks and the semantic interpretability of their latent representations.

## Keywords

*Interpretability, Large Language Models, Semantics, Information Extraction*

# Contents


**1 Introduction** **1**
- 1.1 Problems . . . 1
- 1.2 Solutions . . . 4
- 1.3 Structure of the Thesis . . . 7
- 1.4 List of Publications . . . 7

**2 Background** **9**
- 2.1 Non-financial Reporting . . . 9
- 2.2 Interpretability of Neural Embeddings . . . 10
  - 2.2.1 Activation Patching . . . 12
  - 2.2.2 Vector Symbolic Architectures . . . 14

**3 Proposed Approaches** **17**
- 3.1 Deriving Structured Insights from Textual Documents . . . 17
  - 3.1.1 Problem Statement . . . 17
  - 3.1.2 Data Sources . . . 19
  - 3.1.3 Data Preparation . . . 22
  - 3.1.4 Retrieval-Augmented Triple Generation . . . 25
  - 3.1.5 Quality of Generated Triples . . . 28
  - 3.1.6 Knowledge Graph Generation . . . 31
  - 3.1.7 Approaches for Statistical Analyses . . . 35
- 3.2 Tracing the Dynamics of Factual Knowledge in LLMs . . . 41
  - 3.2.1 Problem Statement . . . 41
  - 3.2.2 Prompt Definition . . . 42
  - 3.2.3 Activation Patching . . . 43
  - 3.2.4 Knowledge Graph Generation . . . 46
- 3.3 Decoding Neural Embeddings with VSAs . . . 47
  - 3.3.1 Problem Statement . . . 47
  - 3.3.2 Synthetic Corpus . . . 48
  - 3.3.3 Training Examples . . . 50
  - 3.3.4 Processing Neural Embeddings . . . 51
  - 3.3.5 Neural VSA Encoder . . . 54
  - 3.3.6 Probing VSA Encodings . . . 56

**4 Experimental Results** **59**
4.1 Deriving Structured Insights from Companies' Non-financial Reports . 59
4.1.1 Network Statistics of Bipartite Graphs . . . . . 60
4.1.2 Diversity Analysis of ESG Category Disclosure . . . . . 61
4.1.3 Company Similarities based on Disclosed Actions . . . . . 64
4.1.4 Correlation Analysis among Company Similarities . . . . . 65
4.1.5 Interpretability of ESG Scores . . . . . 67
4.1.6 Practical and Methodological Implications of the Findings . . . 71
4.2 Tracing the Dynamics of Factual Knowledge in LLMs . . . . . 75
4.2.1 Experimental Setup . . . . . 75
4.2.2 LLM Performance on Claim Verification . . . . . 76
4.2.3 Local Interpretability . . . . . 77
4.2.4 Global Interpretability . . . . . 79
4.3 Extracting Semantic Concepts with VSAs . . . . . 82
4.3.1 Experimental Setup . . . . . 82
4.3.2 Concept Extraction . . . . . 83
4.3.3 Comparison with Logit Attribution . . . . . 88
4.3.4 Comparison with Sparse Autoencoders . . . . . 89
4.3.5 From Input-Completion Tasks to Text Generation . . . . . 92

**5 Related Work** **97**
5.1 Text Analytics on Non-financial Disclosures . . . . . 97
5.2 Knowledge Graphs Generation . . . . . 98
5.3 Probing Neural Embeddings . . . . . 99
5.3.1 Logit Attribution . . . . . 100
5.3.2 Activation Patching . . . . . 101
5.3.3 Sparse Autoencoders . . . . . 102
5.3.4 Supervised Probes . . . . . 103

**6 Conclusions** **105**

**Bibliography** **109**

**A Deriving Structured Insights from Companies' Reports** **127**
A.1 ESG Categories . . . . . 127
A.2 Hyperparameter in LLM's Text Generation . . . . . 128
A.3 Model's Instruction for Triple Generation . . . . . 129
A.4 Model's Instruction for Triple Evaluation . . . . . 130
A.5 Experiments with other Language Models . . . . . 131
A.6 Comparison between Company's Disclosures and Sector Materiality . . 134
A.7 Features for the correlation analysis . . . . . 135
A.8 Statistical significance of the pairwise correlations . . . . . 136
A.9 Region-based interpretability analysis . . . . . 137
A.9.1 ESG scores by region . . . . . 137
A.9.2 Top features impacting the Asian companies' ESG scores . . . . 138

**B Tracing the Dynamics of Factual Knowledge in LLMs** **139**
B.1 Source Prompt . . . 139
B.2 Target Prompt . . . 140
B.3 A Dynamic Knowledge Graph for an Input . . . 141
B.4 Confusion Matrix for LLM's Claim Veritication . . . 142
B.5 Layer-wise Evolution of the Factual Knowledge . . . 143
B.6 Comparison across Different Token sets . . . 145
B.7 Output Comparison among Different Models . . . 147
B.8 Attentions on Prompt Tokens across Layers . . . 149

**C Extracting Semantic Concepts with VSAs** **151**
C.1 Synthetic corpus . . . 151
C.2 Architecture of *hyperdimensional probe* . . . 152
C.3 Unbinding stage from Section 3.3.6 . . . 153
C.4 Experimental comparison . . . 153
C.4.1 DLA-based experimental results . . . 154
C.4.2 Raw results obtained though the DLA probing technique . . . 156
C.5 Applicability to other domains . . . 157
C.5.1 Generalization of input representation . . . 157
C.5.2 Applicability to other downstream tasks . . . 157
C.5.3 Handling of Out-of-Domain Data . . . 158
C.6 Cosine similarities among the VSA codebook items . . . 159
C.7 Question-answering setting . . . 160
C.8 Dimensionality reduction . . . 161
C.8.1 Average correlations among model's hidden layer . . . 161
C.8.2 Analysis of representation redundancy . . . 161
C.8.3 Ablation study on the dimensionality-reduction steps . . . 162
C.9 Computational workload . . . 163

# List of Tables


3.1 Overview of the companies' sustainability reports . . . . . 20
3.2 Semantic search for ESG-related sentences . . . . . 24
3.3 Sample of evaluation scores of five generated triples . . . . . 29
3.4 Comparison of triples generated with/without ICL . . . . . 30
3.5 Comparison of triples generated with/without RAG . . . . . 31
3.6 Sample of semantic clusters for the generated triples . . . . . 32
3.7 The bipartite graphs for examining insights from companies' reports. . 35
3.8 Descriptive statistics of the categorical predictors for ESG scores . . . . 38
3.9 Knowledge bases adopted to build the synthetic corpus $\mathcal{S}$. . . . . 49
3.10 Analogy-style corpus grouped by domain . . . . . 49
3.11 Training performance of neural VSA encoders . . . . . 56

4.1 Descriptive Statistics of the bipartite graphs . . . . . 60
4.2 Graph metrics of category nodes of the company-category graph . . . . 61
4.3 ESG categories with their entropy values . . . . . 62
4.4 ESG categories aggregated by companies' sectors . . . . . 63
4.5 Sample of actions reported by companies . . . . . 66
4.6 Asian companies for the region-based interpretability analysis . . . . . 71
4.7 Performance of the language model on claim verification . . . . . 76
4.8 Concepts extracted by hyperdimensional probe . . . . . 84
4.9 Probing failures grouped by input type . . . . . 86
4.10 Performance of the VSA-based method and LLM performance . . . . . 88
4.11 Concepts extracted though VSA probing when DLA fails . . . . . 89
4.12 Training performance of SAEs on the test set . . . . . 90
4.13 Concepts extracted by Sparse Autoencoders . . . . . 91

A.1 ESG categories of the ESG categorization . . . . . 127
A.2 Comparison of outputs from different LLMs . . . . . 132
A.3 Data-driven findings and sector materiality . . . . . 134
A.4 Similarity measures for company's features . . . . . 135

B.1 Evolution of the output generated by activation patching . . . . . 143
B.2 Ablation of considered token sets . . . . . 145
B.3 Activation patching with different language models . . . . . 147

C.1 Synthetic data including analogy-style inputs . . . . . 151
C.2 Architecture of the neural VSA encoder . . . . . 152
C.3 Statistics for the unbinding stage . . . . . 153
C.4 Concepts extracted using the DLA probing technique . . . . . 154
C.5 Extracted factors for the subset of DLA failures . . . . . 155
C.6 Concepts extracted by DLA when VSA yields no concepts . . . . . 155
C.7 Eigenvalues of the Gram matrix from OLMo-2's residual stream. . . . . 162

# List of Figures


2.1 Decoding semantics from neural embeddings. . . . . 11
2.2 Random vectors are associated to semantic concepts . . . . . 15

3.1 Concept of triple generation with ESG-aligned content . . . . . 18
3.2 Deriving structured insights from textual documents . . . . . 23
3.3 In-context examples for retrieval-augmented triple generation . . . . . 26
3.4 Model instruction for extracting structure information from text . . . . 27
3.5 Evaluation of ESG-oriented triples using a LLM-as-a-judge . . . . . 28
3.6 Knowledge Graph generated via LLM-driven triple generation . . . . . 34
3.7 Descriptive statistics of the numerical predictors for ESG scores . . . . 39
3.8 Framework for decoding factual knowledge from neural encodings . . . 42
3.9 Inference of $\mathcal{M}$ on the source prompt $\mathcal{S}$. . . . . 43
3.10 Patching operation during the inference of $\mathcal{M}$ on the target prompt . . 44
3.11 Binary weights for the input's tokens . . . . . 45
3.12 Graph generation process from the ground literals . . . . . 46
3.13 Framework for probing neural embeddings using VSAs . . . . . 48
3.14 Silhouette scores for varying numbers of clusters . . . . . 52
3.15 K-means clustering on model's hidden layers . . . . . 53
3.16 Regression model to map neural embeddings into VSA . . . . . 55

4.1 Network linking companies that report similar actions . . . . . 65
4.2 Pairwise correlations between companies' similarities . . . . . 67
4.3 Features impacting the most the inference of ESG score . . . . . 68
4.4 Companies' ESG scores grouped by the decade of incorporation . . . . 69
4.5 Explanations of company's ESG score using SHAP values . . . . . 70
4.6 Dynamics of the Factual knowledge from the latent representations . . 78
4.7 Cosine similarities between the graph representations . . . . . 79
4.8 Experimental setup of VSA-based probing . . . . . 82
4.9 Performance of the VSA-based method and LLM performance . . . . . 83
4.10 Experimental metrics on probing the vector space of Llama 4 . . . . . 85
4.11 Concepts extracted before and after LLM's text generation . . . . . 94

A.1 Model instruction for information extraction . . . . . 129
A.2 The model's instruction for the LLM-as-a-judge evaluation. . . . . 130
A.3 Distribution of p-values for Kendall's correlation . . . . . 136

A.4 Companies' ESG scores aggregated by continent . . . . . 137
A.5 Factors with the influence the ESG score for Asian companies. . . . . 138

B.1 Template for the source prompt . . . . . 139
B.2 Template for the target prompt . . . . . 140
B.3 A comprehensive dynamic knowledge graph for an input . . . . . 141
B.4 Confusion matrices for LLM's claim verification . . . . . 142
B.5 Attention matrices across layers . . . . . 149

C.1 Raw outputs obtained using DLA . . . . . 156
C.2 Cosine similarities among the codebook items . . . . . 159
C.3 Correlations in model's hidden layers . . . . . 161

# List of Abbreviations

**AUC** Receiver Operating Characteristic Area Under the Curve.

**CSR** Corporate Social Responsibility.

**DLA** Direct Logit Attribution.

**EBITDA** Earnings Before Interest, Taxes, Depreciation, and Amortization.

**ELK** Eliciting Latent Knowledge.

**ESG** Environmental, Social, and Governance.

**ICL** In-Context Learning.

**IE** Information Extraction.

**KGs** Knowledge Graphs.

**LLMs** Large Language Models.

**MAP-B** Multiply-Add-Permute Bipolar.

**MLP** Multi-Layer Perceptron.

**NLP** Natural Language Processing.

**OIE** Open Information Extraction.

**QA** Question Answering.

**RAG** Retrieval-Augmented Generation.

**SAEs** Sparse Autoencoders.

**SPO** Subject-Predicate-Object.

**SQuAD** Stanford Question Answering Dataset.

**VSAs** Vector Symbolic Architectures.

# Glossary

**Activation Patching** A vector-based probing technique to alter the model's latent representations during inference.

**Bipartite Graph** A graph whose vertices can be divided into two distinct and independent sets or partitions.

**Corpus** A textual dataset with a collection of written texts.

**ESG Ratings** Numerical scores measuring companies' ESG performance.

**Knowledge Graph** A structured way to represent real-world entities and their relationships as a network of nodes and edges.

**Latent Representations** High-dimensional vector space, also known as residual stream, that aggregates the neural embeddings of all hidden layers of transformer models.

**Neural Embeddings** Vector representations in $\mathbb{R}^n$ capturing learned meaning.

**Non-Financial Reporting** Disclosing the impact of Company Social Responsibility.

**Probing** Analysing neural embeddings to uncover semantic meaning.

**Prompt** A natural-language instruction given to a LLM to guide its response.

**Prompt Engineering** Find an effective natural-language instructions to guide a language model in solving an NLP task.

**Prompting** Querying a language model with natural-language instructions.

# Chapter 1

# Introduction

November 2022 marked both the start of my research project and the release of OpenAI's ChatGPT. This timing placed me, exhaustingly but uniquely, at the center of one of the most rapidly evolving areas of recent years; influencing both the broader public and the research community in Natural Language Processing (NLP). I have witnessed significant transformations in this community over the years, from evaluating the impact of Large Language Models (LLMs) on traditional NLP tasks to investigating their internal reasoning and guiding their generative behaviour. Consequently, my research has centered on two major ongoing challenges: exploring the capabilities of these language models in NLP tasks through prompting, and advancing our understanding of their underlying mechanisms through neural probing. Meanwhile, researchers are now progressing from passive investigation toward intentional intervention in LLM's generative behaviour by steering text generation.

## 1.1 Problems

Unlike conventional supervised learning, which trains a machine learning model to map inputs to outputs explicitly, LLMs adopts a two-stage learning paradigm. Initially, foundation language models acquire robust and general-purpose linguistic representations by training on large, raw corpora in a fully unsupervised manner by estimating the probability of observed text. Subsequently, to perform a specific task, a carefully designed task-oriented instruction (prompt) directs the language model to generate the desired output. This prompt-based learning enables LLMs to adapt to new downstream tasks with minimal or no labelled data, performing few-shot or even zero-shot learning by leveraging the linguistic knowledge acquired during unsupervised pre-training [101].

**NLP tasks with LLMs.** However, designing effective model instructions, a task known as prompt engineering, has emerged as a major challenge in successfully applying LLMs to conventional Natural Language Processing tasks. Researchers have proposed a variety of prompting techniques to adapt these powerful language models to traditional NLP tasks such as sentiment analysis, Question Answering, and Information Extraction.

Accordingly, the first part of this PhD thesis (Section 3.1, 20) explores the capabilities of LLMs to tailor the NLP task of Information Extraction (IE) to a specific domain via prompt engineering, aiming at extracting domain-driven insights from textual documents rather based on conventional syntax. This first work addresses the following methodological and experimental research questions:

1. How can the *generative capabilities* of Large Language Models enhance the NLP task of Information Extraction?

2. To what extent can *prompt engineering* steer Information Extraction towards domain-relevant meaningful insights rather than purely syntactic features?

3. Which *prompting techniques* are most effective for tailoring Information Extraction to a specific domain?

4. To what extent can LLMs extract non-trivial, data-driven insights from *companies' non-financial reporting*?

**Black-box nature of LLMs.** Although several studies have already demonstrated that LLMs excel at a wide range of NLP tasks, exploiting vast amounts of learned common-sense and factual knowledge [23, 85, 108, 129, 146], their black-box nature limits our understanding of their internal mechanisms and representations.

The latent representations of transformer-based models[1] is a high-dimensional linear vector space that aggregates the outputs of all hidden layers [51]. The investigation of these vector representations is grounded in the linear representation hypothesis; which asserts that latent features can be represented as linear subspaces [52], formed and accessed throughout the model's inference process [128]. Despite their structured nature, these representations remain challenging to interpret due to their high dimensionality and the abstract characteristics of the features they encode. These features are frequently obscured by the superposition phenomenon, where individual neurons exhibit context-dependent behavior [50]. Consequently, research has focused on developing probing techniques aimed at isolating those subspaces to extract human-interpretable

[1]Often referred to as *vector space*, *latent space* or *residual stream*.

features from neural representations within the vector space of LLMs [128], revealing its embedded semantics. Such probing efforts typically fall into two perspectives [64], serving different yet complementary objectives:

- **Feature extraction** targets input features within the latent space of LLMs;
- **Output distribution inspection** studies the misalignment between the model's internal knowledge and its textual outputs, a task also known as Eliciting Latent Knowledge (ELK).

**Feature extraction for output-oriented analysis.** However, comprehensive interpretability of LLM vector spaces would require combining both, extracting high-level input features that may underlie such misalignment. Unfortunately, existing neural-embedding probing methods restrict this two-sided investigation, due to their constraints on how latent features are defined. As a result, conventional interpretability methods provide only limited access to the broader semantic structure within LLMs, restricting the analysis to task-specific signals [70, 105], the model's output vocabulary [14, 119], or individual entities [64].

Consequently, the second and third parts of this PhD thesis address the gap between these two probing perspectives by extracting input-related semantics from the latent representations of LLMs to enable output-oriented analysis, advancing the interpretability of their internal mechanisms. In particular, the second work (Section 3.2, 21) investigates research questions concerning the internal representation of input's factual knowledge in LLMs and its potential influence on erroneous model responses:

5. Which types of factual knowledge are represented by LLMs when assessing the truthfulness of input statements?
6. How is factual knowledge represented during model's inference, and how does it evolve across the hidden layers of LLMs?
7. To what extent do consistent patterns emerge in the layer-wise evolution of factual knowledge within LLMs?

In contrast, the third work (Section 3.3, 22) answers research questions related to the output-oriented impact of the atomic semantic concepts encoded in the vector space of LLMs:

8. Which semantic concepts can be identified in the internal representations of LLMs during input-completion tasks, and how consistently can they be detected?

9. To what extent do concept-oriented patterns emerge consistently across different input types and across different LLMs?

10. To what extent do question-related and answer-related semantic concepts emerge before and after text generation, and how do their distributions differ?

## 1.2 Solutions

The first research work applies LLMs to extract structured information, aligned to a specific domain, from textual documents (Section 3.1; 20), employing different prompting techniques to extend the conventional and syntax-driven task of Information Extraction.

**Prompt engineering.** In the evolving landscape of prompt engineering, In-Context Learning (ICL, 23) and Retrieval-Augmented Generation (RAG, 62) have emerged as two prominent techniques for enhancing the performance and adaptability of generative language models. In-Context Learning allows language models to perform tasks by conditioning on a small number of examples or task-specific instructions, without any updates in their internal weights. This method empowers models to dynamically adapt to new tasks or domains through cleverly constructed input prompts, leveraging their pre-trained knowledge to produce relevant outputs. The approach presented in Section 3.1 adopts ICL to provide the language model with *domain-aligned examples of input-output pairs*, guiding the model towards syntax-free and domain-aligned information extraction. ICL also helps the model compose its responses according to a specific output format, enabling yielding unified structured information.

On the other hand, Retrieval-Augmented Generation enhances model outputs by integrating external knowledge sources during inference. Instead of relying solely on what the language model has seen during pre-training, RAG architectures retrieve relevant documents or data from an external knowledge base in response to a prompt. Retrieval with generative abilities enables more accurate, factually grounded, and up-to-date responses, especially in knowledge-intensive tasks. The approach presented in

Section 3.1 leverages RAG to guide Information Extraction toward a specific domain by *incorporating domain-relevant knowledge into the model instruction.* In particular, the RAG paradigm improves the alignment of the extraction process with Environmental, Social, and Governance information contained in textual inputs by supplying the language model with a set of relevant aspects and domain-centred descriptions of the targeted data structure.

Together, ICL and RAG enable the extraction of structured, domain-aligned information from complex textual documents (PDFs), facilitating the discovery of non-trivial insights into company's non-financial reporting (Section 2.1) through a unified syntax-free data structure. The extracted domain knowledge is subsequently represented as connected graphs and analyzed through descriptive, similarity-based, correlation, and interpretability techniques (Section 4.1). In addressing the research questions of this work, the proposed approach offers both methodological and empirical contributions: it employs LLMs for structured, domain-aligned information extraction (RQ1–RQ3), while also enabling the extraction of meaningful insights from companies' non-financial reporting practices (RQ4).

**Advancing interpretability of LLMs.** Subsequently, the research questions of this PhD thesis move beyond the use of LLMs as black-box tools and instead focus on the limited understanding of their internal mechanisms and representations. Although the subject of the study shifts from textual documents to neural embeddings, the underlying task remains the extraction of structured information. This transition reframes the investigation from conventional NLP tasks (i.e., IE) to neural probing (Section 2.2).

Specifically, the first probing-oriented work (Section 3.2, 21) introduces a novel methodology for investigating the knowledge-resolution process in LLMs by extracting factual structures from the model's latent space and visualizing their layer-wise dynamics as a dynamic Knowledge Graph. Inspired by the Patchscope framework proposed by Ghandeharioun *et al.* [64], the technique of activation patching (section 2.2.1) is used to decode the factual knowledge embedded within the LLM vector space. Previous studies have examined the factual knowledge of LLMs at the level of individual entities [64], typically within triplet-completion settings [74, 85, 104]. This work builds on this research direction by focusing on factual knowledge at the level of full sentences, moving beyond prior emphasis on individual entities in knowledge-completion tasks to examine the complete knowledge resolution process. It also shifts the focus from the output-centered analyses typical of earlier studies (i.e., ELK) toward joint input–output feature extraction, ultimately providing a more comprehensive understanding of the internal

mechanisms of LLMs. The experimental results (Section 4.2) reveals novel insights into the factual knowledge resolution process of LLMs both locally and globally:

- local interpretability uncovers different factual information (RQ5-RQ6), from subject's information and multi-hop reasoning to representation errors causing erroneous evaluation of the input claim;
- global interpretability reveals patterns in this latent process (RQ7), combining established LLM phenomena (e.g., early layers focusing on syntax) with new findings, such as word-based knowledge evolving into claim-related facts;

Conversely, the third research work (Section 3.3; 22) answers research questions concerning the output-driven interpretability of the atomic semantic concepts encoded in the latent representations of LLMs. The proposed approach advances the semantic-oriented investigation of neural embeddings of LLMs by introducing *Hyperdimensional Probe*, a novel probing paradigm that combines ideas from symbolic representations and neural probing.

Inspired by the sparsity constraints of Sparse Autoencoders (SAEs), the proposed approach addresses key limitations of prior probing methods by exploiting the properties of Vector Symbolic Architectures (VSAs) and hypervector algebra (Section 2.2.2). Functioning as a hybrid supervised probe, this methodology integrates the complementary viewpoints of prior approaches for probing:

- output-oriented investigation of logit-based methods (Section 5.3.1);
- queryable and sparsity-driven proxy space of SAEs (Section 5.3.3);
- top-down interpretability of supervised probes (Section 5.3.4).

It also removes the single-token representation constraint of conventional output-oriented inspection, enabling targeting latent feature with unrestricted abstraction and symbol sources. The novel algorithm proposed for processing neural embeddings also bypasses layer selection of conventional layer-wise probing approaches.

VSA probing is especially well suited to experimental settings that require a constrained or structured conceptual feature space, addressing a key limitation of SAE-based analysis; while also producing findings that are both deeper and more consistent than those obtained through conventional token-oriented approaches. The experimental results (Section 4.3) demonstrate that VSA probing can successfully extract semantic concepts from the latent representations of diverse LLMs (RQ8), despite differences in

embedding dimensionality and across two distinct experimental settings. These findings ultimately highlight varying degrees of conceptual richness across language models and types of input (RQ9). In addition, the study examines concept-aligned dynamics throughout the generation process (RQ10), revealing the representations of question-related and answer-related semantic concepts both before and after text generation.

## 1.3 Structure of the Thesis

The rest of the work is organized as follows. Chapter 2 presents the foundational concepts underpinning this research project, including non-financial reporting (Section 2.1), and interpretability of neural embeddings (Section 2.2). Chapter 3 then introduces the novel approaches for extracting structured insights from textual documents (Section 3.1), unveiling the dynamics of factual knowledge in LLMs (Section 3.2), and decoding semantics from neural embeddings with Vector Symbolic Architectures (3.3). Chapter 4 then reports their corresponding experimental findings. This is followed by the related work chapter (5), which reviews prior research on: text analytics on companies' non-financial disclosures (Section 5.1), knowledge graph generation (Section 5.2), and probing neural embeddings (Section 5.3). Lastly, Chapter 6 offers concluding remarks on this PhD thesis. In addition, Chapters A, B, and C provide supplementary materials for the proposed methodologies, respectively.

## 1.4 List of Publications

The following shows the list of articles published from this PhD dissertation:

1. Marco Bronzini, Carlo Nicolini, Bruno Lepri, Andrea Passerini, and Jacopo Staiano. *Glitter or Gold? Deriving Structured Insights from Sustainability Reports via Large Language Models.* EPJ Data Science, 13(1):41, 2024 [20]

    - *Extracting structured knowledge* from textual documents using LLMs with cutting-edge prompting techniques, such as ICL and RAG (Section 3.1).

2. Marco Bronzini, Carlo Nicolini, Bruno Lepri, Jacopo Staiano, and Andrea Passerini. *Unveiling LLMs: The Evolution of Latent Representations in a Dynamic Knowledge Graph.* In First Conference on Language Modeling, 2024 [21]

- *Advancing the understanding of the internal mechanism* of LLMs by enhancing the interpretability of their knowledge resolution process (Section 3.2).

3. Marco Bronzini, Carlo Nicolini, Bruno Lepri, Jacopo Staiano, and Andrea Passerini. *Hyperdimensional Probe: Decoding LLM Representations via Vector Symbolic Architectures*, 2025. arXiv preprint [22]

   - *Generalizing the extraction of structured knowledge* from neural embeddings of LLMs by designing a novel paradigm for neural probing (Section 3.3).

# Chapter 2

# Background

## 2.1 Non-financial Reporting

Companies' non-financial disclosures are influenced by various exogenous factors, including regional regulations [26] and the political, labor, and cultural context of the countries where companies operate [11, 46, 163]. Pushed by local demands, companies in Europe, America, and Asia could prioritize different initiatives, investments, and disclosures concerning their Corporate Social Responsibility (CSR). For example, Baldini *et al.* [11] show that higher labor union density increases company's disclosures regarding social and governance aspects, while Yu *et al.* [163] report that limited political rights reduce overall non-financial reporting. Regulatory agencies also shape CSR [26]; European firms, for instance, often focus on environmental issues due to stringent climate policies such as the EU Emissions Trading System [55] and the "Fit for 55" plan [56]. This European emphasis may create biases in ESG benchmarking. LaBella *et al.* [95] found rating agencies favor European firms over North American, Emerging Market, or Asian counterparts, partly due to differing reporting standards. Other studies [46, 95, 43] suggest that the size of the company also affects the quality of the disclosure, as larger companies can invest more resources in transparency, indirectly improving their evaluations in the Environmental, Social, and Governance (ESG) aspects. ESG is a framework used by investors and stakeholders to evaluate a company' sustainability, ethical impact, and long-term risk management beyond its financial performance.

**Materiality.** Companies' ESG performance is assessed in comparison to their industry peers, reflecting industry-specific concerns, named as ESG Industry Material-

ity [24, 48, 91]. For example, *packaging* matters more for consumer staples, while *employee safety* is critical in materials industries. Materiality generally identifies two types of sustainability issues in an industry: *financial materiality* relates to ESG issues affecting financial performance, while *impact materiality* covers environmental and social effects caused by a company's activities. Reporting standard bodies, such as the Sustainability Accounting Standards Board (SASB), offer industry-specific materiality matrices to guide disclosures; for example, *water management* in the industry of non-alcoholic beverages.

Therefore, deriving structured insights into companies' non-financial reports can provide stakeholders with deeper and data-driven examination of the progression and diversity of actions taken by companies, as well as correlations across sectors and geographic regions.

**ESG rating scores.** Rating agencies such as Refinitiv, MSCI, and Sustainalytics use companies' non-financial reports and ESG-related information to assess the impact of companies' activities on the environment and society systematically. This assessment, typically done through numerical scores,[1] offers stakeholders valuable measures to evaluate and compare companies in terms of their performance related to ESG aspects.

For example, Refinitiv assigns a company's ESG score as percentage scores [134], wherein lower values (0-25) indicate poor ESG performance and a lack of transparency in publicly reporting material on ESG data (i.e., laggard companies); conversely, higher values (75-100) indicate excellent ESG performance and a high level of transparency in publicly reporting material on ESG data (i.e., leader companies). In addition, it is worth mentioning that a zero score is assigned in the rare case that a company does not disclose any metrics or information relevant to its industry [49, 143]. The combined ESG score is a cumulative measure of E/S/G pillars' weights, which differ by industry for the first two pillars (E and S), while the Iight of the third pillar (G) remains consistent across all industries [134].

## 2.2 Interpretability of Neural Embeddings

Interpretability methods for neural embeddings aim to extract human-interpretable features from the vector spaces of machine learning models (Figure 2.1). However, each probing framework typically target the model's internal representations with respect to

[1] The terms *score* and *rating* are often used interchangeably.

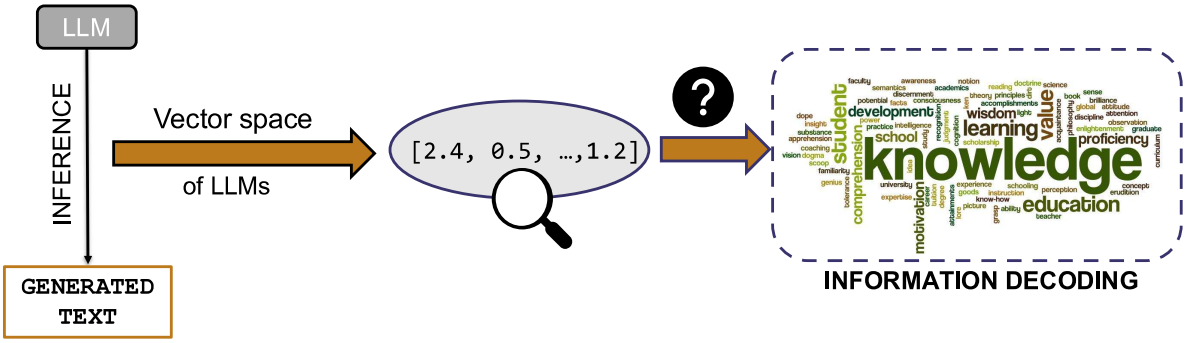



Figure 2.1: **Concept of decoding semantics from neural embeddings.** The objective is to study the vector space of Large Language Models during their inference process. This aims to extract the semantic information encoded in their internal representations, accomplishing information decoding from neural embeddings.

particular tasks, domains, or model classes.

*Predictive approaches* treat neural embeddings as fixed features and analyse the human-interpretable properties encoded in these vector representations, typically casting the probing task as a classification problem. Generally, this family of probing methods uses probing classifiers (e.g., supervised probes), linear separability analyses, or feature attribution techniques (e.g., logit attribution) to uncover correlations between the model's internal representations and its outputs. However, this objective, often referred to as Eliciting Latent Knowledge (ELK), tends blur the distinction between information being present in the representation and it being actually accessible to the probing model. In addition, this complicates the distinction between actual information decoding and probe-induced learning. In contrast, *generative approaches*, such as activation patching, attempt to reconstruct inputs or synthesize outputs from neural embeddings, offering a more holistic view of what the representation encodes, although they can be sensitive to model capacity and training objectives.

Another important distinction concerns causality, as interpretability methods for neural embeddings can be broadly categorized into causal and non-causal approaches. *Causal approaches* alter neural embeddings to test functional importance of specific vector directions or dimensions. These methods often rely on activation patching (Section 2.2.1) to implement feature ablation, counterfactual manipulation, or causal mediation analysis. In contrast, *non-causal approaches* focus on observational patterns such as clustering, similarity metrics, or visualization of latent features, focusing on feature extraction from model's internal representations. This family of non-causal probing includes supervised probes, Sparse Autoencoders (SAEs), as well as Vector Symbolic Architectures (VSAs, Section 2.2.2).

While causal approaches promise stronger claims about mechanistic roles, they are often harder to scale and require careful assumptions, whereas non-causal techniques remain more practical but risk misinterpretation. The two paradigms, however, offer complementary prospectives: casual probing methods focus on where information is used to produce a model's behaviour, whereas non-casual methods study what information is represented internally for a specific behaviour. Overall, the field is moving toward hybrid strategies that combine probing, intervention, and generative reconstruction, yet challenges remain in disentangling distributed representations, ensuring faithfulness, and aligning interpretability results with model behavior in real-world settings.

### 2.2.1 Activation Patching

Activation patching is a causal interpretability method used traditionally to identify which internal representations of a neural network are causally responsible for a particular behaviour or output [33]. At a high level, this technique works by intervening on a model's intermediate activations during a model's inference, namely the latent representations of a transformer-based language model. Given a language model $\mathcal{M}$ with $L$ hidden layers and a prompt $\mathcal{S}$ consisting of a textual input of $T$ tokens. Its latent representations are defined as a three-dimensional tensor $\mathbf{H}$: neural embeddings with latent dimension $D$, as produced at every layer $l \in [1, \ldots, L]$ for each token during inference:

$$\mathbf{H} = \text{HiddenStates}(\mathcal{M}, \mathcal{S}) \in \mathbb{R}^{L \times T \times D}.$$

The standard setup for activation-based interventions involves two distinct objects:

1. a *target/clean* input ($x_{\text{tg}}$), for which the model produces the correct or expected behaviour;

2. a *source/corrupted* input ($x_{\text{src}}$), which differs in some targeted way and causes the model to fail or behave differently.

The key idea is to run the model on the source input ($x_{\text{src}}$), but at a chosen model's component (e.g., a layer or neuron), replace the source activation ("patching" operation) with the corresponding activation from the target run ($x_{\text{tg}}$). If this patch restores the target output (fully or partially), it provides evidence that the patched component contains information that is causally important for the behaviour under study [108].

Formally, let $H^l(x) \in \mathbb{R}^{T \times D}$ denote the neural embeddings at layer $l$ for input $x$. Activation patching replaces a neural embedding representation from the source input

with another representation from the target input

$$H^l(x_{\text{src}}) \leftarrow H^l(x_{\text{tg}}), \tag{2.1}$$

while leaving the rest of the model's inference process unchanged. The effect of this intervention is then measured using causal validation or a task-relevant metric. This method can be applied at different granularities, including: entire layers, individual attention heads, specific neurons or channels, or even specific token positions in transformer models [51]. By relying on explicit interventions rather than correlations, this approach has been widely used to identify mechanistic circuits in transformer models, including those responsible for indirect object identification, factual recall, and arithmetic reasoning [108, 121].

While traditional activation patching focuses on where information is used to produce a target behaviour, it does not directly address what information is represented in a given neural activation. Patchscopes, proposed by Ghandeharioun *et al.* [64], extends and unifies activation-based interventions by reframing them as a general framework for inspecting and decoding latent representations. Rather than measuring the causal effect of a patched activation on a predefined task, Patchscopes uses the model's own generative capabilities to translate its internal states (neural embeddings $\mathbf{H}$) into natural-language descriptions.

In Patchscopes [64], the latent representations extracted from a source prompt ($x_{src}$) is patched into a target prompt ($x_{trg}$) that is explicitly tasked to elicit particular information when the model completes its forward pass, such as entity attributes [64]. By carefully choosing the target prompt, one can probe what information, such as factual knowledge, is encoded in the vector space of a language model.

Conceptually, Patchscopes can be seen as a generative extension of activation patching [64]. While activation patching evaluates the causal contribution of representations to downstream behaviour, Patchscopes uses similar interventions to reveal the semantic content of those representations. These methods provide complementary tools for mechanistic interpretability [51]: activation patching localises functional components of a model, while this activation-based paradigm, Patchscopes by Ghandeharioun *et al.*, helps interpret the semantic information those components encode.

### 2.2.2 Vector Symbolic Architectures

Vector Symbolic Architectures (VSAs)[2] are a computational framework [144] inspired by cognitive science [72, 130], increasingly used to map neural representations to human-readable symbols. It has been used for tasks ranging from visual problems such as multi-attribute digit recognition [61] and Raven's progressive matrices [75] to mechanistic interpretability of language models [94].

This symbolic framework is grounded in the core assumption that individual entities or data structures can be represented as random points in a high-dimensional vector space (codebook of concepts, Figure 2.2). Leveraging the concentration of measure phenomenon [89, 96], this framework enables the representation of an exponentially large number of distinct concepts using random vectors, nearly orthogonal to each other with high probability. To represent a set of predefined elements $N$, such as contextually relevant concepts, a codebook $\Phi$ is introduced, serving as a dictionary that maps predefined concepts to their associated hypervectors (Figure 2.2). Afterwards, orthogonality and hypervector algebra combine sets of random hypervectors with a few elementary operations [63] to build complex and compositional concepts.

**VSA codebook.** Over recent years, a variety of VSA implementations have been proposed, differing in their choice of vector space as well as in the implementation of VSA operators [144]. For instance, the Multiply-Add-Permute architecture (MAP-Bipolar; 63) adopts bipolar hypervectors in $\{-1, 1\}^D$. The dimensionality $D$ is typically chosen between $10^2$ and $10^4$ based on the cardinality of the concept set [88] and the complexity of subsequent representations. MAP-B can theoretically encode $2^D$ orthogonal, independent elements [144]. Its codebook $\Phi \in \{-1, 1\}^{n_c \times D}$ stores $n_c$ atomic concepts as bipolar random vectors, generated deterministically from seeds to ensure orthogonality and independence. As shown in Figure 2.2, VSA codebook $\Phi$ is constructed by associating each random vector with a user-defined semantic concept. Afterwards, the codebook is used for concept retrieval by measuring similarity between a VSA-based representation and known concept vectors $\phi \in \Phi$. Since MAP-B operates in the bipolar domain, cosine similarity is used for vector comparison [144].

**Hypervector algebra.** The algebra of hypervector [89] is based on two main operations: *binding*, and *bundling*. These two vector operations enable the representation of complex cognitive structures, such as textual propositions, in a distributed and noise-

[2] Also known as *Hyperdimensional Computing*.

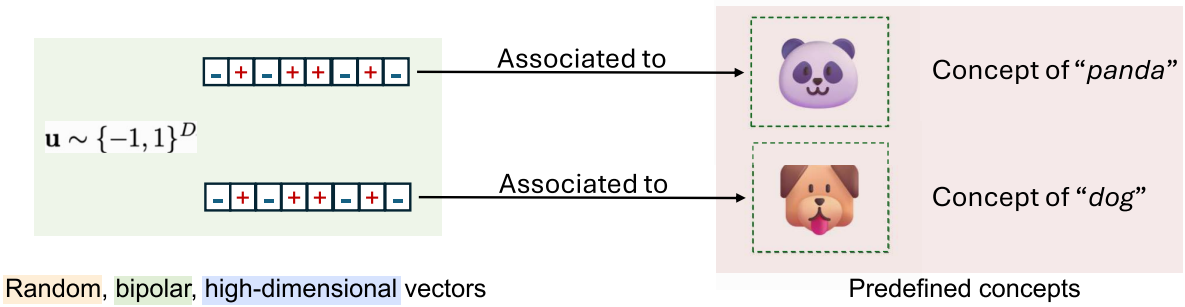



Figure 2.2: **Codebook creation of Vector Symbolic Architectures.** Random and high-dimensional vectors are associated to domain-relevant semantic concepts.

tolerant manner [63, 89].

The operation of *binding* ($\odot$) encodes input features with their associated values. For example, it can associate concepts with contextual information, such as (USA $\odot$ dollar). Whereas, the *bundling* operation ($+$), or superposition, creates set of (contextualized) concepts by combining multiple concepts into one, such as (USA $+$ Mexico). The resulting bundled vector is by design similar to each of its constituents, enabling retrieval. Binding is obtained via Hadamard product (element-wise) while bundling is element-wise sum. Polarization (sign) is typically required after bundling [93] to maintain the bipolar domain. This process however blends the parts irreversibly, diminishing their similarity to the originals in proportion to their number.

Lastly, *unbinding* ($\oslash$) recovers elemental vectors from a binding operation by factoring out one vector via multiplication with its inverse; itself in MAP-B.

**Neural probing.** For the interpretability of neural embedding, the codebook $\Phi$ defines the feature space, the set of all possible input features such as:

$$\texttt{``dog''} \rightarrow \phi_{dog} \in \Phi, \text{ and } \texttt{``animal''} \rightarrow \phi_{animal} \in \Phi.$$

whereas hypervector algebra enable creating VSA-based representations based on the targeted input's features (e.g., lexical semantics). For example, the VSA encoding ($y$) of an input sentence ($x$) might be represented as:

$$x = \textit{“The dog sat beside Charlemagne's throne.”}$$
$$y = (\phi_{\text{dog}} \odot \phi_{\text{sit}}) + (\phi_{\text{charlemagne}} \odot \phi_{\text{throne}})$$

On the other hand, when investigating the concepts encoded in a VSA representation with unknown semantics, concept retrieval can be achieved either through hypervector

algebra (e.g., recovering bound elements via the unbinding operation) or by directly comparing the representation against known concept vectors from the codebook $\Phi$.

# Chapter 3

# Proposed Approaches

This chapter presents the novel approaches proposed in this PhD thesis:

- Extract structured insights from textual documents using a pipeline powered by LLMs [20] in Section 3.1;
- Decode semantic information from neural embeddings of LLMs using:
  - Activation patching [21] in Section 3.2;
  - Vector Symbolic Architectures [22] in Section 3.3.

## 3.1 Deriving Structured Insights from Textual Documents

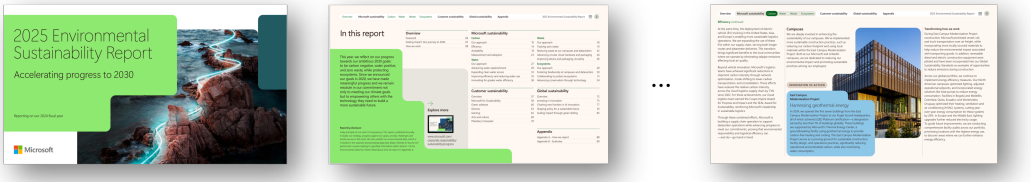
The proposed methodology uses a language model to convert verbose sentences into structured and graph-ready triplets. The language model is tasked with transforming textual inputs into standardized triplets aligned with the ESG domain. Subsequently, these triples are modelled as connected graphs and leveraged to analyse the statistical properties and interpretability of companies' non-financial disclosures. Additional details are provided in Chapter A, within the published paper [20] and its accompanying supplementary material document.[1]

### 3.1.1 Problem Statement

Given a textual document $\mathcal{D}$, a PDF file, that contains non-financial disclosures (Section 2.1) of a company, the objective is to automatically extract and isolate structured

[1] Supplementary Material document from Bronzini et al. [20].

Given a **textual document** $\mathcal{D}$ that contains *non-financial disclosures* of a company:




$d \in \mathcal{D}$ = "Microsoft has invested €125 million in cutting-edge recycling technologies and smart waste management systems in its offices in Zwijndrecht."

---

$f(d) = \{$"~~Microsoft~~", "~~invested~~", "~~125 million~~"$\}$

$f(d) = \{$

CATEGORY : "Waste",

PREDICATE : "Investment in",

OBJECT : "cutting-edge recycling technologies and smart waste management systems"

$\}$

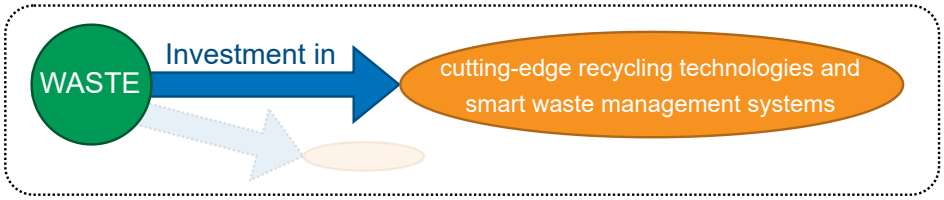



Figure 3.1: **Concept of triple generation with ESG-aligned content.** A sentence $d$ is first extracted from a company's report $D$, and then is processed by function $f(d)$ to extract ESG-related structured information. This function transforms the unstructured textual input into a triple with a predefined structure (category, predicate and object), performing ESG-driven information extraction. Finally, the generated triple can be visualized as a connected graph.

insights into these company's textual disclosures. The desired outcome is to generate structured representations aligned with a user-defined schema (i.e., triples centred on non-financial information), rather than based on the syntactical structure of the sentence such in conventional Subject-Predicate-Object (SPO) triples.

Specifically, for each textual sentence $d \in D$, the objective is to generate a triple that presents an **ESG topic** mentioned directly or indirectly in the sentence, a **predicate** affecting that topic, and an **object** related to the topic undergoing the predicate, as exhibited in Figure 3.1. Thus, the information extraction process relies on the implicit ESG-related topic of the sentence (e.g., `WASTE` in Figure 3.1) rather than its syntactic-based subject (e.g., an explicit noun, such as "Microsoft" in Figure 3.1).

This can be achieved by leveraging a Large Language Model with different prompting strategies to steer the Information Extraction process toward the desired output. A language model $\mathcal{M}$ can be provided with a prompt $\mathcal{S}$ containing specific instructions about the target domain and the required output structure, along with a sentence ($d$) extracted from the company report ($D$). In particular, RAG can be used to enrich the prompt with domain-specific knowledge (e.g., ESG topics), while ICL can improve triple generation by supplying relevant in-context examples.

**Approach.** Section 3.1.2 begins by introducing the data sources for this work, from companies' non-financial reports to ESG rating scores. This is followed by a description of the LLM-based methodology used to extract structured insights from verbose textual documents. This includes data preparation (Section 3.1.3), triple generation (Section 3.1.4) and evaluation (section 3.1.5), as well as creation of graph representations in Section 3.1.6. Lastly, Section 3.1.7 presents the various approaches used to perform statistical analysis on the resulting structured information.

### 3.1.2 Data Sources

This section outlines the three data sources used in this work: (1) sustainability reports from globally diversified companies, (2) an ESG categorization framework, and (3) ESG ratings with associated company information.

#### Sustainability Reports

Sustainability reports are non-financial documents published by companies to disclose information concerning the impact of their activities on the environment and people. Therein are described the actions the company took or expects to take regarding ESG

Table 3.1: **Overview of the considered companies for the ESG-driven study.** Companies' sustainability reports grouped by sector. Each sector showcases the number of companies represented. The `Companies` column shows a glimpse of the representative companies within each sector, offering a snapshot of the prominent companies chosen.

| Sector | Number | Companies |
|---|---|---|
| Industrial | 20 | Airbus, 3M Corporation, Adecco, Daikin Industries, ... |
| Information Technology | 16 | Apple, STMicroelectronics, Intel, NVIDIA, LG Display, ... |
| Financials | 13 | Bank of America, Deutsche Bank, Visa, Mastercard, ... |
| Consumer Staples | 11 | Coca-Cola, British American Tobacco, Walmart, ... |
| Consumer Discretionary | 11 | Amazon, Adidas, Toyota Motor, Alibaba, Tesla, ... |
| Communication Services | 10 | Netflix, Fox, Walt Disney, Activision Blizzard, Meta, ... |
| Health Care | 10 | Amplifon, AstraZeneca, Bayer, Johnson&Johnson, ... |
| Materials | 9 | United States Steel, ArcelorMittal, DuPont, Croda, ... |
| Energy | 9 | Saudi Aramco, TotalEnergies, Paramount Resources, ... |
| Real Estate | 8 | China Evergrande, Park Hotels Resorts, British Land, ... |
| Utilities | 7 | Enel, American Electric Power, Edison, Uniper, ... |
| | 124 | |

matters, such as respect for human rights, fair treatment of employees, anti-corruption and bribery as well as board diversity [54]. However, non-financial reporting can be subjective and opaque due to the complexity of reporting qualitative aspects (Section 2.1), particularly those related to social and governance issues [44]. Furthermore, the lack of a standardised framework for ESG reporting makes quantitative and comparative analyses rather challenging [5, 120].

I initially collected 6,456 sustainability reports from 4,222 different companies using the report URLs available on two public websites [80, 147]. Although these sustainability reports are mainly written in English (94% of all the available documents), the nationality of the companies is fairly diversified, covering 74 different countries. However, the majority of the reports available (56%) come from North American companies. For this study, only English-written documents are considered due to their broad coverage and the extensive availability of pre-trained language models for this language (with many thousands of models available on the Hugging Face platform; 79).

Concerning the period covered, documents were gathered up to companies' fiscal year 2022 (9.6% of the available documents), even though the majority of the documents (54%) refer to fiscal year[2] 2021. This temporal distribution is expected, as the sustainability reports were collected in February 2023, and most of the reports published during 2022 disclose information related to the previous fiscal year (2021).

[2]A fiscal year can start and end at any point, as long as it comprises 12 full months, the most common fiscal year in the business world being the calendar year.

Nevertheless, processing more than six thousand sustainability reports from four thousand different companies poses a significant computational challenge, especially when using resource-intensive generative LLMs that require a significant increase in computational budget. To showcase the proposed pipeline and present meaningful insights, I consequently select a representative subset of companies by balancing sector, region, company notoriety, size and age representation. For the notoriety criteria, I subjectively consider well-known companies with high capitalization (e.g., NVIDIA), low ESG-related performance (e.g., Saudi Aramco) and recent controversies (e.g., First Republic Bank). This subset compounds to 124 companies (Table 3.1); spanning 3 continents, 15 countries, and 11 sectors according to Global Industry Classification Standard (GICS).[3] The sample covers companies established from the 19$^{th}$ to the 21$^{st}$ century exhibiting a wide range of market capitalization from 0.4 USD billion to 2,901 USD billion. For this study I only considered the most recent sustainability report for each company in the selected subset, so to avoid any skewness effect. Note that the proposed methodology can be used as-is for longitudinal studies by using sustainability reports referred to several years.

Further details concerning the distribution of the fiscal years considered for this work, the complete list of the considered companies and the original dataset can be found in the Supplementary Material (SM) document of the published paper [20].

### ESG Categorization

This work exploits the list of ESG topics identified by Berg *et al.* in *Aggregate confusion: The divergence of ESG ratings* [15]. The authors grouped, using a bottom-up methodology, nearly seven hundred ESG-related indicators from six distinct ESG data providers[4] into a unified categorization comprising sixty-four ESG categories. These categories encompass environmental, social and corporate governance issues including Employee Development, Supply Chain, Climate Risk Management, Energy, Financial Inclusion, Biodiversity, Customer Relationship, Access to Basic Services, and Board Diversity. Section A.1 reports the complete list of all the ESG categories.

### ESG Scores and Company Information

Among providers of ESG rating scores (Section 2.1), this work adopts the Refinitiv platform,[5] which provides high-quality financial and ESG data. For each company,

[3] wikipedia.org/wiki/Global_Industry_Classification_Standard
[4] Sustainalytics, S&P Global, Refinitiv, Moody's ESG, MSCI, and MSCI-KLD.
[5] eikon.refinitiv.com – recently rebranded as LSEG Data&Analytics

twelve company-specific attributes were retrieved from the Refinitiv platform, including ESG scores, company characteristics and annual financial information.[6] Specifically, this work gathered: the combined ESG scores, individual scores for the E/S/G pillars, company sector, industry, country, region and continent, number of employees, market capitalization, EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization), and total liabilities.

### 3.1.3 Data Preparation

The proposed pipeline consists of several NLP components to extract structured insights from sustainability reports, including a stage of data preparation (Figure 3.2). This section describes the pre-processing steps for extracting textual content from PDF files, segmenting sentences, and semantic searching to isolate sentences related to Environmental, Social, and Governance aspects based on the ESG categorization presented in the previous section.

#### NLP Pipeline

Sustainability reports are generally lengthy and visually rich documents (see Figure 3.1); our sample, for example, has a median page value equal to 61 pages. Unfortunately, the incentive of companies to present visually appealing infographics and tables results in degraded quality of standard tools for text extraction.

Hence, I rely on a PDF parser (PyMuPDF; 8) to extract texts from PDFs, and then apply conventional preprocessing methods to improve the quality of the extracted text. This reduces the artefacts generated by the parser. Specifically, regular expressions (Regex in Figure 3.2) add a full stop between two sentences when missing, remove new lines in the middle of sentences and remove duplicate white spaces and lines.

Processing textual data also requires defining the granularity of the input data according to purpose, needs and limitations. A text corpus contains textual items representing singular tokens, words, sentences, paragraphs, or entire documents. I adopt sentence-level textual inputs as a good trade-off between the semantic meaning conveyed by a sentence and technical limitations (e.g., the maximum prompt length of the language model). Consequently, after extracting textual content from sustainability reports, I decompose each text into sentences with PySBD [142], a tool widely considered the state of the art in Sentence Boundary Disambiguation (SBD). However, sentence-level segmentation also has some drawbacks; for instance, it can disrupt co-reference

[6] Regarding the same fiscal year of the considered sustainability reports.

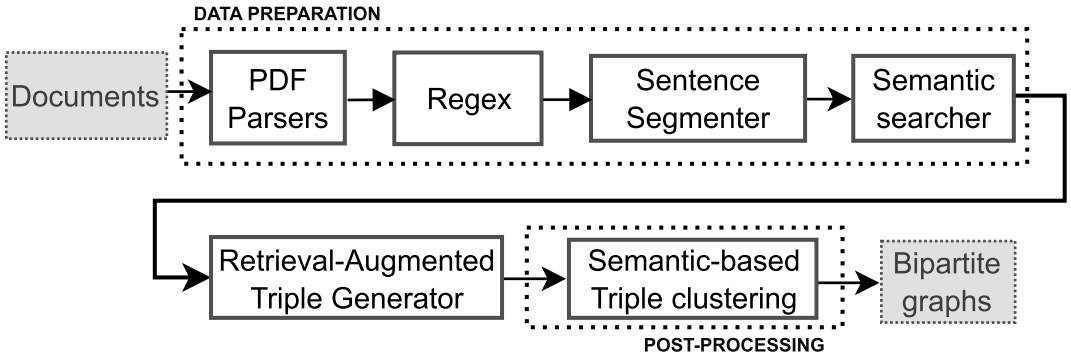



Figure 3.2: **Components of the approach proposed to extract structured insights from textual documents.** Given a collection of textual documents as inputs and preprocessed using different NLP methods, semantically structured insights are extracted by the retrieval-augmented triple generator. Bipartite graphs are created after performing a semantic-based triple clustering.

relationships across sentences, where a pronoun (e.g., "it") refers to a noun explicitly mentioned in the preceding sentence. Although incorporating a co-reference resolution method would improve the overall completeness of the textual input, its impact on the subsequent information extraction process is likely to be limited, since triple generation is guided by the implicit ESG-related topic of each sentence rather than its syntactic subject (see also Figure 3.1).

### Semantic Search

Sustainability reports include generic and vague statements, for example, phrases such as "Air is something that surrounds us 24 hours a day" from Daikin's 2022 sustainability report. Accordingly, a filtering process is required to consider only ESG-related sentences for downstream analysis. The two most well-known filtering methods for information retrieval are keyword-based and vector-based search [36]. I adopt the technique of neural semantic search [13, 69, 113, 135], a vector-based search method, that exploits text embeddings to represent both documents and queries in the same vector space. Relying on text embeddings allows one to move towards a semantic-oriented filtering approach, reducing reliance on individual keywords as well as on the adopted ESG classification framework. This representation allows one to measure the semantic similarity between a document and a query by simply computing the distance, and contrarily the similarity, between their corresponding embedded vectors [13].

This retrieval task involves discovering sentences related to each of the ESG categories (Section 3.1.2) within each sentence-segmented sustainability report. This implies

Table 3.2: **Example of the top three sentences selected for two ESG categories.** The approach is capable of retrieving sentences that pertain to two considered topics. However, it may also pick some meaningless sentences, such as the first one for the `Waste` topic, which comes from an infographic or a tabular layout.

| ESG topic | Similarity score | Sentence from company's report |
|---|---|---|
| Labor Practices | 0.73 | It also guarantees *the implementation of legislation on workers' rights* defining appropriate application standards ... |
| | 0.73 | All Group workers, both in Italy and in Brazil, are covered by *Collective Labour Agreements* reached with trade union organizations and ... |
| | 0.72 | In addition to the protections and rights provided by law and *the national collective labour agreement* for the sector ... |
| Waste | 0.68 | *Total amount of waste by type* and disposal method 306-2 KPI A1.3 & KPI A1.4 Environmental Management |
| | 0.64 | At the same time, to *avoid unnecessary material loss and waste*, the Group has formulated loss rate standards for materials ... |
| | 0.68 | Through continuous publicity and education on *garbage classification*, the project wastes were gradually reduced, recycled and harmless ... |

working in a setting of asymmetric semantic search in which queries (ESG categories) and document items (company sentences) are not interchangeable as they represent different semantic object types and have different lengths; similarly to the question answering framework [113, 135]. In contrast, symmetric semantic search is adopted when queries and corpus documents are interchangeable such as in similar document retrieval systems [25, 126, 162]. After extensive testing, I found that "INSTRUCTOR-xl" [118], an instruction-tuned embedding model [148], is the most suitable choice for this specific task. This approach offers a universal instruction template[7] along with an example list; enabling generating sentence embeddings according to textual descriptions. For embedding the ESG categories (queries), the instruction used is "`represent the title for retrieving relevant statements`". To embed the sentences (document items), the instruction applied is "`represent the statement for retrieval`".

After generating text embeddings for a sentence-segmented sustainability report, I retrieve the most semantically relevant sentences for each ESG topic (Table 3.2; Semantic Searcher in Figure 3.2). The similarity cut-off threshold $t_{sim}$ was set to 0.6 in order to retrieve sentences relevant to a ESG topic. Empirical experiments show an

[7]Instruction template: "Represent the [*domain*] [*text type*] for [*task objective*]:".

acceptable relevance for sentences with a similarity threshold equal to or above 0.4. Thus, I adopt a cut-off point of 0.6 as a good trade-off between sentence coverage and computational workload. In addition, the top $k$ sentences (with $k = 30$) were considered to limit the number of retrieved sentences following the aforementioned trade-off. This filtering layer also helps us reduce the computational time of the follow-up steps (e.g., the inference of the generative language model) and prune the resulting KG.

### 3.1.4 Retrieval-Augmented Triple Generation

Since this work focuses on linking ESG aspects, companies, and disclosed actions through Knowledge Graphs (KGs), ESG-related sentences must be expressed in a unified and standardized format. Namely, triples based on a predefined semantic template. Precisely, each ESG-oriented triple (`cat-pred-obj`) should consist of an ESG category (`cat`) representing a non-financial topic[8] mentioned directly or indirectly in the sentence, a predicate affecting that category (`pred`), and an entity (`obj`) related to the ESG category undergoing the predicate. In addition, an action (`act`) is defined as the concatenation between the ESG category (`cat`) and the predicate (`pred`) of the triple.

Although the latest approaches for Open Information Extraction (OIE) incorporate semantic information to extract triples from texts, they rely heavily on syntax (Section 5.2). However, generating domain-driven structured information requires going beyond the syntactic structure of input text, moving from pure syntax-based extraction toward the definition of a semantic-based and domain-oriented schema for the desired triples. For example, given the sentence: "*Microsoft has invested 125 million in cutting-edge recycling technologies*",[9] conventional OIE techniques [6] would identify and generate a traditional Subject-Predicate-Object triple as the following:

`(Microsoft, Invested, 125 million).`

Although the above SPO triple can well represent the semantic meaning conveyed, it would not be suitable for the objective of this work. Indeed, the ideal triple, centred around ESG-related aspects, would have been:

`(Waste, Investment in, Cutting-edge recycling technologies).`

Where the concatenation of `Waste` (`cat`) and `Investment in` (`pred`) results in the action `Waste:Investment in` (`act`). However, extracting such triples requires defining

[8] The terms "category" and "topic" are interchangeably used.
[9] This sentence is created for explanatory purposes, not reflecting real information.

a non-trivial schema for the desired structured information, where each attribute is shaped by a domain-aligned, semantically grounded meaning. For example, the entity `Waste` (ESG category) is not explicitly mentioned, although it could be inferred from the term "*recycling technologies*". This type of inference involves jointly information extraction and semantic classification tasks. Since this work focuses on ESG-related information, triples should also represent predicates and objects related to an ESG topic rather than generic statements.

Large Language Models have already demonstrated abilities in semantic understanding and handling a wide range of NLP-related tasks [23, 137]. This work employs an instruction-tuned LLM ($\mathcal{M}$) to address this information extraction challenge, alongside specific prompting techniques such as In-Context Learning (ICL), and Retrieval-Augmented Generation (RAG). RAG is used to include ESG-oriented information within the model's instruction, guiding Information Extraction towards the ESG domain using the list of ESG categories and triple descriptors. While, ICL enables providing the language models with domain-aligned examples of input-output pairs.

Specifically, the `Kor` Python library [164] is used to define the schema for the textual instructions used during model inference. `Kor` allows to construct a prompt ($\mathcal{S}$) programmatically by specifying the semantic data schema for the desired output as well as including input-output examples. The in-context example provides the language model with the association of a textual input (company's sentence) with the desired output (ESG-oriented triple; `cat-pred-obj`). Figure 3.3 exhibits two examples included in the model instruction to leverage the ICL abilities of LLMs. In the model instruction $\mathcal{S}$, each element of the triple template (`cat`, `pred` and `obj`) is described with a expressive name and textual description conveying its semantic meaning. The `Kor` library then uses this information to generate, by means of a predefined template, a textual

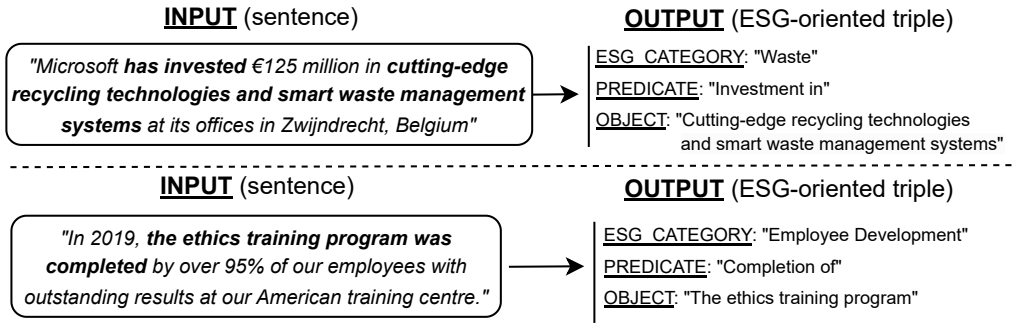



Figure 3.3: **In-context examples included in the prompt.** These sentences were created to cover different syntactical structures, do not represent actual information.

instruction to prompt a language model (Figure 3.4).

The textual description of the attribute `cat` integrates also the sixty-four ESG categories of the ESG categorization described in Section 3.1.2. As this attribute conveys the ESG-related aspect mentioned implicitly or explicitly in the input sentence, this process resembles a task of input classification. However, rather than constraining input classification strictly to an arbitrary fixed list of classes (i.e., `cat` attribute $\mapsto$ ESG categorization), this approach designs a semantically-aware generative-powered classification that remains anchored to, yet extends beyond, the original list of ESG categories provided in the model's instruction. Although the language model is sorely prompted with a fixed list of ESG-related labels, the model is expected to semantically generalize them when processing the input sentences. Leveraging its generative capabilities and semantic understanding, the language model treats the original ESG categorization as semantic reference points rather than rigid, exhaustive classes of the input sentences. Ultimately, this can enhance the Information Extraction process by yielding more granular, data-driven and faithful insights into company's non-financial disclosures. Section A.3 exhibits the full model instruction.

I tested several different instruction-tuned LLMs, including Google's Flan-T5 [31] and multiple LLaMA-based models [152]. Empirical results (Section A.5) showed that the LLaMA-based models perform better at structured information extraction, with `WizardLM-7B` [87, 160] producing the highest-quality results. The language model processes one sentence at time, as preliminary experiments in multi-sentence processing, aimed at also addressing co-reference resolution (Section 3.1.3), yielded poorer results, with generated triples often blending semantic content from multiple input sentences.

Chapter A presents further information concerning triples generation, such as exper-

```
Your goal is to extract structured information from the user's input that matches the form
described below.  When extracting information, please make sure it matches the type of
information exactly.  Do not add any attributes that do not appear in the schema shown
below.
$DATASCHEMA
Please output the extracted information in JSON format.  Do not output anything except for
the extracted information.  Do not add any fields that are not in the schema.  If the text
contains attributes that do not appear in the schema, please ignore them.  All output must
be in JSON format and follow the schema specified above.  Wrap the JSON in <json> tags.
$EXAMPLES
input:  $INPUT
output:
```

Figure 3.4: **The instruction template for constructing the prompt.** The template is created and compiled by the `Kor` library to task an LLM to extract structured data using ICL. `DATASCHEMA` is a placeholder for the output data schema. `EXAMPLES` is a placeholder for the in-context examples as input-output pairs. While `INPUT` represents an ESG-related sentence from which structured data needs to be retrieved.

imenting with different generation hyperparameters (A.2) and language models (A.5).

### 3.1.5 Quality of Generated Triples

This section adopts a LLM-as-a-judge strategy [68] to evaluate qualitatively the extracted structured information by tasking the same language model (WizardLM, 160) to evaluate triple quality. Leveraging also its ICL abilities, the model is tasked to assess how well the structured output aligns and remains consistent with the input sentence, as well as how accurately each triple component (`cat`, `pred`, and `obj`) is represented. For each input-output pair, the language model outputs a numerical evaluation scores on a scale from 0 to 3. Section A.4 shows the full model instruction.

Specifically, this analysis randomly samples 1,000 triples from all the ESG-oriented triples ($\approx 40,000$) generated from all companies' non-financial reports. The triple sample has an average score equal to 2.65, showing a fairly high quality of the structured information extracted. The ESG category (`cat` attribute) and object (`obj` attribute) show high average performance (2.76 and 2.78) in comparison to the generated predicate (`pred`, 2.5). Figure 3.5 shows the distribution of the scores for the three attributes, whereas Table 3.3 showcases the evaluation of five triples. The ESG topic mentioned directly or indirectly in the sentence is semantically captured fairly well, whereas extracting the predicate affecting it might be more challenging. For example, the predicate of the last triple in Table 3.3 displays low quality in terms of syntax and coherence with the action of the original sentence. For instance, the predicate "`Creation of`" could have been more suitable for this sentence. However, this triple notably captures the

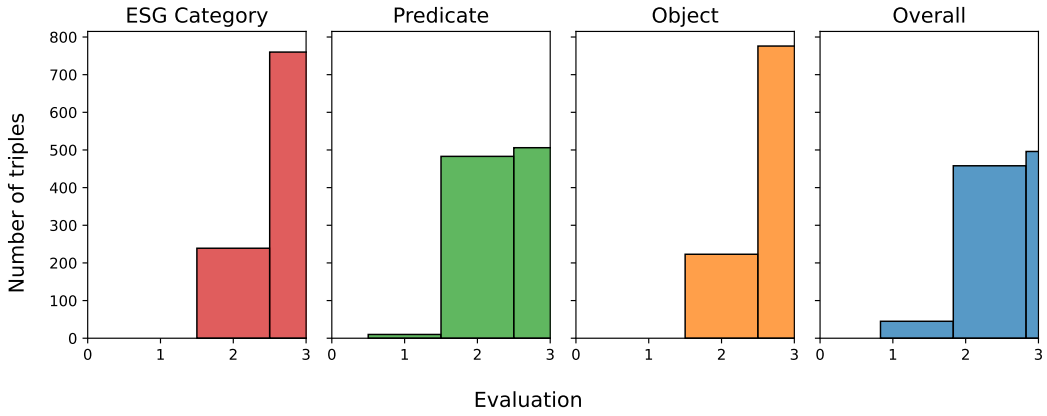



Figure 3.5: Distributions of the evaluation (0-3 scale) of a sample of 1,000 triples using a LLM-as-a-judge strategy, measuring the alignment with the desired content.

Table 3.3: Sample of evaluation scores, with a 0-3 scale, of five generated triples.

| **Document's Sentence** | **Generated ESG triple** | `cat` | `pred` | `obj` | **AVG** |
|---|---|---|---|---|---|
| We evaluate products and materials against HSE compliance, safety and sustainability metrics | `CAT`: "Product Safety", `PRED`: "valuation of", `OBJ`: "Products and materials against HSE compliance, safety and sustainability metrics" | 3 | 3 | 3 | 3 |
| With increasing digitalisation, the ability to provide a data environment meeting the highest security and regulatory standards, such as GDPR, is critical | `CAT`: "Data Privacy", `PRED`: "Meeting the highest security and regulatory standards", `OBJ`: "The ability to provide a data environment" | 3 | 3 | 3 | 3 |
| We partner with trusted, leading non-profit organizations committed to impacting the areas closest to our homes and our hearts ... | `CAT`: "Philanthropy", `PRED`: "Partnership with", `OBJ`: "Trusted, leading non-profit organizations" | 3 | 2 | 3 | 2.67 |
| That data will help us refine operations, modify purchasing decisions and meet our emission reduction goals | `CAT`: "Environmental", `PRED`: "Use of", `OBJ`: "Data" | 2 | 2 | 2 | 2 |
| We also indirectly create employment opportunities for many young people in our supply chain | `CAT`: "Supply Chain", `PRED`: "Indirect", `OBJ`: "Employment opportunities for many young people" | 2 | 1 | 2 | 1.67 |

actual main ESG topic of this tricky sentence (`Supply Chain`), despite the mention of another potential, yet secondary ESG aspect (employment opportunities). Furthermore, the second triple in Table 3.3 displays another phenomenon observed in some triples: lengthy predicate attributes enveloping direct objects. This happens especially when the disclosed actions involve transitive verbs as in this example (to meet something). Thus, despite its semantic correctness and coherence with the disclosed action, this might make our effort in generalising challenging.

This preliminary analysis addresses the research questions related to the novel methodology proposed in this work (Chapter 1), namely the use of LLM generative capabilities and prompting techniques to steer the NLP task of Information Extraction toward a specific domain in order to produce ESG-oriented structured information.

**Ablation on prompting techniques.** Since the model instruction has a great impact on the quality of the generated text [167], this ablation study compares qualitatively the triples generated using different prompting techniques:

- with/without input-output `EXAMPLES` in Figure 3.4, In-Context Learning (ICL);

- with/without the ESG-driven `DATASCHEMA` in Figure 3.4, exploiting the paradigm of Retrieval-Augmented Generation (RAG).

Including in-context `EXAMPLES` in the model's instruction (i.e., ICL) generates better triples in terms of both information completeness and semantic representation, observable respectively in the first two comparisons and the third one in Table 3.4. This also helps the model compose its responses according to a specific output format. Indeed, although the model was explicitly instructed to generate outputs as valid JSON objects, its responses conformed to the JSON schema only when using ICL.

On the other hand, incorporating the ESG-oriented triple schema into the prompt (i.e., RAG) enables the language model to focus more effectively on ESG-related information, as shown in Table 3.5. The semantic schema supplies detailed guidance on the types of information to extract, ranging from ESG-related aspects specified in the `cat` attribute to ESG-specific descriptors such as "a nominalised verb affecting the ESG-related aspect" for the `pred` attribute. The qualitative comparison in Table 3.5 demonstrates that including the ESG-aligned schema (RAG) substantially improves the extraction of information from complex sentences involving multiple causes, as illustrated by the second example in Table 3.5. By enabling the integration of a predefined list of ESG-related aspects into the semantic description of the `cat` attribute, RAG improves the model's ability to align the extraction of structured data with the ESG domain. This also facilitates the alignment of extracted ESG-related aspects with

Table 3.4: Comparison of triples generated with/without in-context examples.

| | | **ESG category** (`cat`) | **Predicate** (`pred`) | **Object** (`obj`) |
|---|---|---|---|---|
| (1) | `EXAMPLES` | Work-Life Balance | Provision of | A policy providing clarity around flexible work options |
| | without `EXAMPLES` | Family Friendly Policies | Provides | Flexible work option |
| (2) | `EXAMPLES` | Health and Safety | Creation of | COVID-19 safety tips for ridesharing |
| | without `EXAMPLES` | Public Health | Created | COVID-19 safety tips |
| (3) | `EXAMPLES` | Climate Risk Management | Partnership with | CoGo to provide personalised carbon footprints |
| | without `EXAMPLES` | Climate Risk Management | Announced | Partnership |

Table 3.5: Comparison of triples generated with/without the ESG-oriented schema.

| | | **ESG category** (`cat`) | **Predicate** (`pred`) | **Object** (`obj`) |
|---|---|---|---|---|
| (1) | `DATASCHEMA` | Work-Life Balance | Provision of | A policy providing clarity around flexible work options |
| | without `DATASCHEMA` | Work-Life Balance | Providing clarity around | Flexible work options available to parents and caregivers |
| (2) | `DATASCHEMA` | Philanthropy | Partnership with | The Bill and Melinda Gates Foundation and the Western Cape Department of Health |
| | without `DATASCHEMA` | Health Care | Delivery of | Over 1.4 million prescriptions |
| (3) | `DATASCHEMA` | Climate Risk Management | Partnership with | CoGo to provide personalised carbon footprints |
| | without `DATASCHEMA` | Carbon Footprint | Partnership with | To provide personalised carbon footprints |

the ESG categorization scheme, thereby improving the standardization of the resulting structured data. For example, in the third comparison, the ESG category is mapped to `Climate Risk Management`, a broader topic defined in the original ESG categorization, rather than the overly specific term `Carbon Footprint`.

### 3.1.6 Knowledge Graph Generation

Before constructing a graph representation (KGs) using the `cat-pred-obj` triples extracted from companies' non-financial documents, data cleaning process is applied to reduce semantic redundancy in the structured information. Reducing semantic redundancy in the extracted information is essential, as a knowledge graph containing multiple nodes or edges with equivalent or highly similar meanings can negatively affect subsequent interpretability analyses.

**Redundancy of the extracted information.** Redundancy is defined as the existence of semantically equivalent textual forms within the values associated with triple attributes. For instance, the expressions *"Partnership with", "Collaboration of", "Working together with"*, and *"Partnership with others to"* for the `pred` attribute (Table 3.6) convey highly similar meanings and should therefore be mapped to a common semantic concept. This semantic redundancy frequently emerged, during manual inspection,

for the `pred` attributes, as companies use plenty of expressions to describe the same actions, such as:

$$\textit{Partnership} \approx \textit{Working together} \approx \textit{Collaboration.}$$

Conversely, semantic redundancy in the `cat` attribute stems from overspecific ESG topics such as: *"Supplier Audits"*, *"Supplier Relationship"*, *"Supplier Risk Assessment"*, and *"Supplier Assessment"*. This can be viewed as a drawback of adopting a generative approach. Without consolidating textual variants that share similar semantic content, the resulting graph structures would contain redundant representations (nodes/edges), leading to increased noise in subsequent analyses.

Here, semantic similarity is quantified by computing the cosine similarity between the embedding representations of the unique values associated with each triple attribute, while semantic redundancy is operationally defined by grouping textual expressions whose cosine similarity meets or exceeds a predefined threshold.

**Semantic clustering.** Consequently, semantic-based clustering is performed on all the ESG categories (`cat`) and predicates (`pred`) presented in the generated triples. The `obj` attribute is excluded from this process, as its content is strongly tied to input sentences and exhibits low cosine similarities among attribute elements (i.e., high degree of diversity), potentially leading to uninformative single-item clusters.

Initially, embedding representations of the unique textual elements are generated using the same embedding model adopted in Section 3.1.3 ("INSTRUCTOR-xl") with the model instruction "`Represent the title`". Secondly, semantic clusters are discovered as high-density regions in the embedded vector space using cosine similarity as a

Table 3.6: Sample of semantic clusters for the `cat` attribute (ESG categories; left) and `pred`icates (right), with elements grouped by a cosine similarity threshold of 0.8.

| **Cluster label** | **Cluster elements** | **Cluster label** | **Cluster elements** |
|---|---|---|---|
| ESG Commitments | ESG Metrics<br>ESG Incentives<br>ESG Promotion<br>... | Assessment of | Monitoring of<br>Assessment on<br>Assessment and monitoring of<br>... |
| Product Sustainability | Sustainability<br>Sustainable Products<br>Supplier Sustainability<br>... | Partnership with | Collaboration of<br>Working together with<br>Partnering with others to<br>... |
| Supplier Assessment | Supplier Audits<br>Supplier Relationship<br>Supplier Risk Assessment<br>... | Opportunity to | Creation of opportunity<br>Opportunities to work on<br>Opportunity to contribute to<br>... |

metric. Semantic redundancy is thus defined by grouping textual expressions with a cosine similarity equal or above a predefined similarity threshold. Empirical analysis was conducted to assess cluster quality across similarity thresholds ranging from 0.5 to 0.9. After manual inspection, the similarity threshold was set to 0.8 as a trade-off between semantic coherence (i.e., semantically equivalent textual forms) and cluster size (i.e., generalization vs. overspecification). Reducing the threshold would decrease cluster quality by grouping together elements with different underlying meanings, while also affecting the subsequent analysis, as excessive generalization could alter the semantic meaning conveyed by the original triples. Each cluster is then labelled according to its centroid, For instance, the cluster denoted as `Partnership with` groups 103 similar predicate elements including all the aforementioned collaboration-related elements. Table 3.6 shows a sample of other clusters and their elements.

Finally, the cluster labels are used to replace the ESG categories (`cat`) and predicates (`pred`) in the original triples, such as:

**cat** : "Health and Safety" $\mapsto$ "Employee Safety"
**pred** : "Designed to continually improve" $\mapsto$ "Improving"
**obj** : "Workplace safety culture".

Figure 3.6 exhibits a portion of the final Knowledge Graph, aligned with the ESG domain, generated using the methodology described in Section 3.1.

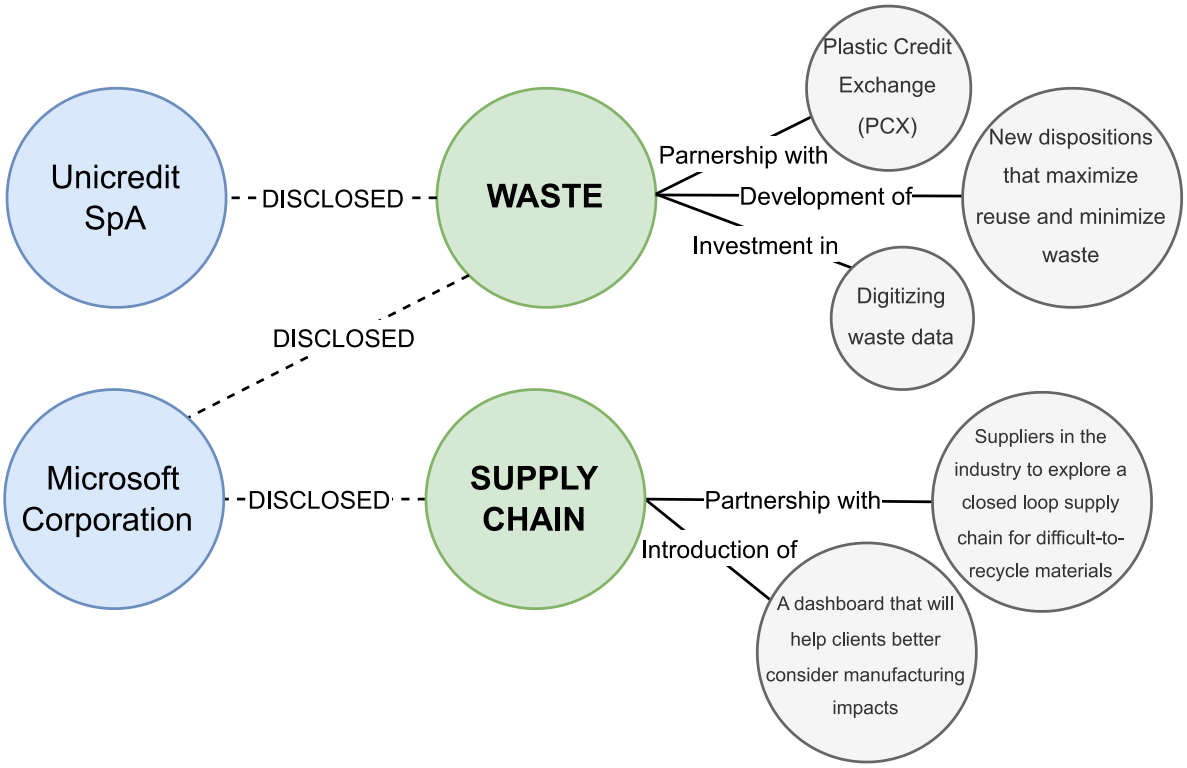



Figure 3.6: **A portion of the Knowledge Graph generated using the proposed methodology.** It portrays the ESG-oriented triples generated using our approach. Blue nodes represent company nodes which are connected to the ESG categories (green nodes) disclosed in the companies' sustainability reports. Category nodes (`cat`) are connected via a labelled edge (`pred`) to the predicate object (`obj`, grey nodes).

### 3.1.7 Approaches for Statistical Analyses

The Knowledge Graph populated with insights extracted from companies' non-financial reports (Figure 3.6) is examined using distinct undirected bipartite graphs.

A bipartite graph is a graph whose vertices can be divided into two distinct and independent sets/partitions [9, 53, 115]. It can be described through the binary *bi-adjacency* matrix $\mathbf{B}$, a $\{0,1\}^{n\times m}$ matrix where $n$ and $m$ are the numbers of nodes in the two partitions. *A bipartite graph can consequently be seen as a special type of knowledge graph whose nodes can be divided into two distinct and independent partitions*, whose edges vary according to the perspective from which it is constructed.

Specifically, three distinct bipartite graphs are constructed to analyze the structured information extracted from companies' non-financial reports by filtering nodes and edges in the Knowledge Graph (Figure 3.6) to isolate specific node types and their corresponding connections. This enables investigating companies' disclosures using three distinct perspectives (Table 3.7): the relationships between category and predicate ($\mathbb{B}_{\text{catpred}}$), between company and category ($\mathbb{B}_{\text{cocat}}$), and between company and action ($\mathbb{B}_{\text{coact}}$).

#### Statistics for Bipartite Graphs

Most of the unipartite graph metrics can be extended to the bipartite case [9, 115]. I thus compute bipartite variants of conventional network statistics, including *degree centrality, closeness centrality*, and *betweenness centrality* [18, 115].

The degree centrality of a node in one partition corresponds to the fraction of nodes in the opposite partition to which it is connected [166]. The closeness centrality of a node [53, 115] is determined by calculating its average shortest path distance to all other nodes. It represents the efficiency of a node to be connected directly to nodes from the other partition and indirectly to nodes from the same partition [57]. For instance given $\mathbb{B}_{\text{cocat}}$, a company node with a high closeness score indicates the company is connected, and thus it is close, to many category nodes which in turn are connected to several other company nodes. Lastly, betweenness centrality [12, 53] assesses the level of influence a

Table 3.7: The bipartite graphs for examining insights from companies' reports.

| Bipartite graph | | Content |
|---|---|---|
| $\mathbb{B}_{\text{catpred}}$ | category-predicate | Predicates (`pred`) disclosed with each ESG category (`cat`) |
| $\mathbb{B}_{\text{cocat}}$ | company-category | ESG categories (`cat`) disclosed by each company |
| $\mathbb{B}_{\text{coact}}$ | company-action | Actions (`act`) disclosed by each company |

node holds over information flow within a graph. In bipartite graphs, it identifies critical mediator nodes that facilitate interactions between the two distinct node partitions [9].

### Variability of ESG-related Actions

Information theory is used to assess the entropy of ESG topics based on their associated predicates extracted from companies' non-financial reports. *Shannon's entropy* [35] measures the information, and thus the variability, present in a set of events $\mathcal{X}$ through their respective occurrence probabilities $p(x)$:

$$H(\mathcal{X}) := -\sum_{x \in \mathcal{X}} p(x) \ln p(x) \tag{3.1}$$

Here, entropy measures the variability of the predicates (events $\mathcal{X}$) disclosed by all companies for a given ESG topic, where $p(x)$ denotes their relative frequencies:

- High entropy denotes high variability in the predicate occurrences, indicating a topic addressed through many actions (predicates) with an almost uniform probability of predicate occurrence;
- Low entropy indicates the predominance of a limited set of predicates for a topic.

### Similarity Analysis

Similarities among companies are computed based on jointly disclosed ESG-related actions using the *Jaccard similarity coefficient* [34], which measures the similarity between two sets as the cardinality of their intersection over the cardinality of their union:

$$J(\mathcal{A}_{c_1}, \mathcal{A}_{c_2}) = \frac{|\mathcal{A}_{c_1} \cap \mathcal{A}_{c_2}|}{|\mathcal{A}_{c_1} \cup \mathcal{A}_{c_2}|} \tag{3.2}$$

where $\mathcal{A}_{c_i}$ is the set of ESG-related actions disclosed by company $c_i$. To mitigate the influence of stochastic fluctuations on the similarity score, a null model [66] was generated with a bootstrapping technique by computing company similarities on the randomised action sets through 1,000 simulations, and subtracted the null model from the observed company similarities.

### Correlation Analysis

This analysis evaluates whether company similarities in terms of jointly disclosed ESG-related actions (Jaccard-based similarity) are correlated with other companies' simi-

larities in ESG scores or other characteristics such as market capitalization or geographical location. First, feature similarities are measured through different strategies, ensuring the same numerical range and monotonicity. The similarities in numerical features (e.g., ESG scores) are measured by computing the absolute numerical difference normalised using max-min scaling [168]. Conversely, similarities in textual features (e.g., company sectors) are represented through neural embeddings generated using the "INSTRUCTOR-xl" embedding model. Their semantic similarity is then evaluated using cosine similarity and normalized to the [0,1] range through min-max scaling [168]. Section A.7 reports the complete list of all the features used for this analysis.

Next, a bivariate analysis is conducted through a correlation analysis to measure the monotonic association between action similarities and similarities of other company's features. I rely on *Kendall's $\tau$ correlation coefficient* [1], a non-parametric and rank-based statistic, defined as:

$$\tau = \frac{n_c - n_d}{n_c + n_d} \tag{3.3}$$

where $n_c$ and $n_d$ are the numbers of concordant and discordant pairs respectively. Rank-based correlation methods overcome some limitations of traditional correlation methods such as the well-known Pearson correlation coefficient [32]: they can measure nonlinear monotonic relationships, are more robust against outliers and normality assumption is not required [125]. High positive coefficients express a high level of order consistency in the company similarities sorted according to actions' and other similarities, while high negative coefficients occur when these two similarities are sorted reversely [1].

### Interpretability of ESG Rating Scores

Lastly, the structured information extracted from companies' reports advances the interpretability of ESG scores (Section 2.1).

A linear regression model is initially used to model the relationships among companies' features (inputs) and ESG scores (outputs). This is followed by an explainability framework to identify the features with the greatest impact on companies' scores. Specifically, a first-order linear regression model is trained on companies' information, and the SHAP (SHapley Additive exPlanations) framework is then adopted to interpret the model's predictions. The predictors used in the regression model consist of features derived from the information extracted from companies' reports (Section 3.1.6), together with additional company-related information (e.g., financial data; Section 3.1.2).

**Data.** First, this work adopts as predictors the percentage of the top ten most disclosed ESG categories for each company. For example, if a company has ten percent of its generated triples concerning the ESG category `Waste`, and that is within its top ten most disclosed topics, the feature `Category:Waste` for this observation has a value of 0.1. I also consider the proportion of the E/S/G pillars based on all the disclosed categories for each company. The entropy of ESG categories (`cat`) and actions (`act`) are also computed for each company, which indirectly captures the diversity of categories and actions disclosed in its sustainability report. Finally, nine company-related features are included as additional predictors in the regression model, comprising five structural company characteristics and four annual financial metrics. Specifically, the structural features include the company's sector, country, region and continent of headquarters, and year of incorporation. The financial features, corresponding to the fiscal year of the analyzed sustainability report, are EBITDA, liabilities, market capitalization, and number of employees.

The descriptive statistics of these report-based and company-based predictors are exhibited in Figure 3.7 and Table 3.8. Categorical variables, such as the company sector, are turned into binary indicator variables (i.e., dummy variables; 45), generating a total of 97 features whereas standardisation is applied to those numerical such as market capitalization. The Supplementary Material document of the published paper [20] also reports an example of an observation, with a complete list of features.

**Regression model.** Specifically, Ordinary Least Squares (OLS) regression [42] with Elastic Net regularization [170] is employed to model the relationships among companies' information, predicting companies' ESG scores from both report-based and company-based features. Elastic Net regularization, a generalization of the Least Absolute Shrinkage and Selection Operator (LASSO; 71) method, enables for both feature selection and regularized model training. This approach enhances performance when the number of predictors ($|\text{features}| = 97$) exceeds the number of observations ($|\text{companies}| = 89$) and when strong pairwise correlations are present [170]. The OLS

Table 3.8: **Descriptive statistics of the categorical predictors for the regression model**. Variables are encoded as binary indicators for each observation.

| Feature name | Unique | Most frequent categories (%) |
|---|---|---|
| Region | 3 | Americas (56.3%), Europe (29.9 %), Asia (13.8%) |
| Sub-region | 7 | Norther America (55.2%), Western Europe (13.8%), Eastern Asia (12.6%), ... |
| Country | 15 | United States of America (51.7%), Italy (8%), United Kingdom (6.9%), ... |
| Sector | 11 | Industrials (14.9%), Information Technology (11.5%), ... |

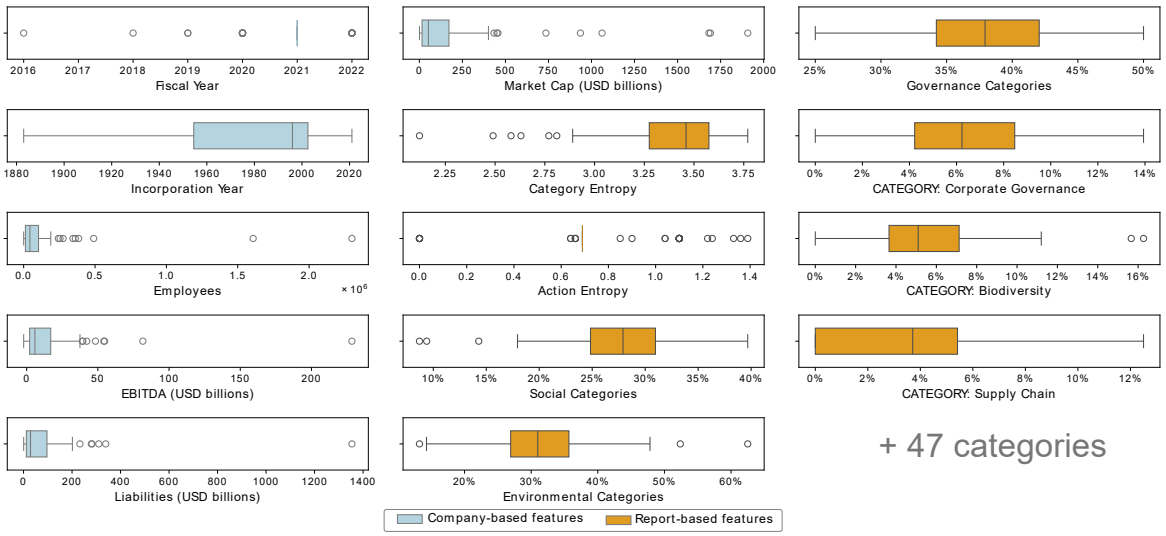



Figure 3.7: **Descriptive statistics of the numerical features used as part of the predictors of the Ordinary Least Squares (OLS) model.** The features labelled with the starting word "category" reflect the percentage disclosure of an ESG topic. To enhance readability, the figure exhibits only three category features for explanatory purposes. The statistics are presented before standardisation.

model is trained using the Elastic Net cost function and an eight-fold cross-validation approach [16].

**Performance of the regression model.** Performance of the OLS regression are here reported through different metrics. Firstly, the Coefficient of Determination $R^2$ [165] measures the proportion of variation in the dependent variable explained by the model predictors, representing the goodness of the inference ability of the model. Low coefficients express a little variation proportion explained by the model predictors, resulting in poor performance on the inference of the dependent variable (ESG scores). In contrast, in the presence of a high variation proportion explained, the model predicts the dependent variable with small errors. The regression model achieves a $R^2$ of 0.71 using the optimal alpha, demonstrating a broad variation explained by our features to infer ESG scores.

Root Mean Square Error (RMSE, 159) is also employed to measure the model's performance. It is a quadratic score, in the same units of the dependent variable, in which the average error in the model predictions is computed by averaging the squared individual errors. The model achieves an RMSE equal to 7.76, representing the average difference between the actual ESG score and the inferred one. Finally, the model achieves a Weighted Mean Absolute Percentage Error (wMAPE, 71) of 7.9%,

indicating that its predictions of companies' scores deviate on average by less than 8%.

To conclude the evaluation of the regression model's performance, a residual analysis is conducted to assess the linear assumptions underlying the model specification. The assumption of normally distributed residuals (i.e., $E_i \sim N(0, \sigma^2)$) is supported by the Anderson–Darling test [114], which yields a $p$-value of 0.066, and is further consistent with the QQ plot of residuals against the normal distribution, where the points approximately align with the reference line. Regarding homoscedasticity (i.e., constant residual variance across predictions) no clear patterns are observed in the scatter plot of residuals versus predicted ESG scores, which suggests that the assumption is reasonably satisfied. This observation is also consistent with the scatter plot of predicted versus actual ESG scores. However, a slight tendency toward overestimation can be observed for ESG scores below 50, suggesting reduced predictive accuracy in this range and indicating a potential limitation of the model in capturing lower-score values. Graphical figures for this residual analysis are provided in the Supplementary Material of the published paper [20].

**Explainability Framework.** Lastly, the SHAP framework [102] enhances the interpretability of the ESG-based regression model by identifying the company features (model's inputs) that most strongly influence the predictions of ESG scores (outputs). SHAP is a model-agnostic framework for attributing feature contributions to model predictions, grounded in cooperative game theory. It provides local explanations by decomposing each model's prediction into an additive combination of feature effects, derived using Shapley values and conditional expectation functions. Specifically, SHAP follows an additive explanation paradigm:

1. it defines a baseline value corresponding to the average model output over the dataset, given by the expected value $\mathbb{E}[f(X)]$;

2. for each individual prediction $f(x)$, it computes the contribution of each input feature on the model's prediction by measuring its marginal effect across all possible subsets of features, referred to as the SHAP value.

Consequently, the SHAP framework decomposes each model's prediction into the sum of feature contributions relative to the expected value, quantifying both the magnitude and direction of each feature's impact on the prediction. Pedictors with positive SHAP values affect the expected model output with additive increments, conversely, those with negative values have additive decrement impacts.

## 3.2 Tracing the Dynamics of Factual Knowledge in Large Language Models

Following the Patchscopes framework proposed by Ghandeharioun *et al.* [64], activation-based interventions (Section 2.2.1) are used to decode the semantics, in the form of factual information, embedded within an LLM vector space and represent its evolution across the model's hidden layers using a dynamic knowledge graph. The procedure is illustrated in Figure 3.8. Additional details are provided in Chapter B, and within the published paper [21].

### 3.2.1 Problem Statement

Given a language model $\mathcal{M}$ with $L$ hidden layers and a prompt $\mathcal{S}$ consisting of a textual input of $T$ tokens $= \langle s_1, \ldots, s_T \rangle$. The focus is on inspecting the latent representations[10] of the language model, represented as a three-dimensional tensor $\mathbf{H}$: neural embeddings with latent dimension $D$, as produced at every hidden layer $l \in [1, \ldots, L]$ for each token during the inference of $\mathcal{M}$ on $\mathcal{S}$:

$$\mathbf{H} = \text{HiddenStates}(\mathcal{M}, \mathcal{S}) \in \mathbb{R}^{L \times T \times D} \tag{3.4}$$

The objective is to extract the semantics encoded in $\mathbf{H}$ by mapping neural embeddings to human-understandable symbols, as illustrated in Figure 2.1. This work presents an approach based on activation-based interventions (Section 2.2.1) to decode factual information from $\mathbf{H}$ as natural-language texts that are both logic-oriented and compatible with graph representations:

$$\begin{aligned} \mathbf{H} &= \mathcal{M}(\text{“}\texttt{Charlemagne was crowned emperor on Christmas Day}\text{”}) \\ f(\mathbf{H}) &= \text{“IsDate(Christmas Day, December 25)} \wedge \text{ IsHoliday(Christmas Day)”} \end{aligned} \tag{3.5}$$

**The approach.** Initially, a model's behaviour is elicited by tasked a language model $\mathcal{M}$ to accomplish a downstream task (i.e., claim verification) via prompting ($\mathcal{S}$, Section 3.2.2). Afterwards, activation patching is used to execute a separate inference of the model $\mathcal{M}$ with a different prompt $\mathcal{T}$ (Section 3.2.3) to decode the semantics embedded in the vector representations of the original model's inference ($\mathcal{M}$ on $\mathcal{S}$). This prompt $\mathcal{T}$ contains a special placeholder token "$x$" for the patching process: substitut-

[10] Latent representations, residual stream, vector space, and hidden states are used interchangeably.

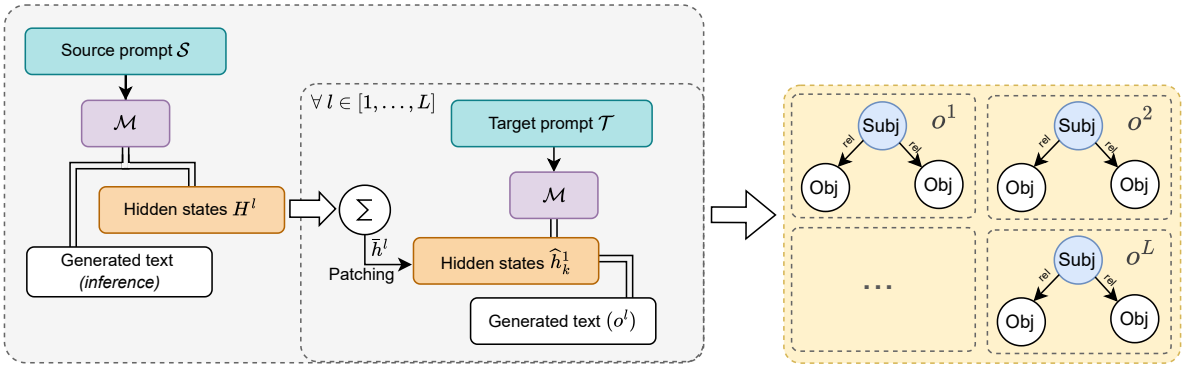



Figure 3.8: **The patching-based approach to decode the factual knowledge from the latent representations of language models**. The textual outputs ($o^l$) are then represented in a dynamic knowledge graph.

ing its neural embedding with a summary of the input's latent representations from the source inference. I specifically intervene during the model computation on prompt $\mathcal{T}$ by mapping the placeholder token's embedding with a weighted sum of the token embeddings of the input sentence obtained at the $l$-th hidden layer of the model inference on $\mathcal{S}$. The execution of this patched inference generates then a structured text $o^l$, containing a list of factual information. The procedure is repeated for the different values of $l \in [1, \ldots, L]$. All the generated texts are then represented as a dynamic knowledge graph, with $l$ being the graph's dynamic and temporal dimension (Section 3.2.4).

In summary, the framework converts the LLM internal vector space into a human-understandable semantic graph by collectively probing the encoded semantics of multiple token representations.

### 3.2.2 Prompt Definition

This section illustrates the procedure used to elicit and shape a targeted model's behaviour, namely recalling factual information related to inputs. The vector space of this model's inference will subsequently be investigated using activation-based interventions as detailed in the following section. Initially, a textual template for the model instruction is defined to elicit verification of the claim: the source prompt $\mathcal{S}$.

This prompt has three different semantic parts: a *system instruction* describing how to perform the claim verification task (pink in Figure 3.9), an *input-output example* to help the model generate the desired output (purple), and the *input sentence* (green). The desired output is a structured text with two attributes: a binary label indicating the truthfulness of the sentence and a list of facts supporting such an evaluation. The

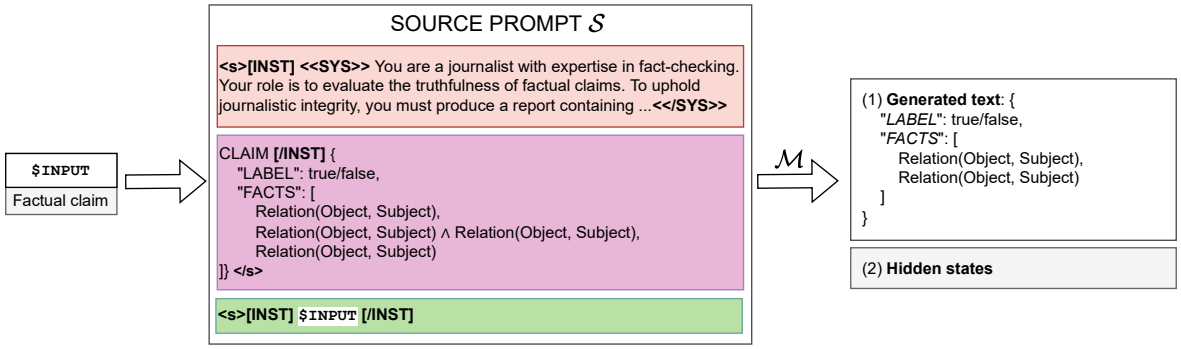



Figure 3.9: Inference of $\mathcal{M}$ on the source prompt $\mathcal{S}$.

facts are represented as a conjunction of ground literals: asserted or negated predicates. For instance, the factual information necessary to evaluate the input claim "Edgar Allan Poe wrote Hamlet" can be represented as:

$$\texttt{AuthorOf(Hamlet, William Shakespeare)} \land \\ \neg\texttt{AuthorOf(Hamlet, Edgar Allan Poe)}$$

where $\land$ and $\neg$ indicate the logical conjunction and negation, respectively. Using ground literals has two advantages: formatting factual knowledge consistently for the subsequent graph generation, and stimulating LLMs to associate and contrast factual information using its logic symbolic knowledge [38]. Preliminary experiments indicate that the use of a simple Subject-Predicate-Object triple representation leads to sub-optimal results, resulting in more isolated and subject-focused information. The complete model instruction ($\mathcal{S}$) is shown in Section B.1.

Afterward, this template is instantiated for each input sentence, and the language model $\mathcal{M}$ processes the resulting prompt $\mathcal{S}$. The model's inference produces a structured textual output (*generated text* in Figure 3.9). Meanwhile, the latent representations (*hidden states*; $H^l$), the neural embeddings across the model's layers, are cached.

### 3.2.3 Activation Patching

This section describes the core stage of the proposed methodology: the probing approach based on activation-level interventions, which is used to extract the semantics encoded in the latent representations of the model's inference previously described.

Here, the matrix $H^l$ is denoted as the latent representations obtained from the $l$-th hidden layer of the model's inference on the source prompt $\mathcal{S}$. Next, I consider a separate inference of the model $\mathcal{M}$ on a different prompt, a target sequence of $m$ tokens

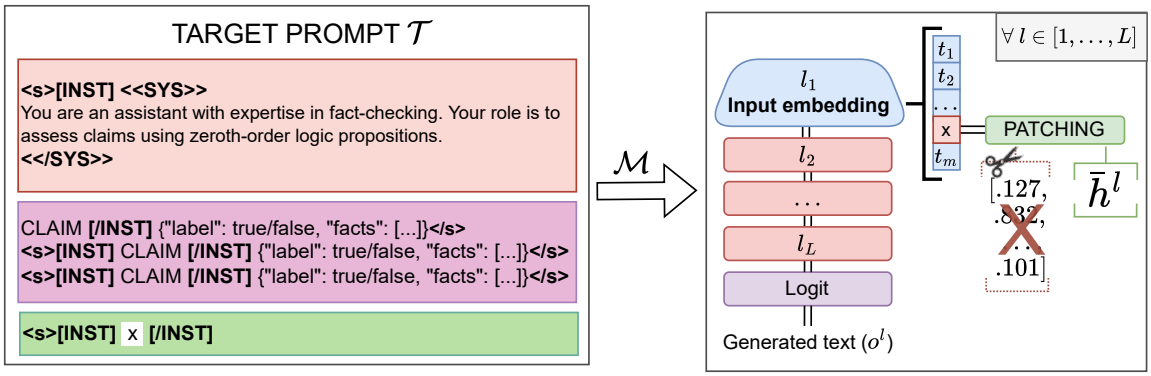



Figure 3.10: Patching operation during the inference of $\mathcal{M}$ on the target prompt $\mathcal{T}$.

$\mathcal{T} = \langle t_1, ..., t_m \rangle$. This prompt mimics the source prompt to generate similar outputs but serves as a propositional probe. It decodes the semantics within the latent representations via activation patching (Section 2.2.1). To perform the patching operation, I include a placeholder token, the character "$x$", within this prompt (green in Figure 3.10). I also augment the in-context examples (from 1 to 3; purple) and reduce the system instruction (pink) to enhance the model's in-context attention during inference. The complete prompt $\mathcal{T}$ is shown in Section B.2.

This patching operation consists of replacing the vector representation of the placeholder token ($\widehat{h}_k^1$, where $k$ is the position of $x$ in $\mathcal{T}$) with a summary of the vector representations of the input sentence from original inference ($\mathcal{M}$ on $\mathcal{S}$), leaving the other latent representations unchanged, and letting $\mathcal{M}$ proceed with the inference, as shown in Figure 3.10. Since activation patching is a vector-level technique, I formally define a matrix-to-vector mapping function:

$$f(H^l, W) \;:\; \mathbb{R}^{n \times d} \mapsto \mathbb{R}^d,$$

parametrized by $W$, that computes a weighted sum of the latent representations of the input part $\mathcal{I} \subset \mathcal{S}$ of the source prompt:

$$f(H^l, W) := \sum_{i=1}^{L^I} w_i h_i^l = \bar{h}^l \tag{3.6}$$

where the input part refers to the input sentence, the factual claim, and $L^I = |\mathcal{I}|$ is its length. The weights $W$ of the weighted sum is determined by performing Part-of-Speech (POS) tagging[11] on the input claim. Nouns and verbs are assigned a weight

[11] spacy.io/models/en

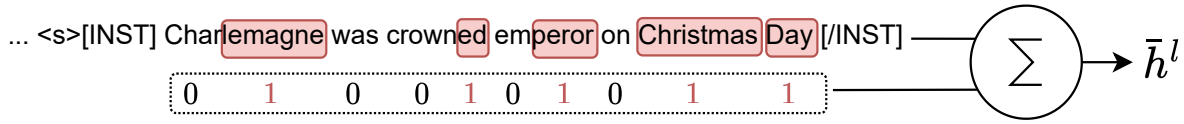



Figure 3.11: Example of the binary weights for the input's tokens $\mathcal{I} \subset \mathcal{S}$. These weights are then used to combine the vector representations of tokens via a weighted sum.

equal to zero to all but their end token which receives a weight of one (Figure 3.11), emphasising sentence's entities and predicates in this summary vector representation. The end tokens of each selected word are considered, following Meng *et al.*'s study [108], which found that the model forms a subject representation at the final token of an entity name. Additionally, empirical experiments showed that using single tokens leads to meaningless texts for punctuation and single-word information for the last input token.

Patching is then applied by replacing $\widehat{h}_k^1$ with $\bar{h}^l$ in the model's input embedding layer, as visually exhibited in Figure 3.10. This model inference, patched with the summary input representation from $l$-th layer, generates a structured text $o^l$, structurally equal to the one from the original model inference ($\mathcal{M}$ on $\mathcal{S}$). By applying this procedure for all values of $l \in [1, \ldots, L]$, structured outputs are produced by using all latent representations of $\mathcal{M}$.

### 3.2.4 Knowledge Graph Generation

Afterwards, I adopt a knowledge graph representation to render the list of all ground literals (factual information) included in the structured output generated by each patched inference ($o^l \in \mathcal{O} \mid \mathcal{M}$ on $\mathcal{T}$; Figure 3.10) and the original inference ($\mathcal{M}$ on $\mathcal{S}$).

The factual information, formatted as first-order logic (FOL), are translated to graph-ready Subject-Predicate-Object triples using simple rewriting rules:

$$(\neg)Relation(object_1, object_2) \rightarrow \langle object_1, (not)relation, object_2 \rangle \quad \text{for binary predicates}$$
$$(\neg)isProperty(object) \rightarrow \langle object, is(not), property \rangle \quad \text{for unary predicates}$$

For example, the ground literal $\neg$AuthorOf(Hamlet, Edgar Allan Poe) is transformed into ⟨Hamlet, notAuthorOf, Edgar Allan Poe⟩.

Afterwards, all SPO triples extracted from $o^l$ are represented as a knowledge graph, as shown in Figure 3.12. Finally, all graphs generated for the different values of $l$ are merged, resulting in a dynamic graph that captures the evolution of factual knowledge across the model's hidden layers.

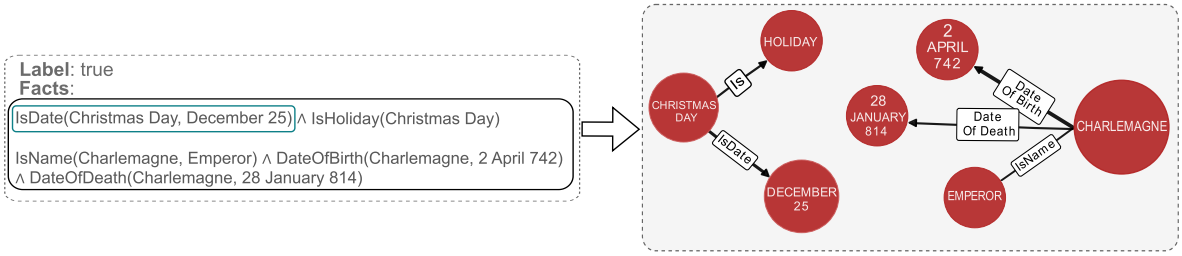



Figure 3.12: Graph generation from the generated ground literals to a connected graph. This process is performed for each $o^l \in \mathcal{O}$ as well as the output of the original inference.

## 3.3 Decoding Neural Embeddings with Vector Symbolic Architectures

This section introduces the framework, grounded in the neurosymbolic paradigm of Vector Symbolic Architectures (VSAs; Section 2.2.2), to extract semantic concepts encoded in neural embeddings of LLMs. Further details of this novel methodology are provided in Chapter C and within the published paper [22]. For example, Section C.5 further discuss the broader applicability of the proposed approach.

### 3.3.1 Problem Statement

Given a language model $\mathcal{M}$ with $L$ hidden layers and a prompt $\mathcal{S}$ consisting of a textual input of $T$ tokens $= \langle s_1, \ldots, s_T \rangle$. The focus is on inspecting the latent representations of the language model, represented as a three-dimensional tensor $\mathbf{H}$: neural embeddings with latent dimension $D$, as produced at every hidden layer $l \in [1, \ldots, L]$ for each token during the inference of $\mathcal{M}$ on $\mathcal{S}$:

$$\mathbf{H} = \text{HiddenStates}(\mathcal{M}, \mathcal{S}) \in \mathbb{R}^{L \times T \times D} \tag{3.7}$$

The objective is to extract the semantics encoded in $\mathbf{H}$ by mapping neural embeddings to human-understandable symbols, as illustrated in Figure 2.1. This work introduces a novel probing methodology grounded in the neurosymbolic paradigm of VSAs, aimed at generalising the extraction of structured knowledge from the vector space of LLMs. Specifically, it extracts the atomic semantic concepts encoded in $\mathbf{H}$:

$$\begin{aligned} \mathbf{H} &= \mathcal{M}(\text{“}\texttt{Denmark is to krone as Mexico is to}\text{”}) \\ f(\mathbf{H}) &= \{\phi_{\text{mexico}}, \phi_{\text{peso}}\} \end{aligned} \tag{3.8}$$

**The Approach.** Section 3.3.2 introduces an analogy-style corpus used as a controlled testbed for evaluating the proposed methodology on textual inputs that require diverse reasoning strategies. Section 3.3.3 then explains how training examples are built using hypervector algebra. Next, the three-stage pipeline is described as follows:

1. processing neural embeddings (Section 3.3.4, $F$ in Figure 3.13);

2. mapping embedding into a controlled proxy space via a neural VSA encoder, producing VSA encodings (Section 3.3.5, $T$);

3. extracting concepts from these VSA-based encodings (Section 3.3.6, $I$).

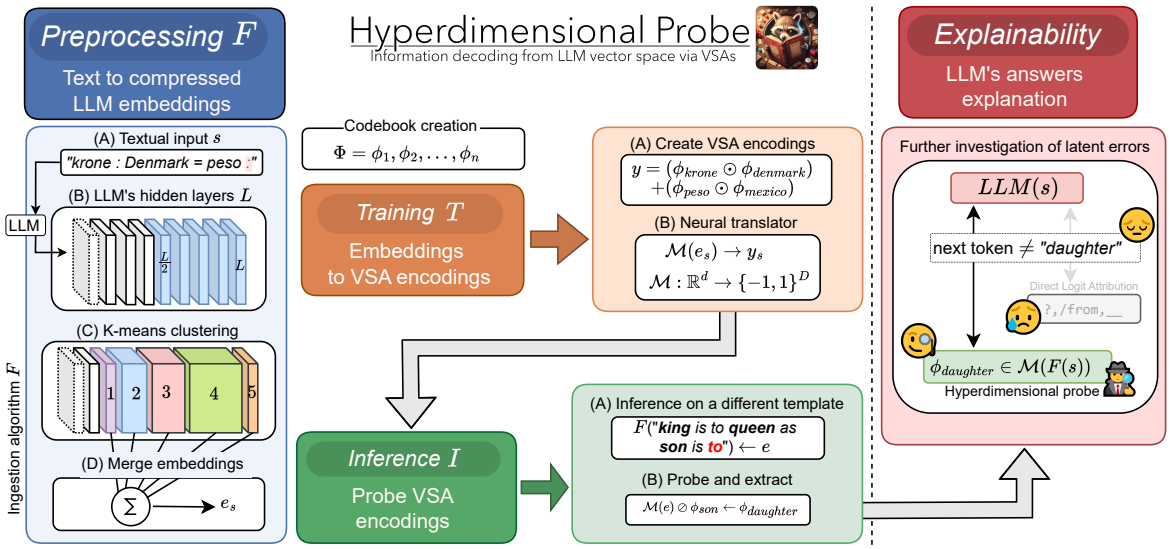



Figure 3.13: **Framework for probing neural embeddings using VSAs**. The model's internal state is initially compressed using dimensionality-reduction steps on LLM embeddings ($F$, **blue**). Next, a neural VSA encoder is trained to map these compressed embeddings into a bounded proxy space: VSA encodings ($T$, **orange**). Concept extraction is then performed by querying this proxy space using hypervector algebra ($I$, **green**). Lastly, extracted concepts ultimately enables deeper analysis on model's internal state and its outputs (**red**).

### 3.3.2 Synthetic Corpus

First, a textual dataset of analogy-style completion tasks is created to test the core components of the proposed approach under controlled and interpretable conditions. Using analogy-style tasks guides LLMs toward concepts and their relations, testing diverse reasoning from syntactic patterns and key-value associations to abstract inference.

**Knowledge bases.** The synthetic corpus focuses on analogies, textual inputs representing pairs of concepts connected by the same type of factual, syntactic, or semantic relationship. Pairs of analogies are collected from two knowledge bases: Google analogy test set [111], and the Bigger Analogy Test Set (BATS, 65). These span 45

Table 3.9: Knowledge bases adopted to build the synthetic corpus $\mathcal{S}$.

| Dataset | | Domains | | Sample example |
|---|---|---|---|---|
| Google Analogy Test Set | 12 | capital world, currency, plural, ... | 33,812 | `Denmark : krone = Mexico : peso` |
| Bigger Analogy Test Set | 33 | verb+ment, occupation, gender, ... | 73,471 | `queen : king = mother : father` |
| Mathematics | 7 | double, square, division2, ... | 6,816 | `4 : 16 = 5 : 25` |
| | 52 | | 114,099 | |

Table 3.10: Categories of the corpus $\mathcal{S}$ identified by clustering domains with an LLM.

| Category | | Domains | Docs |
|---|---|---|---|
| Morphological Modifiers | 14 | noun+less, adj+ness, ... | 34,308 (30%) |
| Verbal & Grammatical Forms | 13 | past tense, plural, ... | 31,219 (27%) |
| Factual Knowledge | 7 | country capital, occupation, ... | 18,800 (17%) |
| Semantic Relations | 8 | family, genders, ... | 16,831 (15%) |
| Mathematics | 7 | math double, math division5, ... | 6,816 (6%) |
| Semantic Hierarchies | 3 | hypernyms, hyponyms, ... | 6,125 (5%) |
| | 52 | | 114,099 (100%) |

domains across five distinct categories, covering a wide range of factual and linguistic relationships, including analogies related to factual knowledge (e.g., a country's currency), semantic relations (e.g., grammatical gender), and morphological modifiers (e.g., verb+men). I also design mathematical analogies using three-digit integers and basic operations such as doubling, cubing, division, and extraction of roots.

**Textual analogies.** After collecting these pairs, I generate 114,099 distinct textual examples, denoted as $\mathcal{S}$, by combining all possible domain pairings. Each training example is formatted as:

$$\texttt{a}_1 \texttt{ : } \texttt{a}_2 \texttt{ = } \texttt{b}_1 \texttt{ : } \texttt{b}_2 \tag{3.9}$$

where $\texttt{a}_1$ and $\texttt{b}_1$ represent the keys of the two pairs, and $\texttt{a}_2$ and $\texttt{b}_2$ are their corresponding values. For example, `Denmark:krone = Mexico:peso` for the countries currencies, and `queen:king = mother:father` for the grammatical gender. Table 3.9 and Table 3.10 display the synthetic corpus grouped respectively by knowledge base and domains. Some concepts span multiple domains, such as `Australia` links to `Canberra`, `English`, and `Australian`. These overlaps can help mitigate the confounding effect of memorizing key-value pairs. For the experiments in Section 4.3, confounding effects are further controlled by using the same pairs while generating a set of textual inputs ($\bar{\mathcal{S}}$) based on a different verbose template compared to the one presented in Equation 3.9:

$$\texttt{a}_1 \texttt{ is to } \texttt{a}_2 \texttt{ as } \texttt{b}_1 \texttt{ is to } \texttt{b}_2.$$

Conversely, data augmentation techniques are applied to $\mathcal{S}$ for training, such as key-value swapping. This triples the corpus size, generating 395,944 distinct instances.

### 3.3.3 Training Examples

This section describes the process of building VSA-based representations for training inputs. This procedure, illustrated with textual templates from Equation 3.9, generalizes to other templates, such as Question Answering, or tasks (e.g., toxicity detection) since VSAs and hypervector algebra can encode complex structures across diverse inputs.

**Codebook construction.** The codebook defines the set of all input features; in this case, the contextually relevant concept, and is later used to construct and query VSA encodings. In the controlled setting, the codebook $\Phi$ (feature set) is constructed directly using all unique words included in the corpus, such as:

$$\texttt{``mexico''} \rightarrow \phi_{mexico} \in \Phi, \text{ and } \texttt{``krone''} \rightarrow \phi_{krone} \in \Phi.$$

Thus, a matrix $\Phi \in \{-1, 1\}^{n_c \times D}$ is created using the `torch-hd` Python library [73], where $D$ is the VSA dimension and $n_c = 2,996$ is the number of concepts/features. The parameter $D$ is set to 4096, representing an appropriate latent dimensionality given the size of the codebook ($\approx 10^3$), which remains well below the theoretical capacity limit of the MAP-B architecture (Section 2.2.2). The average pairwise cosine similarity of the concepts in the codebook is $0 \pm 0.02$ (see Section C.6), confirming orthogonality.

**VSA encodings.** With well-structured textual inputs, extracting input features and building their VSA-based representation is straightforward. The VSA encoding of each training input $s \in \mathcal{S}$ is constructed by leveraging its constituent words (Equation 3.9) and retrieving their corresponding hypervectors accordingly: $\{\phi_{a_1}, \phi_{a_2}, \phi_{b_1}, \phi_{b_2}\} \subset \Phi$.

Hypervector operations, binding ($\odot$) and bundling ($+$; Section 2.2.2), are then used to encode an input sentence. Since the input template represents two conceptual key–value pairs, I first bind each key to its corresponding value, such as linking each country to its currency in Equation 3.10. The entire sentence is then encoded through bundling, producing a superposed set of contextualized concepts represented as key–value associations. Finally, VSA-based encodings are polarized using the sign function to preserve the bipolarity. In summary, the VSA encoding of an input sentence

is computed as:

$$y_s = (\phi_{\text{key}} \odot \phi_{\text{value}}) + (\phi_{\text{key}} \odot \phi_{\text{value}}) + \ldots = (\phi_{a_1} \odot \phi_{a_2}) + (\phi_{b_1} \odot \phi_{b_2}) \tag{3.10}$$

$$\text{“␣Denmark␣:␣krone␣=␣Mexico␣:␣peso”} \mapsto (\phi_{\text{denmark}} \odot \phi_{\text{krone}}) + (\phi_{\text{mexico}} \odot \phi_{\text{peso}})$$

### 3.3.4 Processing Neural Embeddings

The first stage of the proposed pipeline involves feeding textual inputs to an autoregressive transformer model, followed by obtaining and preprocessing its residual stream ($F$ in Figure 3.13). Using the synthetic corpus, a language model is prompted with an input sentence $s \in \mathcal{S}$, $LLM(s)$. For each textual input, its final word ($b_2$ in Equation 3.10) is removed beforehand since it represents the target concept, the value of the second analogy.

**Caching token embeddings.** The probing goal is to inspect the complete internal state of a language model prior to its textual output, capturing all encoded concepts without assuming their relation to the model's output. To this end, I examine the residual stream in the final token representation, focusing on the middle to last layers. Emerging evidence shows that transformers encode next-token information in the final token due to their autoregressive nature [51, 124], refining it in later residual stream layers [14, 74]. Specifically, for an autoregressive language model with $L$ hidden layers, I consider the embeddings (with size $d$) of the last token (“:”) in the latter half, for all $l \in [L/2, \ldots, L]$, yielding a matrix in $\mathbb{R}^{L/2 \times d}$.

**Algorithm 1:** Ingestion procedure $F$

**Data:** Textual sequence $s \in \mathcal{S}$
**Result:** Compressed model state for its next token prediction
**begin**

```
// Get the residual stream from the language model
```
$\mathbf{H} \leftarrow \text{LLM}(s) \in \mathbb{R}^{L \times T \times d}$ ;

```
// Retain embeddings of the last token from the bottom half of the layers
```
$\mathbf{H}^{\star} \leftarrow \mathbf{H}[L/2 : L, -1]$;

```
// Apply K-Means clustering
```
$\mathbf{C} \leftarrow \text{KMeans}_{K=5}(\mathbf{H}^{\star}) \in \mathbb{R}^{K \times d}$ ;

```
// Sum pooling across the centroids
```
$\mathbf{e}_s \leftarrow \sum_{k=1}^{5} \mathbf{C}_k \in \mathbb{R}^{d}$ ;

**return** $\mathbf{e}_s$

However, considering such a wide range of layers presents a computational challenge, as probing a high-dimensional matrix can significantly increase the computational footprint of the probing pipeline. Further, adjacent layer-wise embeddings are highly correlated (0.9) as found in the analysis [22] concerning the average correlations among model's hidden layers. This likely indicates that embeddings across layers encodes redundant numerical patterns, and thus similar information. Consequently, representation redundancy is defined as the approximate linear dependence among neural embeddings across the hidden layers of LLMs. The analysis on representation redundancy (Section C.8.2) shows that the LLM embedding space is roughly low-rank, with only a few rows (i.e., model's hidden layers), or their combinations, contribute meaningful structure.

**Dimensionality reduction.** To reduce the computational cost of the proposed approach, I shrink the encoder's input dimensionality by introducing two dimensionality-reduction steps: *k-means clustering* [81], and *sum pooling*. Clustering reduces redundancy in representation by grouping dense vector regions in the embedding space and computing centroids, achieving knowledge distillation. To determine the optimal range for $k$, I exploit an approach based on silhouette scores [139]. A trade-off between re-

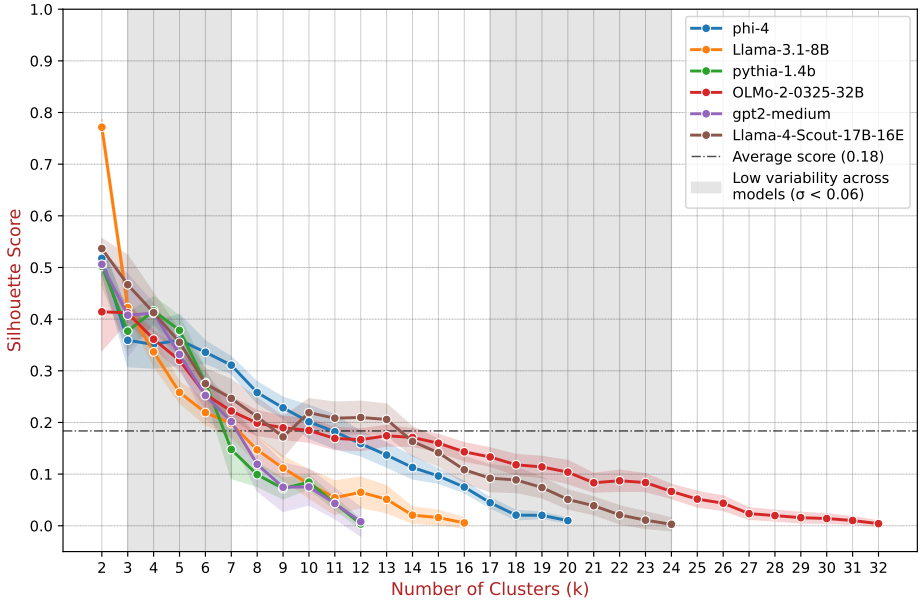



Figure 3.14: **Silhouette scores for varying numbers of clusters**, computed using a random sample of 10,000 textual inputs from $\mathcal{S}$. The six language models have varied layer counts, which results in different maximum possible cluster numbers.

duction, granularity, and model variability emerges with 3–7 clusters (Figure 3.14). The parameter $k$ is set to 5 to maintain the essential data structure while supporting effective dimensionality reduction, as demonstrated in Figure 3.15.

Subsequently, sum pooling is applied, which sums all centroid embeddings to merge the group representatives (k-dimensional matrix) into a single vector, leveraging the additivity property of LLM embeddings demonstrated in prior work [21]. For example, the proposed approach leverages these reduction steps to shrink the probed embedding space of OLMo-2: from $\mathbb{R}^{33\times5120}$ to $\mathbb{R}^{5120}$. Section C.8.3 presents an ablation analysis showing that skipping these two compression steps increases the encoder's trainable parameters tenfold.

In summary, the neural representation of a textual input from a language model is processed through the ingestion procedure $F$, as summarized in Algorithm 1.

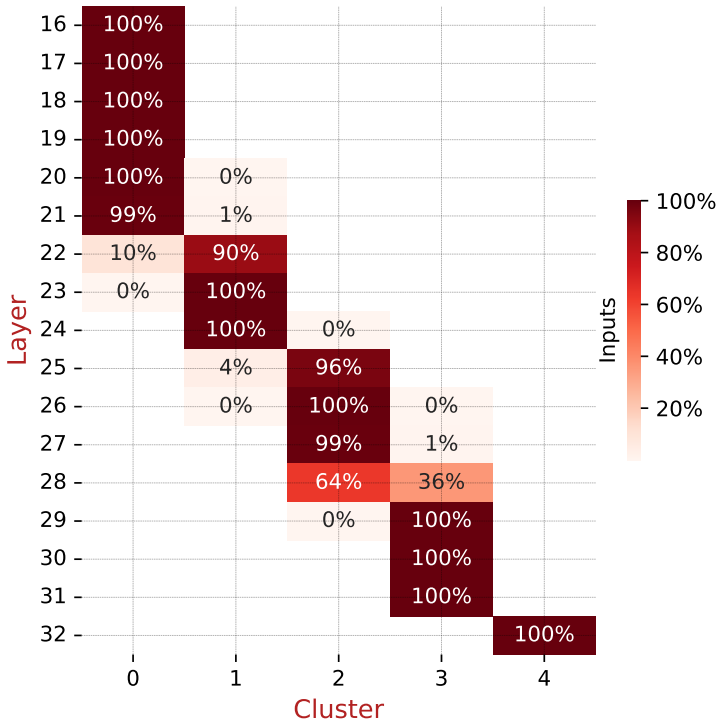



Figure 3.15: **Distribution of cluster assignments using k-means clustering.** Clusters of the hidden layers of Llama3.1-8B found within the ingestion algorithm $F$. It shows the proportions of cluster assignment averaged across all instances.

### 3.3.5 Neural VSA Encoder

Afterwards, a supervised model is trained to map token neural embeddings from an autoregressive transformer into VSA encodings with a known representation ($T$ in Figure 3.13). The supervised regression model $\mathcal{M}$ is defined as a shallow feedforward neural network, which maps the LLM vector space to bipolar hypervectors in MAP-B from VSAs. The model $\mathcal{M}$ is trained on the LLM-VSA dataset generated using the corpus $\mathcal{S}$ (Section 3.3.2), which consists of paired neural embeddings ($e_s$ in Equation 3.11) and their corresponding VSA representations ($y_s$). This encoder infers latent features from the unknown LLM vector space to translate the encoded semantics into VSA representations with explicit and interpretable semantics (Figure 3.16). Formally, I design the *neural VSA encoder* model $\mathcal{M}$ as a three-layer MLP with 55M–71M parameters (depending on the LLM's embedding size $d$), performing a non-linear transformation:

$$\mathcal{M} : \mathbb{R}^d \to \{-1, 1\}^D, \quad e_s \to y_s. \tag{3.11}$$

The encoder's output layer adopts the hyperbolic tangent function (`tanh`) for bipolar outputs, while residual connections are incorporated in the model's architecture to enhance training stability and convergence. The training process minimizes the Binary Cross-Entropy (BCE) error between the bipolar target hypervectors and the predictions. To ensure compatibility with such binary loss function, targets are dynamically converted to binary based on their sign; and predictions are smoothly mapped to the range [0, 1] using the sigmoid function. A Mean Squared Error (MSE) regularization term is added to the loss, with a coefficient of 0.1.[12] The complete model's architecture is reported Section C.2.

**Training details.** The encoder $\mathcal{M}$ was trained for 421 epochs on average via PyTorch Lighting,[13] using early stopping (patient set at 100 epochs) and a batch size of 32. The optimal learning rate was automatically determined using the learning rate finder provided by the aforementioned library, and was approximately set to $3e^{-5}$ on average. *AdamW* was used as optimizer (weight decay of $1e^{-4}$) with a learning rate schedule based on Cosine Annealing with Warm Restarts, starting from the 100th epoch and doubling the restart period thereafter. To adapt the batch size after LR restarts, a Gradient Accumulation Scheduler was used: the effective batch size was doubled at the 110th epoch, quadrupled at the 310th, and increased eightfold at the 410th epoch.

[12] Empirical results showed superior performance compared to other tested coefficients (0.01 - 1).
[13] lightning.ai/docs/pytorch/stable

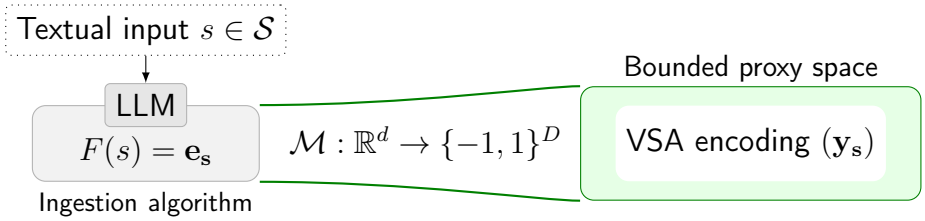



Figure 3.16: The regression model that maps the neural embeddings of a language model ($e_s$) into a controlled vector space ($y_s$) based on VSA-based representations.

During training, the model's outputs are dynamically binarized using the sigmoid function to ensure compatibility with the loss function. This approach demonstrated better empirical performance than linear min-max normalization.

**Language models.** The proposed method is validated on neural embeddings of popular open-weight LLMs, with 355M-109B parameters, available on Hugging Face. This allows experimentation with different embedding sizes and numbers of layers. It is specifically tested using the latest Meta AI's Llama 4 Scout, [109] a multi-modal mixture of 16 experts (MoE), Llama 3.1-8B [67], Microsoft's Phi-4 [2], EleutherAI's Pythia-1.4b [17], AllenAI's OLMo-2-32B [122], and OpenAI's legacy GPT-2-medium [131].

**Performance.** The LLM-VSA dataset uses a random 70-15-15 split of $\mathcal{S}$ for train, validation, and test sets. Since the setting can be interpreted both as a vector-based regression task and a multi-label classification problem,[14] the proposed approach is evaluated using two distinct metrics: cosine similarity and multi-label binary accuracy. For binary accuracy, targets and predictions are binarized based on sign. First, evaluating the cosine similarity between the predicted and target VSA encodings yields a test-set average score of 0.89 (best LLM in Table 3.11, Llama 3.1-8B), indicating strong numerical alignment between our encoder's outputs and the target encodings. Conversely, an average binary accuracy of 0.94 is achieved, indicating robust classification accuracy after polarizing the predictions with the sign function. This means that on average, the VSA encodings produced by the trained model deviated from the targets by only 6% of the vector elements, a negligible error given VSA's large tolerance to noise.

Table 3.11 displays the training performance of the neural VSA encoder $\mathcal{M}$ for all of the six language models considered in this work. All models exhibit consistent performance, layer count has no effect, whereas reducing the embedding dimension is found to be slightly detrimental. Table 3.11 also includes a baseline that compares

[14] VSA encodings can be viewed as vectors with $D$ labels, each assuming one of two possible values.

Table 3.11: Training performance of our neural VSA encoder $\mathcal{M}$ on the test set. Order by model size. `BASELINE` introduces a comparison against a null model with randomly-permuted input embeddings.

| Large Language Model | | | | Cosine similarity | Binary accuracy |
|---|---|---|---|---|---|
| **Name** | **Parameters** | **Embedding dimension** | **Layers from residual stream** | | |
| `BASELINE` | - | 5120 | - | 0.500 | 0.750 |
| Llama 4, Scout, 17B-16E | 109 B | 5120 | $24^{\text{th}}$ to $48^{\text{th}}$ \|25\| | 0.890 | 0.934 |
| OLMo-2 | 32 B | 5120 | $32^{\text{nd}}$ to $64^{\text{th}}$ \|33\| | 0.878 | 0.926 |
| Phi 4 | 14 B | 5120 | $20^{\text{th}}$ to $40^{\text{th}}$ \|21\| | 0.881 | 0.930 |
| Llama 3.1-8B | 8 B | 4096 | $16^{\text{th}}$ to $32^{\text{nd}}$ \|17\| | **0.892** | **0.937** |
| Pythia-1.4b | 1.4 B | 2048 | $12^{\text{th}}$ to $24^{\text{th}}$ \|13\| | 0.861 | 0.916 |
| GPT-2, medium | 355 M | 1024 | $12^{\text{th}}$ to $24^{\text{th}}$ \|13\| | 0.865 | 0.920 |
| | | | **AVERAGE** | 0.878 $\pm$ 0.01 | 0.927 $\pm$ 0.01 |

performance against a null model using randomly permuted input embeddings. Because its test-set predictions collapse toward a nearly constant value near the lower bound of the bipolar domain ($AVG = -0.88 \pm 0.04$), indicating a low-information and weakly discriminative representation, we observe cosine similarities around 0.5. We therefore treat values near 0.5 as the empirical null-signal baseline for this setup.

These results support the initial empirical hypothesis that VSA-based representations (MAP-B) and hypervector algebra can faithfully capture the latent features encoded in neural embeddings of LLMs.

### 3.3.6 Probing VSA Encodings

In the third stage of the proposed pipeline ($I$ in Figure 3.13), the VSA encodings generated by the trained neural VSA encoders $\mathcal{M}$ are inspected. This stage investigates the learned bounded proxy space to extract the semantic concepts embedded in the probed neural embeddings of LLMs. The encoded concepts are retrieved using the *unbinding* operation of hypervector algebra ($\oslash$, Section 2.2.2). This vector operation reverses binding ($\odot$), which originally links a pair's key with its corresponding value, enabling one vector to be extracted from another. Since the generated VSA encoding may encode either no or several concepts, I attempt to extract the target concept ($b_2$) by dynamically testing the unbinding operation with various candidates. This concept-related flexibility represents the novelty and added value of VSA-based probing, allowing one to query the proxy space without prior assumptions on the number of concepts.

Consequently, there is a dichotomy between two scenarios: in the first, no unbinding operations are required when the model encodes a single concept or none; in the second

scenario, when multiple concepts are embedded, the unbinding operation is tested with different concepts to isolate a single one. For example, unbinding a VSA encoding with the concept of `Mexico` and obtaining `Peso` suggests that the probed encoding originally incorporated both the key and value of the target analogy pair:

$$\begin{aligned} \text{LET} \quad & s := \text{“}\texttt{Denmark:krone=Mexico:}\text{”} \mapsto \text{“}\texttt{peso}\text{”} \\ \text{COMPUTE} \quad & y_s = \mathcal{M}(F(s)) \\ \text{QUERY} \quad & y_s \oslash \phi_{\text{mexico}} = \phi_{\text{peso}} + \text{noise} \\ \text{THEN} \quad & y_s \approx (\phi_{\text{mexico}} \odot \phi_{\text{peso}}) \end{aligned} \tag{3.12}$$

When probing a generated VSA encoding ($y_s$ in Equation 3.12), I pick in-context concepts ($\phi_{\text{denmark}}$, $\phi_{\text{krone}}$, and $\phi_{\text{mexico}}$), and their combinations, as candidates for unbinding. The best candidate was chosen by benchmarking the resulting concept after unbinding, against the in-context and target concepts through cosine similarity with a threshold for concept detection equal to 0.1. Empirical tests show relevant concepts exceeded this low threshold, while noisy ones remain below, likely due to VSA's noise tolerance. If no concepts were detected, unbinding was skipped.

In the experimental findings reported in Section 4.3, 80% of unbinding operations, averaged across all models, relied on the key of the target pair. In contrast, no operation was applied in 12% of the cases. Section C.3 reports the proportions of other candidates, highlighting model-specific variations, while also presenting results from unbinding with concept candidates unrelated to the input.

# Chapter 4

# Experimental Results

This chapter introduces the experimental results of the three approaches described in Chapter 3. Section 4.1 illustrates the findings of extracting structured information from companies' non-financial reports using LLMs [20]. This is followed by experimental findings of extracting semantics from neural embeddings of Large Language Models: from tracing the dynamics of factual knowledge encoded in LLMs (Section 4.2; 21) to extracting semantic concepts from neural embeddings with VSAs (Section 4.3; 22).

## 4.1 Deriving Structured Insights from Companies' Non-financial Reports

Section 4.1.1 reports the network statistics computed at the node level from the three bipartite graphs populated with the structured information extracted from companies' non-financial documents. Next, Section 4.1.2 provides a diversity analysis exploring how ESG topics are disclosed among companies and sectors, while Section 4.1.3 illustrates company similarities according to jointly disclosed ESG actions. Afterwards, Section 4.1.4 investigates whether companies' report-based similarities are associated with similarities in other company characteristics. Section 4.1.5 then presents an interpretability analysis of companies' ESG rating scores by identifying the factual aspects that have the greatest impact on these scores.

Finally, Section 4.1.6 discusses the methodological implications of the novel approach, as well as the practical implications of the data-driven findings for companies' non-financial reporting (Section 2.1).

### 4.1.1 Network Statistics of Bipartite Graphs

This presents several network statistics of the three bipartite graphs (Table 4.1): $\mathbb{B}_{\text{cocat}}$, $\mathbb{B}_{\text{catpred}}$, and $\mathbb{B}_{\text{coact}}$. Further statistics and extensive tables for all three graphs are shown in the Supplementary Material document of the published paper [20].

**The ESG movement encompasses many socially responsible issues.** The proposed data-driven methodology unveils that the one hundred and twenty-four companies disclosed 542 distinct ESG topics/categories in their sustainability reports. The company-category bipartite graph $\mathbb{B}_{\text{cocat}}$ has an average degree distribution equal to almost 11%, making this graph relatively connected. There are however some mainstream ESG categories: `Climate Risk Management, Supply Chain, Energy` and `Corporate Governance` are connected to, and thus disclosed by, almost all the considered companies (degree $> 92\%$, Table 4.2). Conversely, the list of the least disclosed categories is rather surprising: `Market Responsibility`, `Anti-Discrimination` and `LGBTQ+ Inclusion` are connected to, and thus disclosed by, less than 5% of all the companies (Table 4.2).

**ESG topics are addressed from several perspectives, with some frequent ones.** The average degree centrality of the category-predicate bipartite graph $\mathbb{B}_{\text{catpred}}$ is less than 1%. However, there are some predominant predicate nodes that are associated with more than ninety ESG categories (degree $\geq 16.6\%$) such as `Commitment and involvement with` (113 categories), `Advisor support for` (102), `Partnership with` (97), and `Establishment of` (94). These prominent nodes interestingly exhibit high closeness centrality ($> 88\%$) in contrast to their relatively low degree centrality values, indicating that common category nodes indirectly connect them.

**Common actions are the exception.** The company-action bipartite graph $\mathbb{B}_{\text{coact}}$ connects the company nodes to almost twenty thousand different ESG-related actions (19,574) disclosed in companies' sustainability reports. However, there are a

| Bipartite graph | | | Partition A | Partition B | Edges | Density (%) |
|---|---|---|---|---|---|---|
| $\mathbb{B}_{\text{catpred}}$ | category (`cat`) | predicate (`pred`) | 542 | 4,864 | 19,574 | 0.7 |
| $\mathbb{B}_{\text{cocat}}$ | company | category (`cat`) | 124 | 542 | 7,455 | 10.8 |
| $\mathbb{B}_{\text{coact}}$ | company | action (`act`) | 124 | 19,574 | 43,169 | 1.7 |

Table 4.1: **Descriptive statistics of the bipartite graphs built from the complete KG**. The category–predicate and company–action graphs have lower density than the company–category graph.

Table 4.2: Sample of the graph metrics of all the 542 category nodes from $\mathbb{B}_{\text{cocat}}$.

| **Category node** (cat) | **Degree** (%) | **Closeness** (%) | **Betweenness** (%) |
|---|---|---|---|
| Supply Chain | 95.3 | 98.9 | 2.3 |
| Climate Risk Management | 94.5 | 98.5 | 2.3 |
| Corporate Governance | 93.0 | 98.1 | 2.1 |
| Community and Society | 91.4 | 98.1 | 2.1 |
| Philanthropy | 88.3 | 96.8 | 2.0 |
| Packaging | 50.0 | 77.7 | 0.6 |
| Human Resources | 39.8 | 74.2 | 0.4 |
| LGBTQ+ Inclusion | 4.7 | 55.4 | 0.0 |
| Anti-Discrimination | 2.3 | 52.1 | 0.0 |
| Marketing Responsibly | 0.8 | 47.8 | 0.0 |

few prominent actions disclosed by the majority of the considered companies: `AIR EMISSION: Reduction of` (degree of, and thus connected to, the 70% of the companies), `ENERGY:Reduction of` (61%), `PHILANTHROPY:Donation by` (60%) and `CLIMATE RISK MANAGEMENT:Assessment of` (56%).

### 4.1.2 Diversity Analysis of ESG Category Disclosure

This section presents an analysis of differences in how ESG topics are disclosed and addressed in the companies' sustainability reports.

**Companies approach ESG topics from plenty of perspectives, especially those generic and vague.** The predicates associated with each ESG category vary significantly with an average Shannon's entropy of 1.5 nats. This broad action variety is predominant in generic and umbrella ESG topics such as `Corporate Governance`, `Human Rights`, and `Supply Chain` (Table 4.3). For example, when addressing `Product Safety`, companies approach it from different perspectives, ranging from developments (2.3%) to regulatory compliance (2.3%) and assessments (4.4%, Table 4.3). However, a notable correlation ($corr = 0.84$) exists between the entropy of categories and the number of companies disclosing information about a particular ESG category. This suggests differences in how each company approaches these ESG topics.

**Topics across sectors versus within a sector.** A proportion of almost 12% among all the 542 ESG categories is disclosed across all company sectors, encompassing various umbrella aspects such as `Climate Risk Management`, `Supply Chain` and `Business Ethics`. However, it is noteworthy that certain sectors emphasise specific topics more

Table 4.3: **Entropy of a sample of ESG categories.** The rightmost column displays the three most frequent predicates for each category, alongside the predicate-wise percentage within companies' category-related disclosures.

| **ESG category** (`cat`) | **Entropy** (nats) | **Companies** (%) | **Category predicates** (`pred`) |
|---|---|---|---|
| Supply Chain | 5.72 | 95 | Partnership with (3.7%), Assessment of (2.4%), Engagement in (2.2%) |
| Corporate Governance | 5.62 | 94 | Establishment of (4.3%), Commitment and involvement with (3.7%), Overseeing (2.9%) |
| Human Rights | 5.62 | 90 | Commitment and involvement with (5.1%), Respect for (2.6%), Assessment of (2.4%) |
| Product Safety | 5.27 | 69 | Assessment of (4.4%), Compliance with (2.3%), Development and implementation of (2.3%) |
| Food Waste | 2.44 | 6 | Commitment and involvement with (14.3%), Reduction of (14.3%), Development of (7.1%) |
| Anti-Slavery Practices | 1.61 | 4 | Undertaking of (20%), Integration of (20%), Required training (20%) |

than others (Table 4.4). For example, `Packaging` is more emphasised in Consumer Staples companies, such as PepsiCo (18% of all the company triples), Coca-Cola (9%), Monster Beverage (6%), and Tesco (5%). This category accounts for 4.9% of all generated company triples within this sector as exhibited in Table 4.4. Another example is `Water` which is more stressed by companies operating in water-intensive sectors such as Consumer Staples (6.6%, Table 4.4, e.g., Coca-Cola) and Materials (4.9%, e.g., DuPont) rather than Financials (0.5%, e.g., Goldman Sachs).

**Comparison with ESG materiality at the sector and industry level** I then compare data-driven evidence of sector-specific ESG topics with sector-level ESG materiality matrix (Section 2.1) identified by Khan *et al.* [91]. For explanatory purposes,

Table 4.4: **Sample of ESG categories disclosed by companies in their sustainability reports.** This table exhibits the category coverage through different percentages concerning: the company triples including a category; the companies reporting a category, aggregated also by sector. The rightmost column reports category-related triples aggregated by the sector of the companies.

| **ESG category** (`cat`) | **Triples** (%) | **Companies** (%) | **Sectors** (%) | **Triples per company sector** (%) |
|---|---|---|---|---|
| Corporate Governance | 6.7 | 94 | 100 | Industrials (7.8%), Financials (7.5%), Health Care (7.4%) |
| Air Emissions | 3.7 | 92 | 100 | Energy (6.9%), Materials (4%), Industrials (3.8%) |
| Water | 3.1 | 85 | 100 | Consumer Staples (6.6%), Materials (4.9%), Energy (4.4%) |
| Green Buildings | 1.5 | 88 | 100 | Real Estate (3.5%), Consumer Discretionary (1.8%), Information Technology (1.6%) |
| Packaging | 0.9 | 49 | 100 | Consumer Staples (4.9%), Consumer Discretionary (1.5%), Information Technology (0.9%) |
| Business Ethics | 0.3 | 61 | 100 | Health Care (0.7%), Utilities (0.5%), Industrials (0.4%) |

I investigate the relevant issues identified for the sector of `Financials`; as is one of the common sectors between the aforementioned work [91] and the company sample of this work. There are thirteen financial companies in the sample of this work: nine commercial banks (e.g., Deutsche Bank), two credit service companies (e.g., Mastercard), one insurance company (Assicurazioni Generali) and one asset management company (3i Group). The data-driven findings uncover that the companies' sustainability reports address the ten issues identified as relevant for this sector with different importance. Financial companies, for example, extensively disclosed actions concerning: *environmental, and social impacts on core assets and operations* (`Climate Risk Management`: 7% of all the sector triples), *Diversity and Inclusion* (`Financial Inclusion`: 6.1% and `Board Diversity`: 5.2%) and *Lifecyle impacts of products and services* (`Product Sustainability`: 4.1%). Conversely, companies' non-financial disclosures neglect to address some relevant issues encompassing: *Access and affordability* (`Access to Basic Services`: 0.8%, `Accessibility`: 0.3%, and `Access to Information`: 0.1%), *Fair marketing and advertising* (`Marketing` and `Advertising`: 0.1%), and *Business ethics and transparency of payments* (`Business Ethics`: 0.2% and `Anti-Money Laundering`:

0.1%). Section A.6 presents the comprehensive comparison for all ten issues that are identified for the financial sector.

Subsequently, UniCredit, a financial company in the Commercial Banks sector, is selected as an illustrative example to compare the data-driven findings with the disclosure topics specified in the SASB standards for its industry. The SASB standard identifies six important topics that should be disclosed by commercial banks.[1] The company weighted these topics differently in its non-financial report. Firstly, it outweighed relevant industry-relevant issues such as: *Financial Inclusion & Capacity Building* (`Financial Inclusion`: 4.6%) and *Data Security* (`Data Privacy`: 2.4%). Conversely, it neglected to report much information on relevant issues, such as *Financed Emissions* (`Product Design`: 0.4%) and *Incorporation of Environmental, Social, and Governance Factors in Credit Analysis* (`Environmental Risk Assessment`: 0.4%).

**Different sectors employ tailored actions to address ESG topics.** Although there are a few widely disclosed actions, the same action is disclosed, on average, by less than 2% of the considered companies. Whereas, only 15% of the company sectors are, on average, engaged in the same action. This unveils different priorities and a variety of approaches among companies and sectors. For example, the `Assessment of` aspects concerning `Climate Risk Management` are more emphasised by Real Estate companies which manage assets vulnerable to climate risks, such as Park Hotels Resorts (2% of all the company triples) and Sun Communities (1%). Conversely, Materials companies such as United States Steel (1%), Yamana Gold (1%) and Aluminum Corporation of China (1%), emphasise instead the `Commitment and involvement` concerning `Employee Safety`, a worker-related topic.

### 4.1.3 Company Similarities based on Disclosed Actions

This section analyses company similarities according to jointly disclosed actions using the Jaccard-based similarity approach described in Section 3.1.7.

**Companies within the same sectors tend to undertake similar actions.** For example, as depicted in Figure 4.1 and outlined in Table 4.5, five companies among the top ten similar companies of `Deutsche Bank` are banks too: `Royal Bank of Canada` (action similarity equal to 7%), `Banco Santander` (6%) and `UniCredit` (6%). Notably, `Visa` and `Mastercard`, both operating in the Credit Services industry, form a distinct

[1] sasb.org/standards/materiality-finder/find/?company[0]=IT0005239360

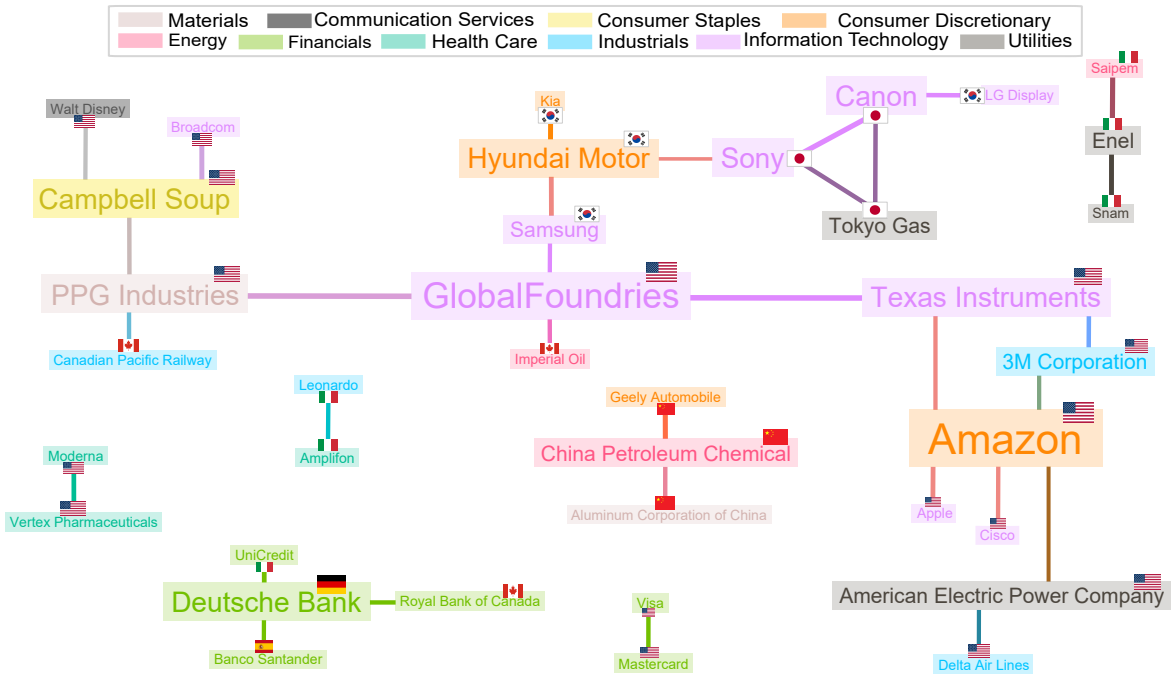



Figure 4.1: **A network diagram linking companies that report similar actions, determined by the Jaccard similarity.** It exhibits only connections between companies with a similarity equal to or greater than 6%. Node colour corresponds to distinct sectors, and node size is proportional to their connectivity. Some connections are noteworthy for linking companies within the same sectors or geographical regions.

cluster (Figure 4.1). Comparably, action similarities emerge in companies operating in the Health Care sector: Moderna, Vertex Pharmaceuticals and AstraZeneca (Table 4.5).

**Companies from the same geographical region tend to perform similar actions.** For example, as can be visually noted in Figure 4.1, 80% of the ten most similar companies of `Sony` (Japan, Eastern Asia) are companies from the same geographical region: 40% from Japan and 40% from South Korea. Similarly, `Geely Automobile` (China, Eastern Asia) has 70% of its ten most similar companies from the same region too: 40% from China and 30% from South Korea. On the west side, there are six European companies in the ten most similar companies of `Enel` (Italy, Southern Europe) with Italian companies representing 40% of the total.

### 4.1.4 Correlation Analysis among Company Similarities

This analysis answers the research question of whether company similarities, computed as jointly-disclosed actions, are associated with similarities in other company character-

Table 4.5: **A company sample with the top three most reported actions and the most similar companies for each.** Company similarities are assessed by computing the Jaccard similarity on the companies' disclosed action set.

| Company | Most reported actions | Most similar companies |
|---|---|---|
| Sony | PACKAGING: Reduction of (x7)<br>PHILANTHROPY: Advisory support for (x6)<br>CORPORATE GOVERNANCE: Establishment of (x6) | Canon (7%), Tokyo Gas (6%), Hyundai Motor (6%), Toshiba (6%), Kia (6%) |
| Deutsche Bank | ENERGY: Reduction of (x4)<br>BIODIVERSITY: Promotion of (x4)<br>CORPORATE GOVERNANCE: Establishment of (x4) | Royal Bank of Canada (7%), Banco Santander (6%), UniCredit (6%) |
| Global-Foundries | WATER: Use of (x6)<br>AIR EMISSIONS: Reduction of (x7)<br>PHILANTHROPY: Donation by (x7) | Texas Instruments (7%), PPG Industries (7%), Samsung (6%), Visa (6%) |
| Geely Automobile | PHILANTHROPY: Participation in (x5)<br>SUPPLY CHAIN: Establishment of (x5)<br>CORPORATE GOVERNANCE: Development and implementation of (x5) | China Petroleum Chemical (7%), Baidu (6%), LG Display (6%), Alibaba (5%), Korean Air Lines (6%) |
| Saudi Aramco | AIR EMISSIONS: Reduction of (x4)<br>BIODIVERSITY: Protection of (x4)<br>ENERGY: Investment in (x4) | Tokyo Gas (5%), Royal Dutch Shell (5%), Yamana Gold (5%), Visa (5%) |
| Philip Morris | WASTE: Continuous efforts to reduce, reuse, or recycle (x5)<br>BIODIVERSITY: Continuing to set goals and work towards (x4)<br>PHILANTHROPY: Investment in (x3) | Croda (4%), 3M (4%), Coca-Cola (4%), GlobalFoundries (4%) Samsung (4%), United States Steel (4%) |

istics. I perform a bivariate correlation analysis using Kendall's correlation coefficient (Section 3.1.7) for each company with all information available (81% of all the companies). Aggregated results are shown in Figure 4.2 using box plots.

**Companies' similarities in disclosed actions are correlated with their geographical regions.** Action similarities have the highest, yet weak, correlation with the `Region` and `Country` of the company headquarters, with a median correlation coefficient of 0.18 and 0.15. This confirms the empirical findings discussed in the previous section (4.1.3). Moreover, only these two features demonstrate median p-values, resulting from the null hypothesis test of zero monotonic correlation, below the established accepting threshold of 5%, respectively 1% and 2% (see Section A.8).

Taking the previous example companies, `Sony` and `Enel` exhibit a relatively high monotonic correlation between their company similarities in terms of disclosed actions and geographical locations. `Sony` has an action-country similarity correlation equal to 0.22 and an action-region correlation equal to 0.20, while `Enel` exhibits a lower action-country correlation (0.14) and a higher action-region correlation (0.25).

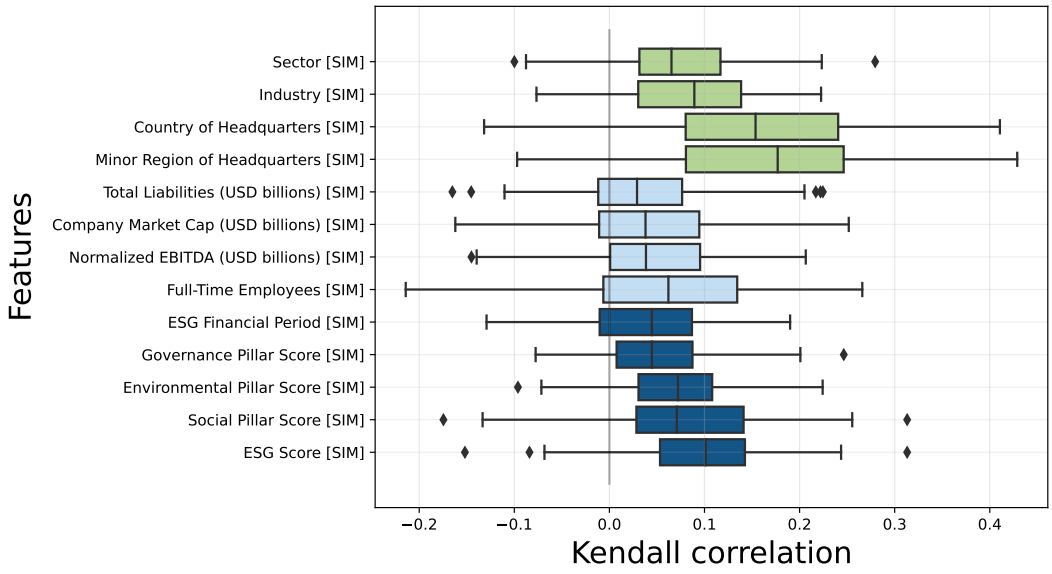



Figure 4.2: **Distributions of pairwise correlations between companies' action similarities and similarities in other company features (rows).** Features are color-grouped according to their type of information. Light-green features categorical company characteristics, while azure and dark blue features represent numerical features concerning companies' financial and ESG information.

**No other statistically significant similarity correlations emerge.** Company similarities in `ESG scores` and `Industries` have a median pairwise correlation with companies' disclosed actions equal to 0.1 and 0.09. Their statistical significance appears however relatively weak due to high median p-values (13% and 15%), which suggest accepting the null hypothesis of zero monotonic correlation.

**ESG scores are only correlated with their underlying components.** After analysing company similarities from the disclosed action perspective, I perform a pairwise monotonic correlation analysis to uncover possible confounding factors for company similarities. Strong monotonic correlations appear between similarities in companies' `Region` and `Country` (median correlation equal to 0.7) as well as between companies' `ESG score` and their `Social` (0.5) and `Environmental Pillar scores` (0.4). No other relevant correlations emerge for ESG scores or other company information.

### 4.1.5 Interpretability of ESG Scores

Finally, the interpretability of ESG rating scores is examined using first-order linear regression and the SHAP framework (Section 3.1.7). Specifically, I assess how different

factual and corporate factors influence the scores, leveraging features derived from companies' reports (e.g., the most disclosed ESG topics) and company-specific information.

**Social-related actions and company transparency have a significant impact on ESG scores.** On average, the aspects with the greatest impact on ESG rating scores are the proportions of disclosed actions related to `Human Rights` and `Employee Development`, with mean SHAP values of 2.6 and 2.7 (Figure 4.3). High disclosing percentages of the former (colour scale in Figure 4.3) positively impact ESG scores, while the latter has the effect of hurting them. High disclosing percentages in actions related to `Philanthropy` (mean SHAP value of 1.1) and `Energy` (0.7) also hurt companies' scores. Similar average magnitude, but an opposite effect, disclosing several actions related to `Waste` (0.8) or `Supply Chain` (0.7) has a positive impact. Furthermore, high variety in the disclosed ESG topics (`Category Entropy`) positively affects ESG scores with a mean SHAP value of 2.1 (Figure 4.3). Sharing a similar average magnitude (1.9), being founded earlier, represented by an older `Incorporation Year`, positively impacts a company's score. This is also validated numerically by a negative Kendall correlation equal to -0.22 (p-value of 0.5%) and visually by Figure 4.4 which groups the companies' ESG scores by their decade of incorporation. The median ESG score of the fifty-three companies funded in the 20$^{\text{th}}$ century is 77.2, whereas the thirty-one companies established in the current century exhibit a lower median score of 68.6. Notably, the

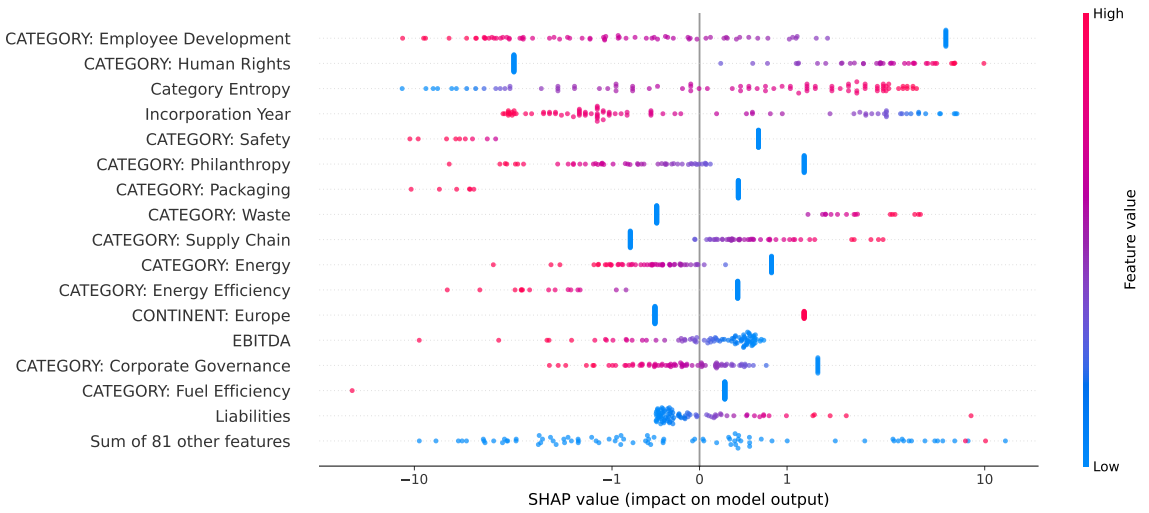



Figure 4.3: **The top sixteen features impacting the most the inference of ESG score**; ordered according to their median shape value. The x-axis represents the degree of a positive and negative impact on model output. Each dot represents a company instance and colours represent the company values of the standardised feature.

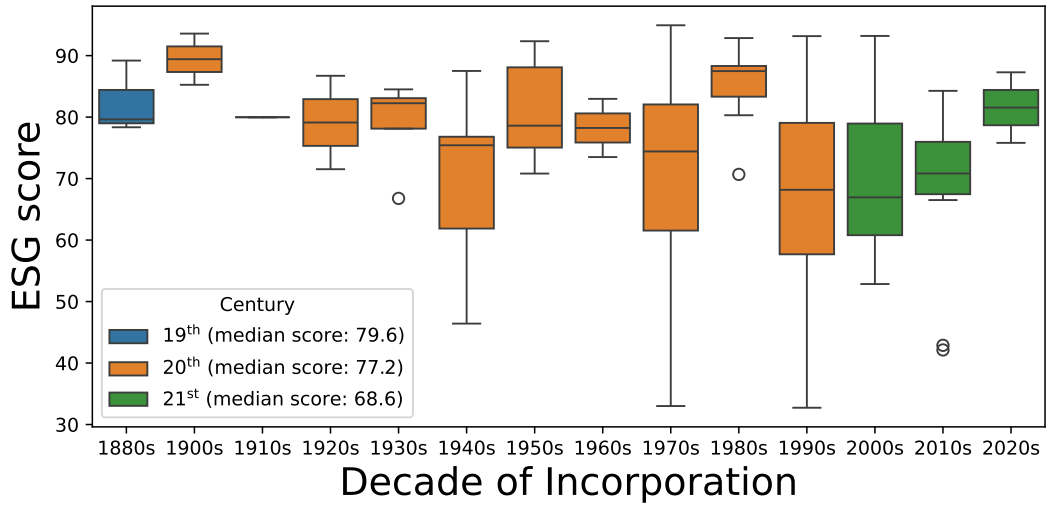



Figure 4.4: **Companies' ESG scores grouped by the decade of incorporation**. The ESG scores refer to the fiscal year of the companies' sustainability reports considered in our work, almost all from the 2020s. Colours map the century of the decades and legend entries also exhibit the median ESG score for each.

three companies founded in the 19$^{th}$ century exhibit the highest median ESG score equal to 79.6. Further noteworthy factors positively impacting ESG scores are being a European company (`CONTINENT:Europe`; mean SHAP value of 0.7; Figure 4.3) and exhibiting a high level of `Liabilities` (0.5). In contrast, high annual earnings (`EBITDA`, 0.6) have a slight negative impact on ESG scores.

The positive influence of being European is further validated by grouping ESG scores by company region: the twenty-six companies from Europe exhibit the highest average ESG score equal to 82, the forty-nine American companies have an average ESG score of 69.7, while the average score of the twelve Asian companies is equal to 67.

**Local interpretability reveals company-specific impacting factors.** Moving from global to local interpretability, the company `Sony` is chosen as an example to investigate the most impacting factors for its ESG score (Figure 4.5). The `Incorporation Year` and `Category Entropy` features, respectively well below (standardized value of -1.03, corresponding to 1946) and above (0.87; 3.6 nats) the average of all companies, have a positive impact on the ESG score. In addition, disclosing several actions related to `Human Rights` (0.57, 4.4% of all its extracted actions) and `Waste` (1.31, 3.4%) has a positive impact. In contrast, disclosing fewer `Energy`-related actions than the average (-0.98, not among its top ten disclosed topics) positively affects its scores. Lastly, being an Asian company (`CONTINENT:Asia` and `CONTINENT:Europe`) slightly hurts its score.

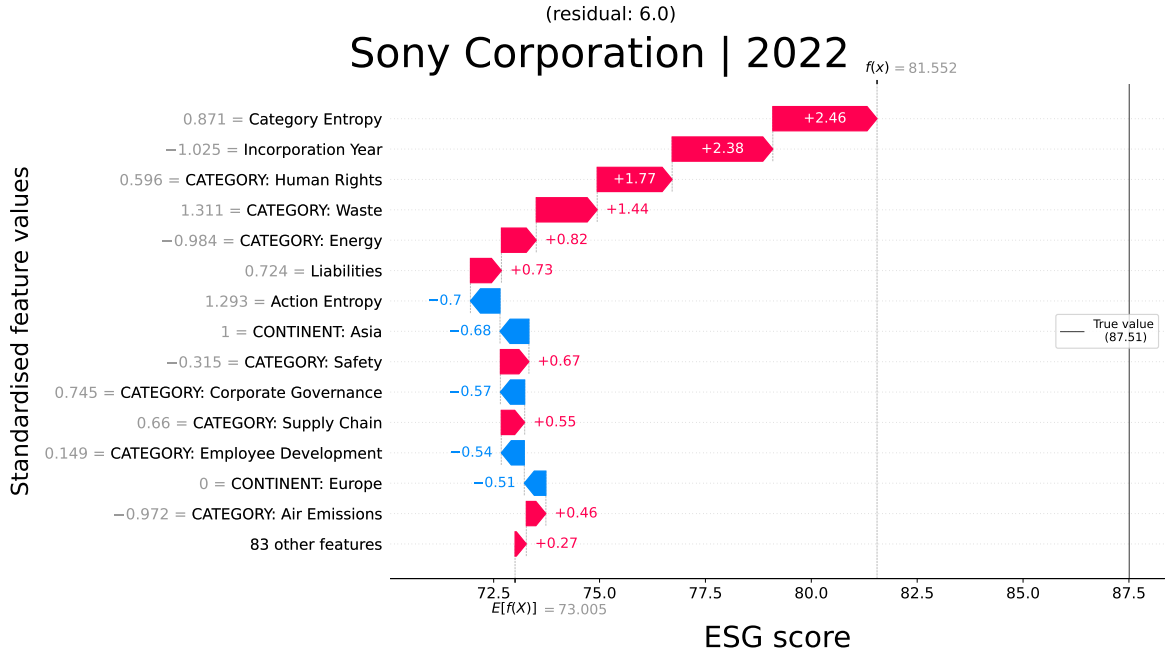



Figure 4.5: **Example of explanations of company's ESG score using report-based and factual-based predictors.** The category-based features are extracted from the 2022 sustainability report of the Japanese company `Sony`, while further company information from the same fiscal year. In addition, the actual ESG score and the model error (residual) are shown.

**Region-based interpretability analysis reveals common patterns.** Lastly, a more granular analysis is conducted by investigating the interpretability of ESG scores concerning a cluster of companies. Following the aforementioned example, Asian companies are considered: five Chinese companies (e.g., Alibaba; Table 4.6), five Japanese companies (e.g., Sony), Aramco (Saudi Arabia) and Greely Automobile (Hong Kong). The `Incorporation Year` strongly affects their ESG scores (Section A.9.2), with an average SHAP value of 2.7, in line with the global interpretation. This is further validated by a strong negative Kendall correlation of -0.61 (p-value of 0.7%). All companies established in the 20$^{\text{th}}$ century, such as Toshiba (1904, score of 93.6), Toyota (1937, score of 84.5) and Geely Automobile (1946, score of 75.4), exhibit ESG scores above 66. Whereas, those established in the current century have all lower scores such as Baidu (2000, score of 53.5), China Evergrande (2006, score of 52.8) and Aramco (2018, score of 42.9). From the geographical point of view, being a Chinese company (`COUNTRY:China`) hurts ESG scores with an average SHAP value of 0.6. This is further emphasised by the observation that all the Chinese companies consistently have ESG scores below 62.5, whereas both Hong Kong-based and Japanese companies consistently

Table 4.6: The twelve Asian companies considered for the region-based interpretability analysis. The companies are ordered by ESG rating score, the rightmost column.

| Company | Sector | Country | Year of Incorporation | Fiscal Year | ESG score |
|---|---|---|---|---|---|
| Toshiba | Industrials | Japan | 1904 | 2021 | 93.58 |
| Sony | Information Technology | | 1946 | 2022 | 87.51 |
| Toyota Motor | Consumer Discretionary | | 1937 | 2020 | 84.51 |
| Tokyo Gas | Utilities | | 1885 | 2021 | 78.35 |
| Geely Automobile | Consumer Discretionary | Hong Kong | 1946 | 2022 | 75.42 |
| Daikin Industries | Industrials | Japan | 1934 | 2022 | 66.78 |
| China Petroleum Chemical Corporation | Energy | China | 2000 | 2021 | 62.49 |
| Aluminum Corporation of China | Materials | | 2001 | 2019 | 57.14 |
| Baidu | Communication Services | | 2000 | 2020 | 53.55 |
| China Evergrande | Real Estate | | 2006 | 2020 | 52.85 |
| Alibaba | Consumer Discretionary | | 1999 | 2022 | 48.33 |
| Saudi Aramco | Energy | Saudi Arabia | 2018 | 2021 | 42.88 |

display higher scores (Section A.9.1). Furthermore, this analysis confirms that disclosing several actions related to `Human Rights` (e.g., Toshiba and Tokyo Gas) and `Waste` (e.g., Toyota and Sony) positively impact ESG scores, whereas their absence negatively affects them (e.g., Baidu and Daikin Industries).

### 4.1.6 Practical and Methodological Implications of the Findings

**The ESG phenomenon has blurred boundaries and includes plenty of topics.** Concerning the ESG topics disclosed by companies, this data-driven methodology extracts 542 distinct ESG topics from companies' non-financial reports, representing an eight-times greater set of 64 topics originally included in the ESG categorization ( section 3.1.2) exploited as a semantic reference in this work. Firstly, this unveils the broad scope of the ESG phenomenon involving socially responsible topics ranging from `Waste Management` and `Supply Chain` to `Employee Safety` and `Tax Compliance`. Secondly, this highlights the presence of widely disclosed topics, such as `Supply Chain`, and sector-focused topics such as `Packaging` and `Water` for the Consumer Staples sector. The diversity analysis reported in Section 4.1.2 confirms this sector-based importance for certain topics. For instance, `Water` is more stressed by water-intensive companies, while `Packaging` is more emphasised by companies producing consumer staples (Table 4.4). This data-driven insight is also validated by ongoing discussions in non-financial reporting literature (Section 2.1) concerning ESG Industry Materiality. I also assess companies' disclosures relative to the sustainability issues expected in their reports, by comparing them with sector-level and industry-level ESG materiality matrices; highlighting differences in topic coverage as outlined in Section 4.1.2.

**Exogenous factors might influence companies' non-financial disclosures.** The findings reported in Section 4.1.3 emphasise company similarities based on their sectors, confirming indirectly a relatively high presence of common strategies among companies within the same sector. However, the most impacting factor in grouping companies based on their non-financial disclosures is their geographical region as shown in Section 4.1.3 and Section 4.1.4. This represents an interesting finding from this data-driven study, validating the ongoing discussions in the literature concerning the influence of the company's geographical origins on their non-financial disclosures (Section 2.1). For example, previous studies have analogously unveiled the impact of exogenous factors on companies' non-financial disclosures, encompassing region-specific regulations [26] to the political, labour, and cultural environment of the company's nation [11, 46, 163].

**Exogenous factors can influence the quality of companies' non-financial disclosures, indirectly impacting their ESG performance assessments.** The interpretability analysis of companies' ESG rating scores (Section 4.1.5) highlights that the company's disclosures impact these scores more than other financial aspects or company characteristics. Disclosing several ESG topics (`category entropy`) positively affects ESG scores, whereas fewer disclosed ESG topics hurt scores. This data-driven insight accordingly confirms that transparency on non-financial information rewards companies' ESG assessment [46, 95]. The analysis of Section 4.1.5 also confirms the negligible impact of governance-related topics towards ESG scores in comparison to social- and environmental-related topics such as `Human Rights`, `Energy` and `Waste`. This analysis also uncovers that disclosing many actions related to `Employee Development` and `Energy` negatively impact ESG scores. One hypothesis is that their high presence in a company's sustainability report might reveal a poor coverage of other important ESG topics [134]. For example, in the local interpretation of Sony's ESG score, a low disclosing percentage of `Energy`-related actions and a high disclosing percentage of `Waste`-related actions positively impact its score.

Another noteworthy finding of this interpretability analysis on companies' ESG rating scores is the region's impact on those scores: being a European company has a positive impact, whereas being a Chinese company hurts them (Section 4.1.5). The Chinese penalization factor might be reinforced by the fact that all the considered Chinese companies were funded between 1999 and 2006, likely due to the remarkable economic development of this region starting in the 2000s, and thus associated with the negative impact of being relatively young companies (Section 4.1.5). Indeed, our interpretability analysis also uncovers a beneficial effect associated with earlier incorporation

years. However, further investigation might be necessary to exclude spurious correlations. The impact of the company's region on ESG scores is however coherent with the region-based disclosing similarity previously highlighted and validated by ongoing discussions in the ESG literature (Section 2.1) regarding an European bias. Further validation of the positive impact of being a European can be found by delving into the environmental performance of European companies in our sample: they demonstrate the highest average environmental score of 81.8, surpassing Asian companies with an average score of 69 and American companies with the lowest average score of 65.6.

On the other hand, previous studies (Section 2.1) also suggest that companies' ESG assessments might indirectly suffer from biases based on company size. Section 4.1.5 unveils negligible evidence of this company size bias: company's market capitalization has no impact on interpreting ESG scores, while a greater number of full-time employees has a positive, yet marginal, impact on scores (average SHAP value of 0.24). In addition, the pairwise correlations of company similarities concerning these two features and similarities in ESG scores (Section 4.1.4) are not statistically significant: a monotonic correlation of 0.1 for the number of employees (p-value of 71%), and a zero correlation for the market capitalization (p-value of 78%).

**Leveraging ESG categorization and LLMs improves retrieval of comprehensive, data-driven insights.** The flexibility and generative abilities of LLMs allow the proposed approach to highlight, and overcome some limitations of current ESG-related resources, such as the ESG categorization proposed by Berg *et al.* [15].

This categorization extrapolates a concise set of Environmental, Social, and Governance topics by categorising several ESG-related indicators shared among ESG rating providers. It thus captures the scope of the ESG phenomenon from the perspective of rating agencies, offering a viewpoint different from the actual disclosures of companies. In contrast, the proposed approach enables a comprehensive examination of non-financial disclosures from the companies' perspective, revealing significant differences between the ESG topics they report and those emphasized by rating agencies. It extracts a set of ESG topics/categories disclosed in companies' reports that is eight times larger than the original list of categories of this classification (542 versus 64). For instance, the proposed approach unveils `Education` as a pivotal ESG topic disclosed by more than two-thirds of the selected companies. This topic is not explicitly included in the categorization, although it could be framed within three of its categories: *Access to Basic Services, Human rights* (article n°26) or *Philanthropy*. `Circular Economy` is a further example; it might fall under the categorization categories of *Waste* or *Resource*

*Efficiency*. Whereas, `Air Quality` could be framed within *Green Buildings* or *Health and Safety*. The aforementioned ESG categorization [15] consequently encompasses critical topics hidden within vague categories or potentially overlooks them altogether, resulting in a reduction of substantial significance in subsequent analysis.

The proposed approach addresses this narrow viewpoint by exploiting the generative abilities of LLMs in conjunction with the cutting-edge prompting technique of In-Context Learning and the Retrieval-Augmented Generation paradigm. This allows for the joint simulation of a supervised text classifier's outputs, where the labels correspond to the ESG categories, while also enabling the semantic generalization of those labels. The ESG categorization is consequently leveraged as semantic guidance by the LLM, helping itself to extract more suitable topics while keeping the domain and semantics of the original ESG categorization. This semantic generalization can also diminish the reliance on specific ESG categorizations when classifying sentences since the ESG categories are exploited as semantic references rather than fixed labels. Nevertheless, this could also lead to the undesirable phenomenon of over-specialization which was tackled using semantic clustering, a process outlined in Section 3.1.6.

In the data preparation stage, the ESG categorization [15] was also adopted to filter the sentences of the companies' non-financial reports through text embeddings (Section 3.1.3). This semantic-oriented filtering approach allows the proposed method to further move towards a taxonomy-agnostic methodology as filtering is based on semantics rather than single keywords.

**Extracting insights from companies' sustainability reports using LLMs, and graph representations.** Lastly, the proposed approach differs from previous ESG-focused and LLM-based tools for text analytics on non-financial reporting (Section 5.1), such as ChatClimate [155] and ChatReport [116] by employing ICL and RAG to generate a graph representation (KGs). ChatClimate applies the RAG paradigm to enhance ESG-related Question Answering (QA), while ChatReport uses it to operationalize compliance assessment of sustainability reports against the Task Force on Climate-related Financial Disclosures (TCFD) guidelines. By contrast, the proposed method combines structured Information Extraction and graphs to unveil meaningful insights into the actual actions disclosed in companies' sustainability reports.

## 4.2 Tracing the Dynamics of Factual Knowledge in Large Language Models

This section presents the experimental results of the approach described in Section 3.2 through local and global interpretability analyses of the factual knowledge extracted from the latent representations of Large Language Models [21].

After describing the experimental setup in Section 4.2.1, Section 4.2.2 reports the effectiveness of LLMs in claim verification. Next, Section 4.2.3 presents a local analysis of the factual information extracted for three varied inputs. Lastly, Section 4.2.4 shows the global analysis of patterns in the factual knowledge resolution process.

### 4.2.1 Experimental Setup

To analyse the model behaviour in recalling factual knowledge to support or debunk input sentences, a language model is prompted with a collection of factual and common-sense claims sampled from two well-known claim verification datasets: *FEVER* [151] and *CLIMATE-FEVER* [41]. As in conventional mechanistic interpretability workflows [33], these textual inputs serve to elicit a model behaviour: unfolding the factual information an LLM holds internally when evaluating the input's truthfulness.

Each claim is labelled as `Supported` (True), `Refuted` (False) or `NotEnoughInfo`. For instance, the claim *"Charlemagne was crowned emperor on Christmas Day"* from the FEVER dataset is classified as `Supported`, whereas *"Berlin has a population of 4 million people"* is classified as `Refuted`. These are all factual claims and a language model should rely on specific factual knowledge to evaluate their truthfulness. On the other hand, *CLIMATE-FEVER*, a dataset mimicking the FEVER methodology, compounds to real-world claims regarding climate change [41]. Its claims may require more common-sense reasoning, and elicit subjective dichotomies. For example, the claim *"Global warming is increasing the magnitude and frequency of droughts and floods"* is labelled as `Supported`, whereas *"Renewable energy investment kills jobs"* is classified as `Refuted`.

The two datasets are initially filtered by excluding the claims labelled with *NotEnoughInfo*, avoiding prompting the model with unverifiable sentences, and considering claims neither too short nor too long ($35 \leq |characters| \leq 120$). Afterwards, I randomly sample 1,000 claims from FEVER (*Supported*: 726, *Refuted*: 274) and 400 from CLIMATE-FEVER (*Supported*: 274, *Refuted*: 126).

I adopt the LLaMA 2 language model [152] in its 7-billion instruction-tuned version[2]

[2] huggingface.co/meta-llama/Llama-2-7b-chat-hf

to showcase the proposed framework. Section B.7 shows also the output generated by similar language models: the model's 13-billion version, the newest LLaMA 3 [4] and the 7-billion Falcon model [7].

### 4.2.2 LLM Performance on Claim Verification

**Method.** This section measures the effectiveness of the language model $\mathcal{M}$ to classify the input sentences, accomplishing the task of claim verification. This analysis serves the preliminary purpose of determining whether the language model can perform claim verification with sufficiently reliable accuracy to justify subsequent probing analyses. This analysis uses the model's task performance as a proxy and preliminary indicator that the desired behaviour was successfully elicited: recalling knowledge related the trustfulness of the textual inputs.

The classification metrics consider the binary label within the structured text generated during the inference process (Figure 3.9, $\mathcal{M}$ on $\mathcal{S}$). Table 4.7 shows the metrics grouped for each target binary label: *true* (supported) and *false* (refuted). It also exhibits a self-consistency metric that measures the average consistency of the inferences' binary labels with those generated by the patched inferences ($o^l \in \mathcal{O} \mid \mathcal{M}$ on $\mathcal{T}$).

**Findings.** The language model (LLaMa 2) reaches ROC AUC scores equal to 0.68 and 0.74 for *FEVER* and *CLIMATE-FEVER* respectively. Table 4.7 also reports the F1 score and, the majority baselines for the two datasets. The language model outperforms the baselines on the ROC AUC score by +0.18 on FEVER and +0.24 on CLIMATE-FEVER, indicating a substantially stronger ability to discriminate between true and false claims. This result suggests that the model is leveraging information relevant to the verification task when classifying textual inputs, thereby supporting the use of subsequent probing analyses to investigate the actual knowledge encoded in the model's latent representations.

Table 4.7: **Classification performance of the language model on the task of claim verification.** The `AVG` columns indicate the weighted average performance, where `BASELINE` indicates the majority baselines for the two datasets.

| DATASET | PRECISION | | | RECALL | | | F1 | | | ROC AUC | ACCURACY | SELF-CONSISTENCY | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TRUE | FALSE | AVG | TRUE | FALSE | AVG | TRUE | FALSE | AVG | | | TRUE* | FALSE* |
| BASELINE | 0.726 | 0.000 | 0.527 | **1.000** | 0.000 | **0.726** | **0.841** | 0.000 | **0.611** | 0.500 | **0.726** | - | - |
| FEVER | **0.932** | **0.388** | **0.783** | 0.456 | **0.912** | 0.581 | 0.612 | **0.545** | 0.594 | **0.684** | 0.581 | 0.54 ± 0.2 | 0.14 ± 0.1 |
| BASELINE | 0.685 | 0.000 | 0.469 | **1.000** | 0.000 | 0.685 | **0.813** | 0.000 | 0.557 | 0.500 | 0.685 | - | - |
| CLIMATE-FEVER | **0.873** | **0.552** | **0.772** | 0.704 | **0.722** | **0.710** | 0.780 | **0.625** | **0.731** | **0.739** | **0.710** | 0.49 ± 0.2 | 0.10 ± 0.1 |

However, examining the model's performance reveals a significant imbalance, especially on the *FEVER* dataset. It has high precision and low recall for the claims classified as true, and low precision and high recall for those classified as false (Table 4.7). Intuitively, this may be because falsifying a claim often requires less factual knowledge than is needed to prove it true. On the other hand, the *CLIMATE-FEVER* dataset demonstrates balanced recall performance for both classes, with a precision reduction in the false ones. This suggests the model encounters comparable difficulty in verifying or debunking claims when this mainly depends on common-sense reasoning (Section 4.2.1) while erroneously classifying more claims as false. The confusion matrices are reported Section B.4.

A dichotomy between the two classes is also observed in the self-consistency metric: when the inference generates a true label, on average, almost half of its hidden layers generate the same binary label (Table 4.7), whereas just about ten percent of the layers' labels coincide with the inference prediction for the false label.

### 4.2.3 Local Interpretability

**Method.** This analysis investigates the factual information encoded within the model's latent representations of three varied input claims. Claim A (*"Renewable energy investment kills jobs"*) is from the CLIMATE-FEVER dataset, and requires common-sense reasoning. In contrast, claim B (*"Charlemagne was crowned emperor on Christmas Day"*) and claim C (*"George Lucas founded a visual effects company"*) are from FEVER and require multi-hop reasoning. Figure 4.6 exhibits five snapshots of the dynamic knowledge graphs related to these three inputs, while Section B.3 exhibits all the graph representations generated for claim B.

**Findings.** All claims in Figure 4.6 show word-based factual information in the early layer $L_3$. However, while claims B and C accurately present information about their entities, claim A incorrectly exhibits information about the last in-context example (The Beatles, see Section B.2). The following section further hypothesises that example leakage may occur when the probed latent representation lacks factual knowledge, shifting the model's attention to the provided context. Claim A then unfolds an interesting knowledge evolution: Layer 7 encodes factual knowledge concerning `Energy`, whereas Layer 17 and Layer 19 create a clear dichotomy between entities that are `Job Killer` (e.g., `Coal` and `Nuclear Energy`) and `Job Creator` (i.e., `Energy Efficiency Upgrades`). Although this can be common-sense reasoning, it might unveil biases in

this energy-related dichotomy. Meanwhile, Claim B unveils a swap in the entity representation in the middle layers. Layer 9 correctly encodes factual knowledge concerning the subject entity (`Charlemagne`). Layer 14 then confuses and swaps it with `Charles V` (another former Holy Roman emperor), yet succeeds in multi-hop reasoning by connecting Christmas Day, its actual date, and the emperor's coronation. The third claim exhibits another error related to internal knowledge representation. Layer 9 correctly encodes `George Lucas` as *Founder Of* the company `Industrial Light & Magic`, while treating its acronym (`ILM`) as a different entity. Subsequently, layer 19 associates the information of being a special effects company only with its acronym, while separately connecting its full version to the founder, `George Lucas`. This disjoint association and entity duplication lead to a multi-hop reasoning error in the inference (Figure 4.6). While unifying the company's representations, the model fails to reconnect `George Lucas` as its founder. This knowledge-recalling issue might stem from attention mechanisms or catastrophic forgetting.

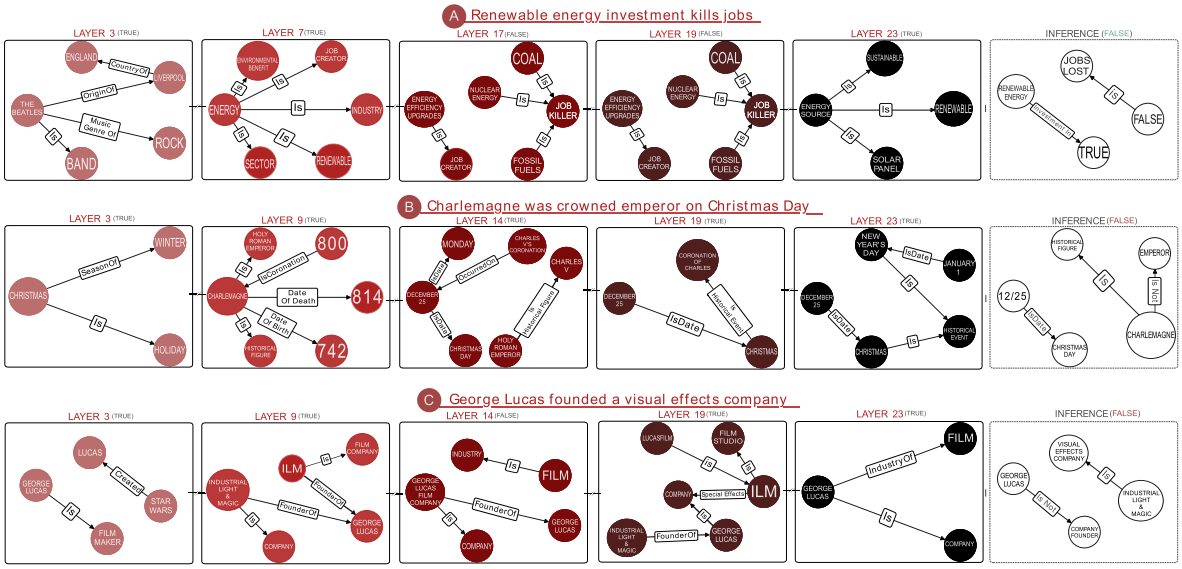



Figure 4.6: **Evolution of the factual knowledge decoded from the latent representations of the language model.** It exhibits five snapshots of the dynamic knowledge graphs created for three input claims. A colour gradient highlights the temporal progression ($L_3 \rightarrow L_{23}$) of the model's inference. On the other hand, the white graph on the rightmost indicates the parsed textual output from the original inference.

### 4.2.4 Global Interpretability

**Method.** This analysis reveals patterns in the evolution of factual knowledge. Behaviour changes across the hidden layers are sought, thereby identifying transaction points in this latent evolution. The Knowledge Graphs (KGs) generated by the latent representation from each hidden layer are examinated by calculating graph-level similarities and identifying cross-layer similarities.

Initially, I compute node embeddings for each dynamic knowledge graph derived from a subset of over five hundred FEVER dataset inferences. I consider input claims with character lengths falling within the first and third quartiles. I adopt the Multi-Scale Attributed Node Embedding method (MUSAE, 140) to generate node embeddings $Z \in \mathbb{R}^{n \times d}$ for each graph [141], where $n$ represents the number of graph nodes and $d$ is the embedding dimension ($d = 4096$). Next, the semantic correspondence between two graphs ($G$ and $G'$) is assessed by using a custom graph similarity measure on the node embeddings: (1) computing pair-wise node similarity, (2) identifying the highest similarity matching for each node of $G$, and (3) averaging these similarities:

$$sim(G, G') = \frac{1}{n_G} \sum_{i=1}^{n_G} \max_{j \in [1, n_{G'}]} sim_{cos}(Z_i(G), Z_j(G')) \tag{4.1}$$

where $n_G$ and $n_{G'}$ are the number of nodes of $G$ and $G'$ respectively, and $sim_{cos}$ is the cosine similarity. To investigate patterns in the evolution of the encoded factual knowledge, I focus on the similarities of the graphs generated using the latent representations of two consecutive layers: $o^l$ and $o^{l-1}$. This process is repeated for all the dynamic knowledge graphs and report aggregated results in Figure 4.7. Afterwards, I

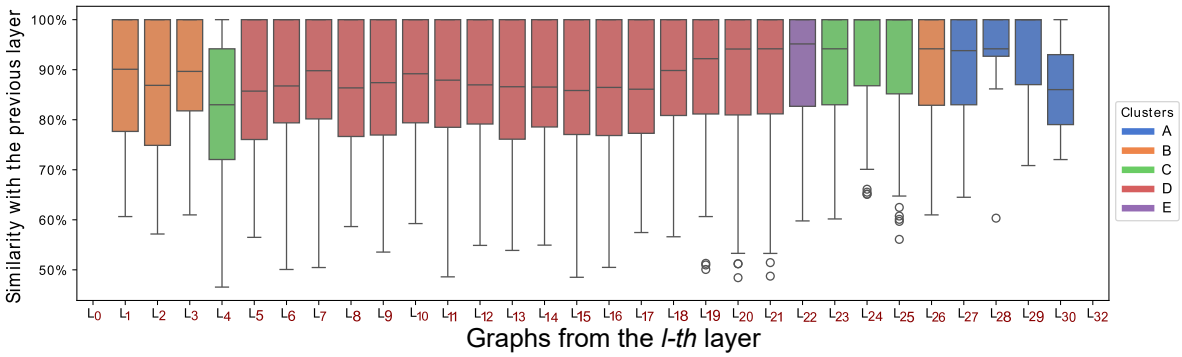



Figure 4.7: **Cosine similarities between graph representations of the factual knowledge extracted from the neural embeddings** of $l$-th and $l$-1-th hidden layer. Colours map the layer clusters discovered by the MeanShift clustering algorithm.

employ the MeanShift clustering algorithm[3] [29] on the similarity data. The bandwidth hyper-parameter, which affects the cluster granularity, is estimated using the 25th percentile of the sample data. This clustering procedure identifies layers exhibiting similar changes in the evolution of their factual knowledge (colours in Figure 4.7), indirectly spotting transaction points.

**Findings.** The transaction points unfolded in Figure 4.7 are validated by examining the type of factual information decoded from the cluster items across several individual inferences. I observed that early layers ($L_1$:$L_3$; cluster B; orange) focus on the syntax and entity resolution process, as already demonstrated by Ghandeharioun *et al.* [64]. I also discovered a slightly different evolution pattern when tokens represent partial or complete words. When tokens correspond to complete words, such as "Germany" and "Robin", the model encodes factual information related to the entity or its semantic context, for instance, `IsCountry(Mexico)` and `SuperHeroOf(Robin, Batman)`. However, in the case of incomplete words, the model accomplishes entity resolution using encoded word-based representations. For example, merging the token representations of "jack" and "eman", the model tries to decode the subject entity as "Jackman", yet the original subject was "Bo*jack* Hors*eman*". Further, these early layers produce valid outputs (structured texts) in only 40% of the considered inferences. In contrast, the latent representations from the 4$^{\text{th}}$ hidden layer (cluster C in Figure 4.7; green) generates valid factual information regarding the subject's semantic context in 75% of the inferences. For instance, `WasQueen(Mary Queen of Scots)` for the original subject "Empress Matilda" or `IsHistoricalFigure(Charles)` for "Charlemagne". Thus, these empirical results indicate that the model retains minimal, if any, factual knowledge in these early layer.

Middle layers ($L_5$:$L_{21}$, cluster D in Figure 4.7; red) exhibit comprehensive factual knowledge concerning the subject entity, with an evolution towards the factual information of the claim. For example, the claim concerning whether Empress Matilda moved to Germany exhibits this dynamics:

$$L_7 = \{\ldots, \texttt{BornIn(Empress Matilda, England)}\}$$
$$L_{15} = \{\ldots, \texttt{DiedIn(Empress Matilda, England)}\}$$
$$L_{18} = \{\ldots, \texttt{MovedTo(Empress Matilda, Germany)}\}$$

These middle layers also exhibit an interesting pattern: whenever the language model

[3] scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift

achieves comprehensive factual knowledge for an entity, its attention moves towards another entity in the sentence (or a semantic entity neighbour), and represents other contextualised information. For example, in another input claim, the language model encodes the factual information concerning "Robin" in layer 4, then concerning "The Joker" between layers 6 and 17, and eventually further factual information concerning "Robin" between layers 18 and 19.

On the other hand, there was a decrease in factual knowledge decoded from the latent representations of the bottom layers ($L_{27}$:$L_{32}$; cluster A in Figure 4.7; blue) and the initial layer. Over 90% of the inferences patched with these layers yield unstructured texts, often containing nonsensical text or references to the final in-context example. These findings suggest that the model's attention may have shifted toward the in-context examples in the final layers. Indeed, this seems supported by the analysis of the attention matrices across layers (see Section B.8), which shows a slight concentration of attention towards groups of tokens within the in-context example tokens in the final layers. Regarding the underlying cause of this attention shift, two speculative hypotheses may be advanced:

1. it may originate from the limited amount of factual knowledge encoded in these latent representations, or

2. the previously mentioned store-and-seek pattern might influence this lack of encoded knowledge.

As a result, one possible hypothesis is that the beginning of this attention shift corresponds to the conclusion of the factual knowledge resolution process, which could be influenced by factors such as the number of contextually relevant entities or the extensiveness of their semantic domain. For example, the beginning of output degradation is differentiated for the three inputs which pertain to increasingly specific domains: an empress's life, a TV show's creator, and a comic book event. It respectively begins at the $24^{\text{th}}$, $22^{\text{nd}}$ and $20^{\text{th}}$ layers. Interestingly, for the broader domain, the model represents factual information about a different empress between the $20^{\text{th}}$ and $23^{\text{rd}}$ layers, validating this transition point.

## 4.3 Extracting Semantic Concepts with Vector Symbolic Architectures

This section presents the experimental results of the probing methodology, grounded in VSAs, described in Section 3.3. It extracts semantic concepts during analogy-style input completion and text generation in a Question Answering (QA) setting [22].

Section 4.3.1 first outlines the experimental setup concerning analogy-style inputs. Next, Section 4.3.2 presents the findings on concept extraction using the VSA-based probing method. This is followed by an experimental comparison with traditional logit-based analyses in Section 4.3.3, which exhibits more superficial probing capabilities. Next, Section 4.3.4 reports an experimental comparison with Sparse Autoencoders (SAEs), highlighting SAE-based probing produces noisier interpretability analyses in bounded conceptual frameworks. Lastly, Section 4.3.5 applies the entire methodology in a different experimental setting, changing the scope of investigation from input competition to text generation.

### 4.3.1 Experimental Setup

**Data.** The trained neural VSA encoders $\mathcal{M}$ (Section 3.3.5) are tested on the set of textual input formatted using the verbose template (Section 3.3.2). Thus, the proposed methodology is validated using inputs with syntactic structures that differ from those seen during the training stage ($\bar{\mathcal{S}} \neq \mathcal{S}$). Therefore, information decoding is performed from the vector representation of a different token, shifting from the colon token of the training examples to the token "`to`" in $\bar{\mathcal{S}}$, as displayed in Figure 4.8. This further mitigates confounding effects caused by probe-induced learning, more pronounced in conventional supervised probes.

**Metrics.** The experimental evaluation has a two-fold objective: evaluating LLMs' performance on the next-token prediction task, and assessing the ability of the probing method to retrieve concepts from their latent representations using *precision@k*. The

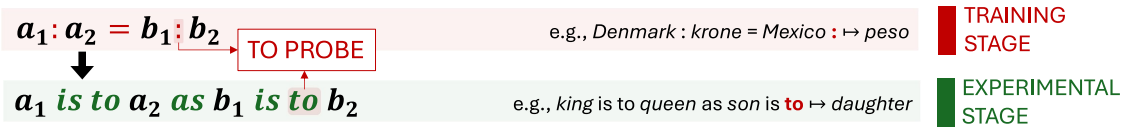



Figure 4.8: The proposed methodology is validated using textual inputs ($\mathcal{S}$, red) with syntactic structures that differ from those seen during our training stage ($\bar{\mathcal{S}}$, green).

task performance of language models are measured via: binary precision on the next-token prediction against the target word; softmax score of the most likely next token and the target one; and rank of the target token on the ordered softmax scores. Specifically, their performance is measured using the most likely next token (*next-token@1*) and the top-5 tokens (*next-token@5*), capturing uncertainty in the model's softmax distribution. On the other hand, to evaluate the performance of the VSA-based probing approach, the binary precision of retrieving the target VSA concept from LLMs' latent representations is measured using *probing@1*, and *probing@5*.

### 4.3.2 Concept Extraction

Figure 4.9 reports VSA-based target concept retrieval, LLM downstream performance, and logit-based comparison. Table 4.10 shows the same results with variability and two validation tests. Table 4.8 presents the final results of VSA probing, with each row showing the unique concept sets produced by the extraction procedure of Section 3.3.6. Each set includes the concepts used for unbinding and those identified afterward; $\phi_{\text{mexico}} \mapsto$ `Key`, and $\phi_{\text{peso}} \mapsto$ `Target` for the example within Equation 3.12.

**High variability in LLMs' next-token predictions.** In an unexpected contrast, the largest model evaluated (109B; Llama 4, Scout) exhibited the lowest precision@1 in the next-token prediction task (8%, Figure 4.9), even underperforming legacy GPT-2. However, its next-token@5 was comparable to others (still the lowest), but ranked

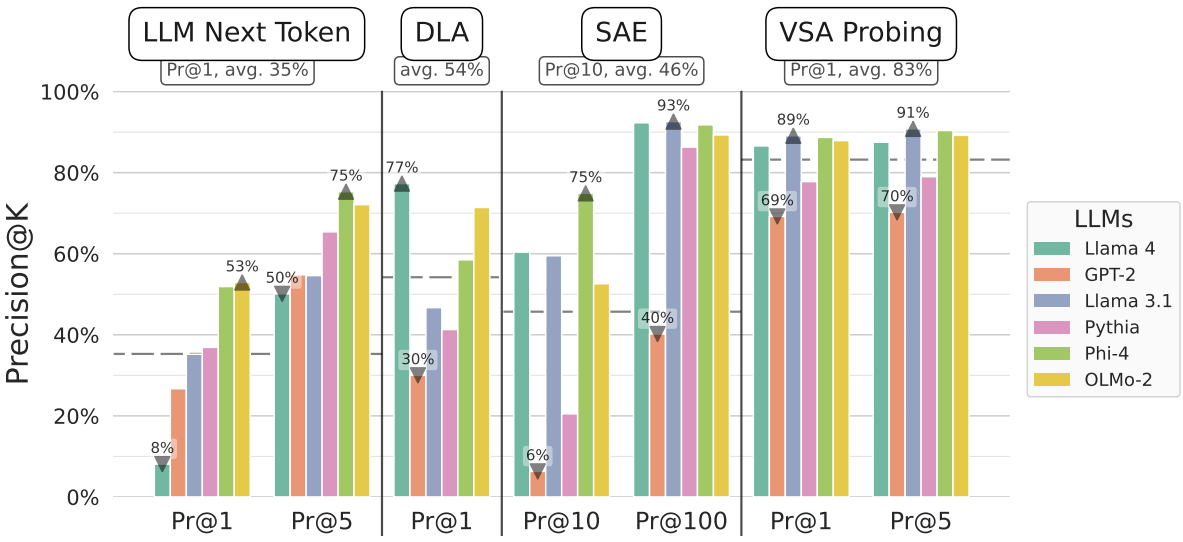



Figure 4.9: LM performance on analogy completion with targets (left) and our decoding method's ability to extract targets from LLM embeddings (right), with logit-based and SAE-based results shown in the middle.

Table 4.8: **Concepts extracted by hyperdimensional probe.** `Key`|`Target` indicates extraction of the key ($b_1$) and value ($b_2$) of the target analogy; `Key` for only $b_1$. `Example` refers to $a_1$ and $a_2$, the in-context example's concepts; `Context`|`Target` for all concepts. `Out-of-context` indicates concepts unrelated to input, `Key Values` refer to analogy keys originating from a different domain. `NONE` means no concepts. *Our probe captures target-aligned combinations in roughly 80% of cases.*

| **Extracted Concepts** (%) | **GPT-2** | **Pythia** | **Llama 3.1** | **Phi-4** | **OLMo-2** | **Llama4, Scout** | **AVERAGE** |
|---|---|---|---|---|---|---|---|
| Key \| **Target** | 60.0 | 66.9 | **85.4** | 84.8 | 80.1 | 79.0 | 76.0 ± 9.4 |
| `NONE` | **21.9** | 16.7 | 6.9 | 7.6 | 8.5 | 11.5 | 12.2 ± 5.4 |
| Key | 4.5 | **6.1** | 0.6 | 1.0 | 1.8 | 3.4 | 2.9 ± 2.0 |
| Example | **5.8** | 2.4 | 1.5 | 1.3 | 2.0 | 0.8 | 2.3 ± 1.6 |
| Context \| **Target** | 1.1 | 1.9 | 1.5 | 1.2 | **4.4** | 2.4 | 2.1 ± 1.1 |
| Key \| Key Values | 1.3 | 1.4 | **1.8** | 1.5 | 1.1 | 0.8 | 1.3 ± 0.3 |
| Out-of-context | **1.6** | 1.2 | 0.5 | 0.7 | 0.8 | 1.1 | 1.0 ± 0.4 |
| Example Value \| Key Values | **1.5** | 1.3 | 0.3 | 0.0 | 0.2 | 0.1 | 0.6 ± 0.6 |
| Key Values \| **Target** | **0.4** | 0.1 | 0.3 | **0.4** | 0.1 | 0.1 | 0.2 ± 0.1 |
| **Target** | 0.1 | 0.1 | 0.1 | **0.2** | 0.1 | 0.1 | 0.1 ± 0.0 |

among the best in probing@1. *Strong probing performance suggests that the final state encodes the target concept, but the model often fails to output it.* This might be caused by exogenous, such as prompt design and endogenous factors, including model's tokenization. Figure 4.10 displays the experimental metrics for a single language model.

**Variability introduced by prompt design and tokenization.** Llama 4 most frequently generated a white space token for our corpus $\bar{\mathcal{S}}$, accounting for 76% of its outputs, considerably higher than the 8% average observed in the other models (30% for Llama 3.1). Its next most common tokens were: `?` (9%), `what` (6%) and `x` (0.7%). The target token had a median rank of 5, with its SoftMax score trailing the top-1 token by a median difference of 0.85 (Figure 4.10), which starkly contrasts other models with 0.05. Thus, the model confidently predicted a space, with the target word often within its top five predictions. These insights, and the strong performance of our hyperdimensinal probe (probing@1 = 87%), suggest issues in handling the syntactical structure of our corpus rather than lack of analogical reasoning. Possibly influenced by its tokenizer (see space-token frequency in the other Llama), which emphasizes prompt engineering importance and variability caused by models' tokenizers. This may be further worsened by the model's multimodality and the complexity of its MoE architecture.

This finding highlights the variability introduced by prompt design and tokenization, which may have a greater impact on standard logit-based methods such as Direct Logit Attribution (DLA), as these rely sorely on the model's output vocabulary.

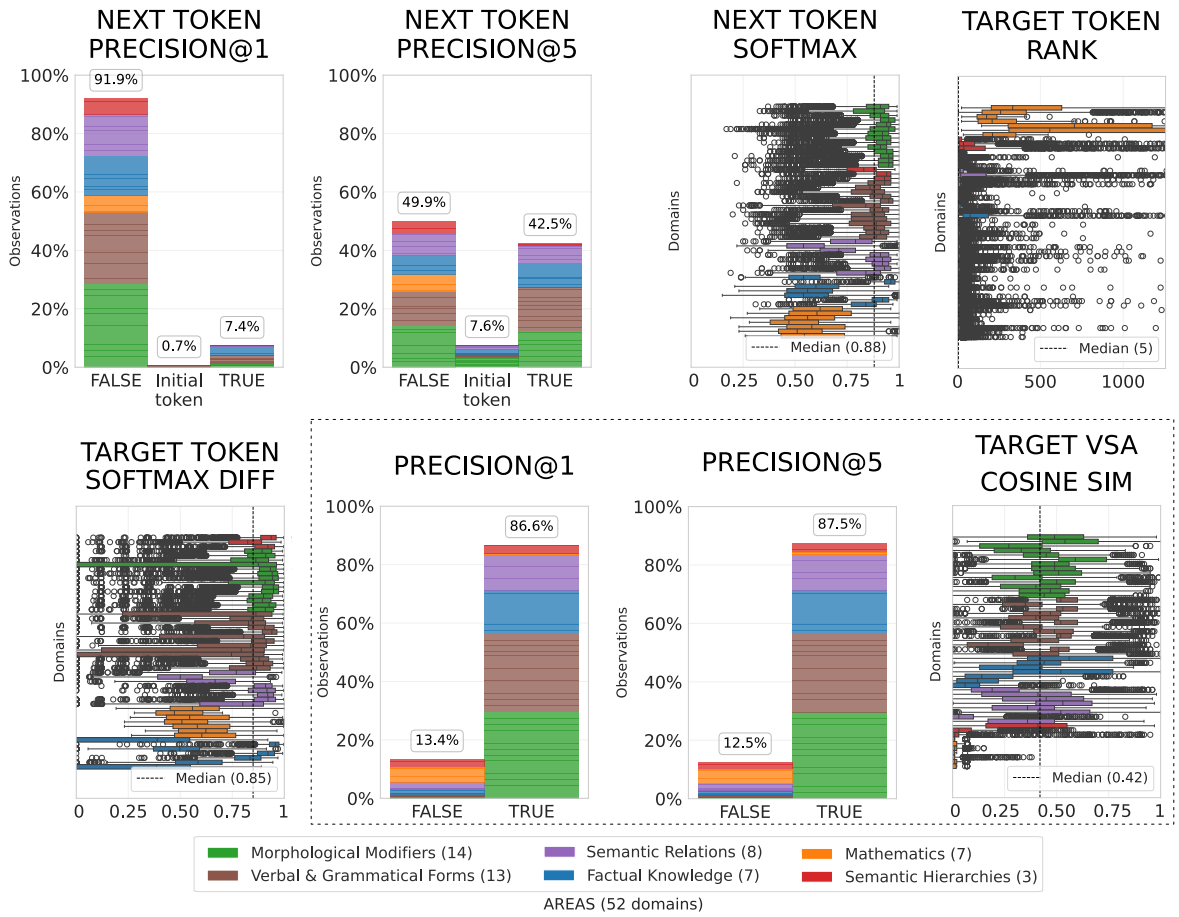



Figure 4.10: **Overview of the experimental metrics for Llama 4 Scout: highlighting the performance of both the language model and the neural VSA encoder.** The colour palette associates each instance with its corresponding semantic area, facilitating visualization of performance across different input types. The first row presents metrics related to the LLM's input completion, including the precision and confidence (softmax score) of next-token predictions, as well as the rank of the correct targets within model's ordered predictions. *Precision@k* then evaluates the encoder's ability to retrieve targets from model's neural embeddings; it is shown as a categorical variable to better illustrate distribution across input types. Finally, the extracted VSA representation is assessed by computing the cosine similarity between the extracted concept and the target concept from the codebook $\Phi$.

**VSA probing exposes varying conceptual richness.** Regarding the actual concepts extracted by our hyperdimensional probe, It achieves an average probing@1 across all models equal to 83% (right side of Figure 4.9), extracting the target concept with its key for most cases (60% for GPT-2; 85% for Llama 3.1; Table 4.8). Notably, GPT-2 shows the highest proportion of cases with no concepts extracted (22%) followed by cases extracting only the in-context example concepts (6%), and ranks second in extracting only the target concept keys (5%). Thus, its internal states largely fall back on input-related concepts, *reflecting limited understanding needed to complete the analogy-style input correctly.* On the other hand, OLMo-2 has the highest proportion of instances in which our probing approach retrieves the target concept alongside all in-context concepts (Context | Target, 4%; Table 4.8), *indicating its richer representation in its final state for both the input context and next word.* This latent richness is then reflected in its performance on next-token prediction, achieving the highest next token@1 equal to 48% (Figure 4.9). In cases where the target word was not among the top five predictions of Llama 4, nearly 50% of the instances (Figure 4.10; 28% for OLMo-2), The probing method successfully extracted the target concept and its associated key in 70% of instances, while no concept was retrieved in 18% of cases (26% for OLMo-2). Although the first insight supports previous observations, the absence of extracted concepts merits a more granular analysis across analogy categories.

**Distribution of instances containing no concepts.** Probe performance is then examined across different input types, namely analogy categories. Success and failure cases are defined by the presence or absence of concepts extracted by VSA probing. Table 4.9 displays the distribution of instances with no concept extracted grouped by input category, such as analogy related to factual knowledge or morphological modifiers.

The probe most frequently encounters conceptually-empty representations in mathematical analogies (89%; Table 4.9), followed by semantic hierarchies (35%). Factual

Table 4.9: **Subset of inputs with no extracted concepts grouped by input type**. OLMo-2 shows richer embeddings than Llama 4, with lower proportions of instances with conceptually-blank representations for most of the areas. Llama 4 slightly outperform OLMo2 in mathematical and grammatical analogies.

| Area | Llama 4 (docs, %) | OLMo2 (docs, %) | AVG (docs, %) | Sample | Domain |
|---|---|---|---|---|---|
| Mathematics | **87.8** | 91.1 | 89.5 | `80 is to 160 as 98 is to` | math double |
| Semantic Hierarchies | 38.8 | **30.8** | 34.8 | `limousine is to car as monorail is to` | hyponyms |
| Semantic Relations | 10.0 | **3.9** | 7.0 | `Croatia is to Croatian as Switzerland is to` | nationality adjective |
| Factual Knowledge | 5.5 | **5.1** | 5.3 | `euclid is to Greek as galilei is to` | name nationality |
| Verbal & Grammatical Forms | **1.4** | 2.1 | 1.9 | `seeing is to saw as describing is to` | past tense |
| Morphological Modifiers | 1.1 | **0.8** | 1.0 | `agree is to agreement as excite is to` | verb+ment |

and morphological analogies show much lower rates, at 5.3% and 1.0%, respectively. While the comparison reported in Table 4.9 show variability across models, these findings nonetheless points to a common pattern of representational blankness, likely driven by the type of reasoning involved:

1. **Linguistic analogies** yield the lowest rate of missing concept extraction (1–1.8%), suggesting richer LLM representations, likely due to reliance on all concepts to capture *implicit syntactic patterns*.

2. **Factual knowledge** and **semantic relations** show slightly higher but still low blank rates (5.3–7%). Since these analogies rely on *key–value associations*, blanks may reflect missing associations in the model.

3. **Semantic hierarchies** (34.8%) and **mathematical analogies** (89.5%) exhibit the highest blank rates. Both types demand more *abstract reasoning*; however, the substantially higher rate for mathematics likely reflects the relative scarcity of numerical analogical tasks, compared to equation solving or other standard math problems that are more prevalent in the training data.

**Validation strategy.** To assess information decoding of the proposed probe, two control tests are conducted (see Table 4.10) as proposed by Hewitt and Liang [76]:

1. **Permuted Baseline**: the encoder's outputs are compared against a null model by inputting the trained encoder with randomly permuted LLM's neural embeddings;

2. **Unrelated Baseline**: Attempt of extracting concepts that are unrelated to the input using VSA-based probing (Section 3.3.6).

Both baselines achieve fairly low precision probing scores, reinforcing the effectiveness of our method. These results show that: (1) Applying our VSA-based probing using concepts irrelevant to input texts results in meaningless outputs; and (2) corrupted or nonsensical input embeddings also produce poor results.

That said, it is crucial to recognize a fundamental limitation of all probing approaches: by definition, the human-interpretable information encoded in LLM embeddings is not explicitly known. Consequently, no probing method can provide absolute certainty in decoding such information. To address this, we further validated our method by evaluating the trained probes on textual inputs distinct from those used during training, thereby reinforcing the reliability of our information decoding approach.

Table 4.10: **Experimental results on LLM analogy-style completion tasks with VSA-based probing for retrieving the target concept.** Models are ordered by precision@1 for next-token prediction. Statistical variability is reported using 95% confidence intervals. To control for randomness, two baselines, using Llama 3.1-8B, are included: a null model with randomly permuted input embeddings (*permuted baseline*) and extraction of concept pairs unrelated to the inputs (*unrelated baseline*).

| MODEL | LLM's Next Token | | DLA-based Probing | VSA-based Probing | |
|---|---|---|---|---|---|
| | Precision@1 | Precision@5 | Precision@1 | Precision@1 | Precision@5 |
| *Permuted baseline* | - | - | - | 0.080 (079-082) | 0.103 (101-104) |
| *Unrelated baseline* | - | - | - | 0.099 (097-101) | 0.105 (103-107) |
| Llama 4 Scout, 17B-16E | 0.081 (079-082) | 0.501 (498-504) | **0.773** (771-775) | 0.866 (864-868) | 0.875 (873-877) |
| GPT-2, medium | 0.267 (265-270) | 0.548 (545-551) | 0.300 (298-303) | 0.692 (689-695) | 0.702 (699-704) |
| Pythia-1.4b | 0.369 (366-372) | 0.654 (651-657) | 0.413 (410-416) | 0.778 (776-781) | 0.790 (788-792) |
| Llama 3.1-8B | 0.352 (349-354) | 0.546 (543-549) | 0.467 (464-470) | **0.891** (889-893) | **0.908** (907-910) |
| Phi 4 | 0.519 (516-522) | **0.753** (750-755) | 0.585 (582-588) | 0.887 (886-889) | 0.904 (902-905) |
| OLMo-2 | **0.529** (526-532) | 0.721 (719-724) | 0.714 (712-717) | 0.879 (877-881) | 0.892 (890-894) |
| **AVERAGE** | 0.352 | 0.621 | 0.542 | 0.832 | 0.845 |

### 4.3.3 Comparison with Logit Attribution

For validation, VSA-based findings are compared with conventional output distribution inspection. Specifically, Direct Logit Attribution (DLA)[4] is applied to all six language models, using $\bar{S}$, to underscore the limits of traditional logit-based investigations restricted to an LLM's output token space. Section C.4 provides details on the comparison choices and further analysis.

I adopt fuzzy token-to-concept matching with the concept set,[5] and consider projected next-token predictions from the model's middle to last layers of the last token, as VSA probing. DLA produces no concepts in nearly 30% of analogies on average ($\mathcal{D}$ in Table 4.11; +17% compared to VSA in Table 4.8). It also exhibits lower performance and greater variability in target concept retrieval, with a similarly large drop in comparison to VSA-based precision@1 (Figure 4.9).

In instances without concepts from DLA ($\mathcal{D} \subset \bar{\mathcal{S}}$ in Table 4.11), our VSA probing extracts, on average, the key-target pair in 57% of all analogies, while returning none for 28%. For instance, for the analogy `king is to queen as son is to` $\mapsto$ `daughter`, using OLMo-2, our probe extracts the key-target concepts (`son` and `daughter`), while DLA produces no concepts. The model predicts the next token prediction as `?` with a softmax score of 0.06, followed by `father` (0.05); the target word has a rank of 37.

Since conventional output inspections operate within the model's vocabulary space, *token-level logit analyses often produce shallower findings and show greater variability*

[4] Also known as Logit Lens by Nostalgebraist [119].
[5] For instance, the token "pes" is mapped into the concept `peso`.

Table 4.11: **Concepts extracted though VSA-based probing when DLA yields no concepts.** The table highlights VSA can also capture model's variability (e.g., in-context concepts, target concepts). The table highlights key shared items across models, covering nearly 98% of all extracted concepts.

| | GPT-2 | Pythia | Llama 3.1 | Phi-4 | OLMo-2 | Llama 4, Scout | AVERAGE |
|---|---|---|---|---|---|---|---|
| NONE: no concepts from DLA (%; $\mathcal{D} \subset \bar{\mathcal{S}}$) | 33.9 | 32.8 | **47.4** | 33.1 | 14.6 | 15.4 | 29.5 ± 11.4 |
| Concepts extracted from $\mathcal{D}$ by VSA (%) | | | | | | | |
| Key \| **Target** | 53.5 | 56.5 | **76.6** | 70.5 | 44.5 | 42.6 | 57.4 ± 12.5 |
| NONE | 26.8 | 24.3 | 13.7 | 18.6 | 41.0 | **43.2** | 27.9 ± 10.9 |
| Example | **6.8** | 3.0 | 2.1 | 2.1 | 3.4 | 2.0 | 3.2 ± 1.7 |
| Key | 5.2 | **7.3** | 0.7 | 0.9 | 1.6 | 3.0 | 3.1 ± 2.4 |
| Out-of-context | 2.0 | 1.8 | 1.1 | 1.9 | 3.8 | **4.5** | 2.5 ± 1.2 |
| Key \| Pair Values | 1.8 | 2.1 | 2.5 | **3.2** | 2.4 | 1.8 | 2.3 ± 0.5 |
| Context \| **Target** | 0.7 | 1.2 | 1.1 | 0.7 | **1.9** | 1.4 | 1.2 ± 0.4 |
| **Target** | 0.1 | 0.1 | **0.2** | 0.0 | 0.1 | 0.0 | 0.1 ± 0.1 |

caused likely by exogenous factors, such as prompt design, or endogenous ones.

### 4.3.4 Comparison with Sparse Autoencoders

Lastly, this section compares the VSA-based results with those derived from an analysis using Sparse Autoencoders (SAEs; Section 5.3.3). Section C.4 provides further details on the experimental comparison decisions, including the absence of additional steps required for the the DLA-based approach, and the implications of SAE's bottom-up paradigm.

**Training.** Autoencoders are trained on neural embeddings from each of the six language models, using the training corpus $\mathcal{S}$. Input embeddings are ingested using the same procedure as the VSA-based method (Algorithm 1; Section 3.3.4), which eliminates the need for the conventional search over the optimal model's hidden layer. We define a shallow encoder-decoder neural network to reconstruct LLM's embeddings with a sparsity constraint in its latent representations. Specifically, the SAE's encoder is a fully-connected linear layer with a ReLU activation function, while the decoder has a single linear layer. The SAE's latent dimensionality is defined as a fixed multiple of the input embedding dimension, specifically a factor of 4. For instance, the SAE for GPT-2 uses a latent size of 4096 ($1024 \times 4$), whereas OLMo-2's uses 20480 ($5120 \times 4$). Regarding sparsity, we adopt a top-k sparsity constraint in the forward process [103], where $k$ is set to a fixed proportion of SAE latent dimensionality (2%, e.g., $k = 82$ for GPT-2, and $k = 410$ for OLMo-2). The training process minimizes the Mean Squared Error (MSE) between the LLM embeddings and the reconstructed ones. L1 regularization is also added to the loss function to further encourage sparsity during training.

Table 4.12: Training performance of SAEs on the test set, order by model size. `BASELINE` compares against a null model using randomly permuted input embeddings.

| Large Language Model | | | Sparse Autoencoder (SAEs) | | Cosine similarity |
|---|---|---|---|---|---|
| **Name** | **Layers from residual stream** | **Embedding dimension** | **Latent dimension** | **Top-k** | |
| `BASELINE` | - | 5120 | 20480 | 410 | 0.584 |
| Llama 4, Scout, 17B-16E | $24^{\text{th}}$ to $48^{\text{th}}$ \|25\| | 5120 | 20480 | 410 | 0.979 |
| OLMo-2 | $32^{\text{nd}}$ to $64^{\text{th}}$ \|33\| | 5120 | 20480 | 410 | 0.976 |
| Phi 4 | $20^{\text{th}}$ to $40^{\text{th}}$ \|21\| | 5120 | 20480 | 410 | 0.985 |
| Llama 3.1-8B | $16^{\text{th}}$ to $32^{\text{nd}}$ \|17\| | 4096 | 16384 | 328 | 0.979 |
| Pythia-1.4b | $12^{\text{th}}$ to $24^{\text{th}}$ \|13\| | 2048 | 8192 | 164 | 0.994 |
| GPT-2, medium | $12^{\text{th}}$ to $24^{\text{th}}$ \|13\| | 1024 | 4096 | 82 | 0.988 |
| | | | | **AVERAGE** | 0.984 $\pm$ 0.01 |

Using a random 70–15–15 split of $\mathcal{S}$ into training, validation, and test sets, the SAEs obtain an average cosine similarity of 0.98 between the input embeddings and their reconstructions on the test set (Table 4.12). Following the VSA-based experimental setup (Section 4.3.1), the trained SAEs are subsequently tested on textual inputs with syntax differing from that used during training ($\bar{\mathcal{S}}$, with $\bar{\mathcal{S}} \neq \mathcal{S}$).

**Feature naming.** The feature naming stage uses a logit-based approach. SAE-activated neurons are interpreted via Direct Logit Attribution, projecting the SAE space into the LLM's output vocabulary using the SAE decoder and LLM's unembedding layer [50]. Specifically, each activated neuron is associated with the top-10 tokens from its induced logit distribution, which are then compared to our concept set using the fuzzy token-to-concept matching function (e.g., "extrov" $\mapsto$ `extrovert`) from Section 4.3.3.

**Concept extraction.** We initially evaluate SAE's ability to extract target concepts from LLM embeddings by checking whether the targets appear among the top-k most strongly activated SAE neurons. Unlike VSA- and logit-based probing, SAEs adopt a bottom-up approach that uncovers an unbounded set of latent features without inherent relevance ranking or filtering, limiting alignment with our bounded input–output concept framework. Consequently, we extend the measurement scope of the previous analysis by considering concepts derived from the top-10 and top-100 SAE neurons, reported as precision@10 and precision@100, respectively. In general, the SAEs achieve an average precision@10 across all models equal to 46% (Figure 4.9 and Table 4.10), reaching comparable performance to VSA probing (probing@1) only when considering

Table 4.13: **Concepts extracted by sparse autoencoders within the top-10 most strongly activated SAE neuron.** `Key|Target` indicates extraction of the key ($b_1$) and value ($b_2$) of the target analogy; `Key` for only $b_1$. `Example` refers to $a_1$ and $a_2$, the in-context example's concepts; `Context|Target` for all concepts. `Out-of-context` indicates concepts not referred to the four in-context concepts.

| Extracted Concepts (%) | GPT-2 | Pythia | Llama 3.1 | Phi-4 | OLMo-2 | Llama 4, Scout | AVERAGE |
|---|---|---|---|---|---|---|---|
| Out-Of-Context | **83.9** | 61.5 | 31.3 | 14.6 | 39.2 | 29.3 | 43.3 |
| Key \| Target \| Out-Of-Context | 2.6 | 11.7 | 45.4 | **54.1** | 37.5 | 42.3 | 32.3 |
| Target \| Out-Of-Context | 2.9 | 4.5 | 7.2 | 8.7 | 4.9 | **9.0** | 6.2 |
| Key \| Out-Of-Context | 2.6 | 4.4 | 5.3 | **6.2** | 4.2 | 5.4 | 4.7 |
| Context \| Target \| Out-Of-Context | 0.0 | 1.7 | 0.0 | 3.5 | **4.8** | 3.6 | 2.3 |
| Example \| Out-Of-Context | 2.2 | **7.1** | 1.0 | 0.8 | 1.4 | 1.3 | 2.3 |
| Example Value \| Key \| Target \| Out-Of-Context | 0.0 | 0.5 | 0.0 | **4.5** | 2.3 | 2.5 | 1.6 |
| Example Value \| Out-Of-Context | 2.4 | **3.0** | 1.3 | 1.5 | 1.0 | 1.8 | 1.8 |
| Example Key \| Out-Of-Context | **2.1** | **2.1** | 0.5 | 0.5 | 0.5 | 0.6 | 1.1 |
| Example Value \| Target \| Out-Of-Context | 0.0 | 1.1 | 0.0 | **2.1** | 1.0 | 1.6 | 1.0 |
| Example Key \| Key \| Target \| Out-Of-Context | 0.1 | 0.4 | 1.1 | 1.5 | **1.7** | 0.9 | 0.9 |

the top-100 activated SAE neurons (precision@100). Regarding the actual concepts extracted from the top-10 most strongly activated SAE neuron (Table 4.13), the majority of the instances (43%) across all language models produce only concepts that are not referred to the four in-context concepts for inputs (`out-of-context` in Table 4.13). This is followed by the extraction of the key with target concepts (with further out-of-context concepts) in about 32% of the instances, whereas key concepts or target concepts (with further out-of-context concepts) is found respectively in 6.2% and 4.7% of the experimental instances. For instance, in the earlier example (*king is to queen as son is to*), the SAE-based analysis on OLMo-2 detects only concepts that lie outside the bounded in-context concept set (*out-of-context*), surfacing concepts such as `man`, `human`, `girl`, `boy`, and `gender`. This prevents direct answers to the original research questions about in-context input concepts, but it aligns with the input type (gender analogy) and prior output distribution findings (Section 4.3.3), where *father* is among the model's most likely predictions for this input.

While this suggests that SAEs are capable of capturing a broad range of latent features in LLMs, as VSA-based probing, it also highlights an important difference in their operational characteristics. In particular, SAEs lack explicit relevance filtering or task-specific targeting during feature extraction. In contrast to logit- and VSA-based methods, which are constrained by next-token representations and predefined concept sets, respectively; SAEs operate bottom-up, uncovering a wide and potentially unbounded set of features without intrinsic task-aligned ranking. As a result, task-relevant features (e.g., in-context concepts in analogy inputs) may co-occur with many unrelated features, affecting alignment with a subsequent investigation within a task-oriented concept space. Additionally, the choice of $k$ in selecting top-k SAE ac-

tivations introduces a tuning parameter that can influence evaluations and complicate comparisons with methods such as VSA probing that do not require such selection.

The absence of relevance-based feature filtering, stemming from SAEs' objective of uncovering latent structure in an unsupervised manner, can ultimately lead to *noisier interpretability analyses in bounded conceptual frameworks*, such as those required by our research questions. In summary, VSA probing is particularly effective in experimental settings that require a constrained or structured conceptual feature space.

### 4.3.5 From Input-Completion Tasks to Text Generation

Finally, this section investigates the internal state of LLMs during text generation within a Question-Answering (QA) context. The entire probing pipeline (Section 3.3 is applied the popular SQuAD dataset [133], showcasing the proposed approach in a more realistic setting compared to the earlier controlled testbed, and enabling to explore concept-aligned dynamics during the model's generation process.

This dataset evaluates extractive QA, where each answer is a text span within a given context, through questions generated by crowdworkers over Wikipedia articles (context). This aligns with our concept-focused probing, as questions and answers target contextual concepts, allowing us to benchmark feature extraction against features derived from both. The concept set is created by extracting both question-related and answer-related concepts using lexical semantics. Nouns, verbs, and adjectives are identified using WordNet [112] and DBpedia [97]. Using the VSA operation of bundling, 634,368 training examples $\mathcal{Q}$ are generated by progressively pairing SQuAD questions with their lexical features:

$$
\begin{aligned}
&(q_1)\ \text{“What was the } \texttt{name}\text{”} \mapsto \phi_{\text{name}} \\
&(q_2)\ \text{“What was the } \texttt{name}\text{ of the } \texttt{ship}\text{”} \mapsto \phi_{\text{name}} + \phi_{\text{ship}} \\
&(q_3)\ \text{“What was the } \texttt{name}\text{ of the } \texttt{ship}\text{ that } \texttt{Napoleon}\text{”} \mapsto \phi_{\text{name}} + \phi_{\text{ship}} + \phi_{\text{napoleon}} \\
&\dots \\
&(q_n)\ \text{“}\dots\dots\text{”} \mapsto \phi_{\dots} + \phi_{\dots}
\end{aligned}
\tag{4.2}
$$

The neural VSA encoder, trained with $\mathcal{Q}$, achieves a test-set cosine similarity of 0.53, and a binary accuracy of 0.74 for mapping neural embeddings of Llama 3.1 [67] into these VSA encodings. The null model for this setup, with randomly-permuted input embeddings, achieves a cosine similarity of 0.24.

During the inference stage, a separate SQuAD-based corpus $\bar{\mathcal{Q}}$ is leveraged to extract

semantic concepts from LLM's latent representations using the trained neural VSA encoder. Initially, 10,000 items are randomly sampled from SQuAD. Subsequently, each textual input for the inference stage, $q \in \bar{\mathcal{Q}}$, is formatted as a SQuAD-style question, preceded by its corresponding context, to construct the model's prompt as follows:

> "Napoleon III responded with a show of force . . . by the Greek Orthodox Church.
> Q: What was the name of the ship that Napoleon sent to the Black Sea? (4.3)
> A ($\leq$ 3 words):"

The trained encoder is thus applied on $\bar{\mathcal{Q}}$ to probe the concepts encoded in LLM's neural embeddings both before and after text generation. Consequently, two moments are identified in the model's autoregressive generation process of $n$ new tokens $\mathcal{T} = \langle t_1, \ldots, t_n \rangle$. Before text generation is defined as the model's internal state prior to the generation of the first token $e^{[t_1]}$, whereas after text generation refers to the model's state immediately before yielding the final token $e^{[t_n]}$. Afterwards, the VSA encodings generated by the encoder are directly compared with the concept set[6] using cosine similarity with a concept detection threshold of 0.1, as unbinding is not required.

On average, the probe extracts three concepts both before and after text generation. The LLM's performance on the QA task yields an average token-level F1 score of 0.69 (95% CI: 0.68–0.70), with 68% of outputs (95% CI: 67–69) containing target answers.

**Observing concept drift in generation.** The extracted concepts are benchmarked against the question-answer feature sets. The comparison is accomplished by defining semantic-based concept relevance computed with the cosine similarity between concept embeddings and question-answer features.

The average similarity of extracted concepts related to the question decreases after text generation: by 4.8% for the entire sample and by 8.0% in the LLM error subset (32% of the sample), while no significant differences are observed prior to generation (Figure 4.11; left). For answer-related concepts, overall no change is observed before and after text generation, but the LLM error subset shows a slight increase (+3.2–3.5%; Figure 4.11, right). *This suggests that LLM failures may stem from losing focus on the question rather than from a lack of answer-related knowledge.* This hypothesis is supported by a weak positive Spearman correlation (0.2 with a p-value of $1e^{-99}$) between LLM's F1 score and the proportion of question-related concepts extracted after text generation. For example, for the SQuAD query "What do laboratories *try*

[6] Codebook $\Phi$ with all question-related and answer-related concepts.

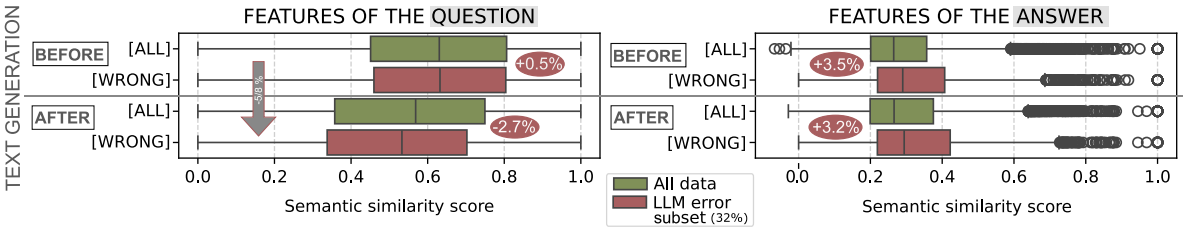



Figure 4.11: Concepts extracted *before* and *after* the LLM's text generation, with respect to *question* and *answer* features. Red denotes the subset of failure instances, while green the full sample $\bar{\mathcal{Q}}$.

to *produce hydrogen* from?" (target answer: "*solar energy* and *water*"), the model erroneously outputs "*water* and heat" (F1 = 0.57). Before the model's text generation, our proposed approach extracts the concepts $\phi_{\text{try}}$, $\phi_{\text{produce}}$, $\phi_{\text{hydrogen}}$ (question) and $\phi_{\text{solar}}$, $\phi_{\text{water}}$ (answer); after generation, the question-related concept set reduced to $\phi_{\text{produce}}$ and the answer set gained the concept $\phi_{\text{energy}}$. During text generation, the model lost the concept of hydrogen while refining concepts related to answering. This leads to a response that no longer focuses on the question's subject, but reflect the more general concepts still active in the model's internal state.

**GPT-2's internal state confirms its in-context overreliance.** To contrast the QA-related findings from the encoder trained on a medium-scale language model (Llama 3.1; 8B parameters), we also evaluate this setup using GPT-2, a smaller language model with 355M parameters. Regarding the model's QA-related performance, GPT-2 archives an average token-based F1 score of 0.133 (95% CI: 0.13–0.14), and 21% of the outputs (95% CI: 20–22) contained the target answers. On the other hand, the encoder trained on its neural embeddings achieves a test-set cosine similarity of 0.52 and a binary accuracy of 0.74, comparable to llama's training performance. We then apply the trained probe to $\bar{\mathcal{Q}}$, probing the model's internal state both before and after text generation. Our probe extracts an average of six concepts before both the first and last token generation. The average semantic-oriented similarity of extracted concepts decreases after text generation by 13.7% for question-related concepts and 4.5% for answer-related concepts. Unlike the prior analysis, no concept-based differences are observed in the LLM error subset (79%), suggesting erroneous answers of GPT-2 may stem from a different reason than llama's shifted focus. For the aforementioned example (i.e., "What do laboratories *try* to *produce hydrogen* from?"), GPT-2 erroneously outputs "the most important step in the process is to use a large amount of *water* to produce" (F1 = 0.11). Before the model generates text, our probe identifies the concepts $\phi_{\text{produce}}$ (question),

$\phi_{\text{solar}}$ (answer), and the out-of-domain concepts $\phi_{\text{name}}, \phi_{\text{process}}, \phi_{\text{use}}, \phi_{\text{work}}$. After generation, the answer concept shifts to $\phi_{\text{water}}$, no question-related concepts are detected, and the out-of-domain set changes, gaining $\phi_{\text{use}}, \phi_{\text{one}}, \phi_{\text{main}}, \phi_{\text{vacuum}}$ while losing $\phi_{\text{name}}$ and $\phi_{\text{work}}$. While losing question-related concepts aligns with llama's findings, the wide set of out-of-domain, yet topic-related concepts confirms earlier observations of GPT-2's latent representations when processing analogy-style inputs (Section 4.3.2). The model's internal representations tend to fall back on concepts closely associated with the input text (e.g., *use*, *process*, *name*), suggesting a shallow grasp of the downstream objective; namely, answering with the energy sources used for hydrogen production. This behavior corresponds with the model's weak performance on the QA-based task.

# Chapter 5

# Related Work

This chapter presents previous work related to the challenges addressed with the three approaches presented in Chapter 3. Section 5.1 outlines prior work on Natural Language Processing applied on companies' non-financial disclosures. This is followed by an overview of the existing methods for generating Knowledge Graphs (Section 5.2). Lastly, Section 5.3 presents existing approaches to extract the semantic and factual information encoded in neural embeddings.

## 5.1 Text Analytics on Non-financial Disclosures

Several works have already explored the use of NLP techniques to process companies' non-financial documents and extract meaningful insights concerning statements, facts and actions disclosed by companies. For example, Chou *et al.* [30] applied distributional semantics on 10-K filings[1] for extracting topics related to climate change that are disclosed by companies, aiming to monitor environmental policy compliance. Raghupathi *et al.* [132] applied traditional approaches ranging from trigram co-occurrences to clustering and classification to gain insights into the shareholders' perspectives and objectives regarding sustainability and climate change.

Large Language Models (LLMs) have also been used for text analytics in the domain of Environmental, Social, and Governance (ESG). Jacouton *et al.* [106, 107] introduced *SDG Prospector*, a tool exploiting language models to identify paragraphs in Public Development Banks' sustainability reports that address Sustainable Development Goals (SDGs). Similarly, Webersinke *et al.* [158] introduced *ClimateBert*, a transformer-based language model fine-tuned for climate-related classification tasks. Vaghefi *et al.* [155]

[1] Annual financial reports required by the U.S. Securities and Exchange Commission.

adopted the paradigm of RAG [98] to enrich GPT-4 [3] with the ability to reliably answer climate-related questions, by augmenting the posed questions with contextual information retrieved from the Sixth Assessment Report, released by the United Nations Intergovernmental Panel on Climate Change (IPCC). The authors [155] also released a conversational agent based on their proposed approach. Ni *et al.* [116], developed *ChatReport*, an LLM-based tool that evaluates companies' sustainability reports according to the eleven recommendations [149] provided by the Task Force for Climate-Related Financial Disclosures (TCFD). The tool combines semantic search to identify text chunks that are pertinent to each recommendation and prompting to summarize them. In contrast, the approach presented in Section 3.1 differs from existing approaches by jointly:

1. leveraging the generative abilities of LLMs to extract graph-ready information for constructing ESG-aligned KGs, rather than targeting classification tasks or chat-based applications;

2. employing an open-sourced language model rather than a commercial one, such as one from OpenAI's GPT family;

3. relying directly on companies' non-financial reports rather than third-party data.

The proposed method, together with graph representations, enables the extraction of non-trivial, data-driven insights into the ESG topics and actions disclosed by companies.

## 5.2 Knowledge Graphs Generation

Knowledge Graphs (KGs; 59, 78) offer a versatile approach to represent real-world knowledge that can be leveraged in various use cases and domains [83, 171], such as question-answering [84], recommendation systems [145], and information retrieval [136]. The task of Knwoledge Graph Generation, also known as Knowledge Acquisition, populates a graph representation with information extracted from unstructured, semi-structured or structured sources as well as augmenting existing graphs [58, 83, 161]. Traditional approaches involve several NLP tasks that are disjointly learned, leading to an information-extraction process that is prone to error accumulation. To overcome this problem, new single-stage NLP pipelines have been proposed to jointly extract both entities and relations [90, 157].

In this context, Open Information Extraction (OIE) has emerged as the general-purpose task to extract structured information from textual content [117]. OIE methods rely generally on the syntactic structure of the input sentence, generating Subject-

Predicate-Object (SPO) triples. These syntax-driven triples can then be leveraged to generate knowledge graphs based on the subjects, predicates and objects extracted from all sentences. This approach can mitigate the impact of depending on external knowledge, such as patterns and domain-specific heuristic rules present in the training data. Recently, OIE models, such as Multi$^2$OIE [138] and DeepEx [156], have employed transformer-based models (e.g., BERT; 39) to enhance information extraction with semantic understanding [100]. Several researchers [27, 110, 153, 169] have already demonstrated the ability of Large Language Models, such as OpenAI's GPT models, to extract structured knowledge from textual documents for KG generation. LLMs can also improve Knowledge Graphs in multiple respects; for example, by enriching entity and relation representations through neural embeddings, generating new facts (KG Completion; 28), producing natural-language descriptions of KG facts (KG-to-Text; 10) and Question Answering [127].

The methodology proposed in Section 3.1 pursue this promising direction by leveraging the semantic understanding and generative capabilities of LLMs, with cutting-edge prompting techniques (ICL and RAG), to address key limitations of conventional OIE methods in extracting structured and domain-aligned information from textual documents. Conventional methodologies for Information Extraction are inherently constrained in terms of the types and structural forms of information they can capture, owing to their strong reliance on syntax. In contrast, the usage of LLMs enable the generation of semantically informed, domain-aligned triples rather than purely syntax-driven ones (i.e., SPO), facilitating the construction of domain-oriented KGs as required in the problem statement of Section 3.1.1.

## 5.3 Probing Neural Embeddings

Neural probing refers to the task of extracting latent features from neural embeddings, with the objective of uncovering the semantic and human-interpretable information encoded in these vector representations. Researchers typically study the latent representations of transformer-based models, a high-dimensional linear vector space that aggregates the outputs of all the hidden layers of the model [51]. This investigation is grounded in the linear representation hypothesis, which posits that latent features can be encoded as linear subspaces [52], formed and accessed during the inference of neural models [128].

In recent years, four main paradigms have been adopted to extract latent features from neural embeddings of Large Language Models [60]: logit-based methods

(Section 5.3.1) for output-oriented inspection, together with activation patching (Section 5.3.2), Sparse Autoencoders (SAEs; Section 5.3.3), and supervised probes (Section 5.3.4) for input-focused feature extraction.

Meanwhile, symbolic paradigms have also been applied to interpret neural embeddings, including Vector Symbolic Architectures (VSAs; Section 2.2.2). VSAs are a computational framework [63, 144] inspired by cognitive science [72, 130], increasingly used to map neural representations to human-readable symbols. This includes visual problems such as multi-attribute digit recognition [61] and Raven's progressive matrices [75], and recently, the mechanistic interpretability of LLMs [94]. Specifically, Knittel *et al.* [94] apply VSAs to study the layer-wise dynamics of transformer neural weights (GPT-2), showing that they can be interpreted as VSA-related circuits involving word embeddings, attention mechanisms, and outputs of Multi-Layer Perceptron (MLP). In contrast, the approach introduced in Section 3.3 leverages VSAs for the purpose of information decoding, aiming to extract semantic concepts embedded within the neural embeddings of neural models, rather than focusing on the analysis of contributions made by individual model components.

### 5.3.1 Logit Attribution

Logit attribution projects the model's embedding space onto its output vocabulary through either: the model's unembedding matrix (Logit Lens[2] by Nostalgebraist; 119); a learned affine transformation (Tuned Lens by Belrose *et al.*; 14); or patched LLM inference (Patchscope by Ghandeharioun *et al.*; 64). This lens-style approach offers insight into the model's output distribution by generating projected logits at a chosen point in the forward pass [82], revealing LLM's next-token predictions under the assumption that all subsequent layers are bypassed. This logit-driven paradigm, however, limits the interpretability of neural embeddings, as it relies sorely on the output vocabulary of LLMs. As a result, information extraction is constrained to token-level and surface latent features, restricting abstraction, compositionality, and the scope of features that can be targeted. Consequently, logit-driven analysis remains largely token-aligned and output-oriented, rather than supporting the semantic, higher-level abstraction enabled by the approach proposed in Section 3.3.

Moreover, reliance on the model's output vocabulary makes logit-based analyses sensitive to variations introduced by prompt design and tokenization. By contrast, the proposed approach remains unaffected by such exogenous and endogenous factors.

[2] also known as Direct Logit Attribution (DLA).

### 5.3.2 Activation Patching

The technique of activation patching is traditionally applied in mechanistic interpretability to study components and internal workings of machine-learned models. Conventional workflows involve eliciting a behaviour to observe, discovering patterns via patching and generating functional hypotheses [33]. Neural networks are viewed as computational graphs, and circuits are sub-graphs with distinct functionalities [33]. The patching technique studies the model's inference process by altering its latent representations during the inference process (64; Section 2.2.1), the neural embeddings of tokens in LLMs. For example, Meng *et al.* [108] focused on causal tracing by replacing the latent representation of a noise-corrupted inference with the correct ones, identifying the crucial activations in rectifying the correction.

While traditional patching approaches focus on localising information within a model's internal components, Patchscopes by Ghandeharioun *et al.* [64] apply the activation patching technique to information decoding. This shifts the scope from investigating where the information is encoded to identifying what information is encoded. The authors [64] propose a patching-based framework that employs the same language model that produced the neural embeddings to extract the semantics encoded in these vector representations. They introduce a de-facto probe based on LLMs that eliminates the need for training a tailored probing model. They adopt activation patching to dynamically replace the latent representation of a placeholder token during a patched inference of a language model. This separate inference is tasked with interpreting the semantic content of the injected vector representation. Specifically, Ghandeharioun *et al.* [64] study the entity resolution process across the early layers of LLMs by patching the neural embeddings of a subject entity, such as "Alexander the Great", into a separate inference of the same model. The latter is tasked with generating a natural-language description of the injected entity-related vector representation.

Following the framework proposed by Ghandeharioun *et al.* [64], the approach described in Section 3.2 exploits the patching technique to investigate the factual knowledge resolution process in LLMs. The proposed method adopts a patching-based probe to decode the semantics, in the form of factual information, embedded in the neural embeddings of LLMs. Graph representation is then exploited to represent the layer-wise evolution of the extracted information. Although Ghandeharioun *et al.* [64] open a novel use case for activation patching; their study, along with others [74, 85, 104] focuses on the factual understanding of LLMs regarding single entities in scenarios involving knowledge completion, such as incomplete triplets. In contrast, the approach

of Section 3.2 focuses on entire sentences, exploring the extensiveness and evolution of factual knowledge embedded within LLMs when tasked with evaluating the truthfulness of the input. This represents a step towards understanding the factual knowledge resolution process rather than factual knowledge of single entities.

### 5.3.3 Sparse Autoencoders

Sparse Autoencoders (SAEs) provide input-oriented feature extraction using a proxy space learned via sparse dictionary learning [123], uncovering superposed latent features encoded in neural embeddings [37]. An autoencoder reconstructs the neural embeddings of an LLM in an unsupervised fashion, while enforcing sparsity in its learned internal representations. The sparsity constraint helps to isolate latent atomic features, which might be represented as a linear combination of multiple neurons due to superposition. Superposition denotes an underlying representational mechanism in which latent features are encoded in overlapping neuron activations, allowing more features to be represented than permitted by the vector dimensionality [60].

Once trained, these SAE's learned representations serve as a proxy layer for the analysis of the original neural embeddings. Subsequently, activated SAE neurons are interpreted using two main strategies: (1) identifying representative tokens via logit-based methods (Section 5.3.1; 47, 92); or (2) grouping textual inputs that share SAE neuron activations, followed by manual [86] or automatic [19, 99] feature naming.

The approach presented in Section 3.3 addresses several shortcomings of SAEs that hinder the comprehensive interpretability of neural embeddings:

1. their *unbounded feature space* yields a set of latent features that are difficult to control and align for subsequent interpretability analyses;

2. their *feature-naming* stage restricts semantic grounding to output-vocabulary labels or ambiguous, data-dependent descriptors;

3. their dependence on *layer-localized representations* requires identifying a task-specific optimal hidden layer.

The experimental comparison presented in Section 4.3.4 validates the advantages of VSA-based probing compared to a SAE-based analysis. While the findings suggest that SAEs are capable of capturing a broad range of latent features in LLMs as VSA probing, they also highlight an important difference in their operational characteristics. Unlike VSA probing and logit-based methods, SAEs adopt a bottom-up approach

that uncovers an unbounded set of latent features without inherent relevance ranking or filtering, a constraint of SAE-based analysis. This limits alignment with our bounded input–output concept framework. The absence of relevance-based feature filtering, stemming from SAEs' objective of uncovering latent structure in an unsupervised manner, can ultimately lead to *noisier interpretability analyses in bounded conceptual frameworks*, such as those required by our research questions. Thus, VSA probing is particularly effective in experimental setups that require a constrained or structured conceptual feature space.

### 5.3.4 Supervised Probes

Researchers have also begun training supervised models[3] to map neural embeddings onto human-interpretable features, assessing the extent to which these features are captured in the embeddings [150]. Supervised probes is a mapping paradigm that projects the model's latent representations to task-relevant features, measuring how much information about them is embedded, interpreting neural probing as a classification task.

These probes are generally designed for specific experimental goals and tailored to task-aligned features, investigating whether transformer-based models capture those features through linear representations. For example, researchers target a wide range of latent feature, ranging from syntactic structure, as in "A Polar Coordinate System Represents Syntax in Large Language Models" [40]; to real-world knowledge, as in "Language Models Represent Space and Time" [70]; and to abstract semantics, as in "The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets" [105]. However, tailoring probes to specific tasks limits generalizability, while directly learning targeted features complicates distinguishing actual information decoding from probe-induced learning. This dichotomy in learning sources also contributes to the ongoing debate [76] about the effectiveness of these probes in decoding information, as accurately assessing probe performance remains an open challenge in probe-based studies.

Although the method described in Section 3.3 qualifies as a supervised probe, since it maps LLM internal representations to human-understandable features, it effectively functions as a *hybrid supervised probe*. The proposed paradigm integrates the *top-down interpretability* of conventional supervised probes, the SAE's ability to learn a *sparsity-driven proxy space*, and a higher-level, *jointly input–output analytic perspective* that goes beyond conventional logit-based methods. The VSA-based proxy space

[3] Often referred to as probes.

avoids both the unbounded feature spaces of SAEs and the necessity for post-hoc feature naming or filtering (Section 5.3.3). Further, the embedding-ingestion algorithm (Section 3.3.4) *removes the requirement to identify an optimal hidden layer of prior probing paradigms*, such as probes, SAEs, and lens-style methods. It also *mitigates the dichotomy between information decoding and probe-induced learning* of conventional probes by learning a vector transformation from neural embeddings to a controlled proxy space (Section 3.3.5) rather than targeting latent features directly. Lastly, it addresses the core limitations of conventional output distribution investigations (logit attribution; Section 5.3.1) by supporting feature sets of arbitrary abstraction, cardinality, and symbolic origin.

Overall, the approach described in Section 3.3 harnesses Vector Symbolic Architectures and hypervector algebra to address key shortcomings of traditional neural probing methods (section 5.3), bringing together their complementary perspectives:

1. define a bounded feature space (probes) rather than unbounded space (SAEs);
2. learn a sparse proxy space (SAEs) rather than target task-specific feature (probes);
3. query a proxy space holistically (SAEs), in contrast to classification-based probes;
4. inspect the latent representations without layer selection (probes, logit, SAEs);
5. target concept-oriented latent features (VSAs) rather than token-aligned features.

# Chapter 6

# Conclusions

This PhD thesis follows the disruptive transformation of the Natural Language Processing community caused by the broad adoption of Large Language Models, covering the use of LLMs in conventional NLP tasks and the investigation of their underlying mechanisms. Following the research efforts of the community, this thesis has answered research questions concerning the practical usage of LLMs as black-box tool for conventional tasks and advancing the interpretability of their internal representations, tackling two distinct challenges:

- applying LLMs for domain-driven Information Extraction (Section 3.1), and
- extracting semantics from neural embeddings of LLMs (Sections 3.2 and 3.3).

Although the subject of the study shifts from textual documents to neural embeddings, the underlying task remains the extraction of structured information.

The approach described in Section 3.1 [20] takes advantage of the generative capabilities of LLMs to extract structured information aligned with a specific domain from complex textual documents (i.e., companies' non-financial reports), generating a Knowledge Graph centred on Environmental, Social, and Governance aspects. This methodology, powered by LLMs, overcomes one of the main constraints of conventional Open Information Extraction methods (Section 5.2): extracting structured information aligned with a predefined data schema (RQ1-RQ2) rather than the syntactic structure of input sentences. In-Context Learning provides the language model with domain-aligned input-output examples, guiding extraction towards syntax-free and domain-aligned structured information. On the other hand, Retrieval-Augmented Generation improves the alignment between the extraction and the domain by incorporating domain-relevant knowledge (e.g., ESG topics) into the model instruction. The semantic understanding of

LLMs, along with the flexibility provided by these two prompting techniques (RQ3), reveals non-trivial and data-driven insights (Section 4.1, RQ4) into companies' non-financial reporting (Section 2.1). The usage of LLMs enables a deeper and data-driven exploration of companies' non-financial actions, providing novel methodological and experimental contributions compared to conventional text analytics on non-financial disclosures (section 5.1).

Subsequently, the research questions in this PhD thesis extend beyond treating LLMs as black-box tools and instead tackle the limited understanding of their internal representations. The approach presented in Section 3.2 [21], based on activation patching (Section 2.2.1), investigates the factual knowledge resolution process of LLMs, unveiling which factual and common-sense knowledge is represented in the latent representations of LLMs during their inference. This work addresses the limited scope of prior studies by focusing on the factual knowledge conveyed by entire sentences, rather than on individual entities in knowledge-completion tasks. As reported in Section 4.2, this probing approach performs both local and global interpretability analyses of the semantic information internally represented by LLMs during claim verification. It advances our understanding in the internal mechanisms of LLMs, uncovering non-trivial insights into the dynamics of LLM's knowledge resolution process from extracting the factual information represented internally by LLMs (RQ5), to identifying layer-wise patterns in this latent process (RQ6-RQ7).

On the other hand, the third research work (Section 3.3, 22) tackles the limited understanding of the internal representations of LLMs from another perspective. It addresses research questions concerning the output-driven interpretability of the atomic semantic concepts encoded in LLM's vector space. It introduces a novel probing paradigm that combines ideas from neural probing and symbolic representations, exploiting the properties of Vector Symbolic Architectures and hypervector algebra (Section 2.2.2) for concept-based Information Extraction. Inspired by the sparsity constraint of Sparse Autoencoders (SAEs), it unifies the complementary viewpoints of conventional probing methods for joint input–output investigation, from top-down interpretability of standard supervised probes to queryable proxy space of SAEs (Section 5.3). This novel probing paradigm is tested on the latent representations of six diverse LLMs, and across two distinct experimental settings (RQ8), yielding concept-driven insights into the latent representations of LLMs. The experimental results in Section 4.3 address research questions on LLM reasoning, revelling non-trivial patterns across the semantic concepts represented internally by diverse LLMs and across different input types (RQ9). The proposed approach also enables an investigation of vector space of LLMs during

QA-driven text generation (RQ10), revealing how question-related and answer-related concepts manifest before and after generation.

This research project started at the end of 2022, coinciding with the disruptive transformation of the Natural Language Processing community driven by Large Language Models. This timing placed me, both exhaustingly and uniquely, at the center of one of the most rapidly evolving research landscapes in recent years. Following the research developments in the community, the research questions of this PhD thesis progressively shifted from using LLMs as black-box tools toward investigating and interpreting their underlying mechanisms In addressing the questions, the novel methodologies proposed in this PhD thesis advance both the application of LLMs to traditional Information Extraction tasks and the semantic interpretability of their neural embeddings.

# Bibliography


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

[2] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[4] AI@Meta. Llama 3 model card. *Meta AI*, 2024.

[5] Elmira Aliakbari and Steven Globerman. The impracticality of standardizing esg reporting. *ESG*, page 117, 2023.

[6] Allen Institute for AI. Open information extraction - demo.

[7] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

[8] Artifex. PyMuPDF. https://pymupdf.readthedocs.io/en/latest.

[9] Armen S Asratian, Tristan MJ Denley, and Roland Häggkvist. *Bipartite graphs and their applications*, volume 131. Cambridge university press, 1998.

[10] Agnes Axelsson and Gabriel Skantze. Using large language models for zero-shot natural language generation from knowledge graphs, 2023.

[11] Maria Baldini, Lorenzo Dal Maso, Giovanni Liberatore, Francesco Mazzi, and Simone Terzani. Role of country-and firm-level determinants in environmental, social, and governance disclosure. *Journal of Business Ethics*, 150:79–98, 2018.

[12] Marc Barthelemy. Betweenness centrality in large complex networks. *The European physical journal B*, 38(2):163–168, 2004.

[13] Hannah Bast, Björn Buchhold, and Elmar Haussmann. Semantic search on text and knowledge bases. *Foundations and Trends® in Information Retrieval*, 10(2-3):119–271, 2016.

[14] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

[15] Florian Berg, Julian F Koelbel, and Roberto Rigobon. Aggregate confusion: The divergence of esg ratings. *Review of Finance*, 26(6):1315–1344, 2022.

[16] Daniel Berrar. Cross-validation, 01 2018.

[17] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[18] Ulrik Brandes. *Network analysis: methodological foundations*, volume 3418. Springer Science & Business Media, 2005.

[19] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, page 2, 2023.

[20] Marco Bronzini, Carlo Nicolini, Bruno Lepri, Andrea Passerini, and Jacopo Staiano. Glitter or Gold? Deriving structured insights from Sustainability Reports via Large Language Models. *EPJ Data Science*, 13(1):41, 2024. 10.1140/epjds/s13688-024-00481-2.

[21] Marco Bronzini, Carlo Nicolini, Bruno Lepri, Jacopo Staiano, and Andrea Passerini. Unveiling LLMs: The Evolution of Latent Representations in a Dynamic Knowledge Graph. In *First Conference on Language Modeling*, 2024. 10.48550/arXiv.2404.03623.

[22] Marco Bronzini, Carlo Nicolini, Bruno Lepri, Jacopo Staiano, and Andrea Passerini. Hyperdimensional Probe: Decoding LLM Representations via Vector Symbolic Architectures, 2025. *arXiv preprint* 10.48550/arXiv.2509.25045.

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

[24] Cristiano Busco, Costanza Consolandi, Robert G Eccles, and Elena Sofra. A preliminary analysis of sasb reporting: Disclosure topics, financial relevance, and the financial intensity of esg materiality. *Journal of Applied Corporate Finance*, 32(2):117–125, 2020.

[25] Stefan Buttcher, Charles LA Clarke, and Gordon V Cormack. *Information retrieval: Implementing and evaluating search engines*. Mit Press, 2016.

[26] John L Campbell. Why would corporations behave in socially responsible ways? an institutional theory of corporate social responsibility. *Academy of management Review*, 32(3):946–967, 2007.

[27] Salvatore Carta, Alessandro Giuliani, Leonardo Piano, Alessandro Sebastian Podda, Livio Pompianu, and Sandro Gabriele Tiddia. Iterative zero-shot llm prompting for knowledge graph construction, 2023.

[28] Jiao Chen, Luyi Ma, Xiaohan Li, Nikhil Thakurdesai, Jianpeng Xu, Jason H. D. Cho, Kaushiki Nag, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Knowledge graph completion models are few-shot learners: An empirical study of relation labeling in e-commerce with llms, 2023.

[29] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

[30] Christine Chou, Robin Clark, and Steven O Kimbrough. What do firms say in reporting on impacts of climate change? an approach to monitoring esg actions

and environmental policy. *Corporate Social Responsibility and Environmental Management*, 2023.

[31] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[32] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.

[33] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36, 2024.

[34] Luciano da F Costa. Further generalizations of the jaccard index. *arXiv preprint arXiv:2110.09619*, 2021.

[35] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[36] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading, 2010.

[37] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[38] Lennert De Smet, Pedro Zuidberg Dos Martires, Robin Manhaeve, Giuseppe Marra, Angelika Kimmig, and Luc De Readt. Neural probabilistic logic programming in discrete-continuous domains. In *Uncertainty in Artificial Intelligence*, pages 529–538. PMLR, 2023.

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

[40] Pablo J Diego Simon, Stéphane d'Ascoli, Emmanuel Chemla, Yair Lakretz, and Jean-Rémi King. A polar coordinate system represents syntax in large language models. *Advances in Neural Information Processing Systems*, 37:105375–105396, 2024.

[41] Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. Climate-fever: A dataset for verification of real-world climate claims. *Tackling Climate Change with Machine Learning workshop at NeurIPS 2020*, 2020.

[42] Clara Dismuke and Richard Lindrooth. Ordinary least squares. *Methods and designs for outcomes research*, 93(1):93–104, 2006.

[43] Juris Dobrick, Christian Klein, and Bernhard Zwergel. Size bias in refinitiv esg data. *Finance Research Letters*, 55:104014, 2023.

[44] Timothy M Doyle. Ratings that don't rate: the subjective world of esg ratings agencies. *American Council for Capital Formation*, pages 65–71, 2018.

[45] Norman R Draper and Harry Smith. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998.

[46] Samuel Drempetic, Christian Klein, and Bernhard Zwergel. The influence of firm size on the esg score: Corporate sustainability ratings under review. *Journal of business ethics*, 167:333–360, 2020.

[47] Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits. *arXiv preprint arXiv:2406.11944*, 2024.

[48] Robert G Eccles, Michael P Krzus, Jean Rogers, and George Serafeim. The need for sector-specific materiality and sustainability reporting standards. *Journal of applied corporate finance*, 24(2):65–71, 2012.

[49] Torsten Ehlers, Ulrike Elsenhuber, Kumar Jegarasasingam, and Eric Jondeau. Deconstructing esg scores: How to invest with your own criteria? *IMF Working Papers*, 2023(057):A001, 2023.

[50] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

[51] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy

Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[52] Joshua Engels, Isaac Liao, Eric J Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are linear. *arXiv e-prints*, pages arXiv–2405, 2024.

[53] Ernesto Estrada. *The Structure of Complex Networks: Theory and Applications*. Oxford University Press, 10 2011.

[54] European Union. Corporate sustainability reporting directive (csrd).

[55] European Union. Eu emissions trading system (eu ets).

[56] European Union. Fit for 55.

[57] Katherine Faust. Centrality in affiliation networks. *Social networks*, 19(2):157–191, 1997.

[58] Dieter Fensel, U Simsek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, and Alexander Wahler. *Knowledge graphs*. Springer, 2020.

[59] Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, Alexander Wahler, Dieter Fensel, et al. Introduction: what is a knowledge graph? *Knowledge graphs: Methodology, tools and selected use cases*, pages 1–10, 2020.

[60] Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-Jussà. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*, 2024.

[61] E Paxon Frady, Spencer J Kent, Bruno A Olshausen, and Friedrich T Sommer. Resonator networks, 1: An efficient solution for factoring high-dimensional, distributed representations of data structures. *Neural computation*, 32(12):2311–2331, 2020.

[62] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

[63] R. W. Gayler. Multiplicative binding, representation operators & analogy. In D. Gentner, K. J. Holyoak, and B. N. Kokinov, editors, *Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences*, pages 1–4, 1998.

[64] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: a unifying framework for inspecting hidden representations of language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[65] Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15. ACL, 2016.

[66] Gotelli, Nicholas J, and Werner Ulrich. Statistical challenges in null model analysis. *Oikos*, 121(2):171–180, 2012.

[67] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[68] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 2024.

[69] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–42, 2022.

[70] Wes Gurnee and Max Tegmark. Language models represent space and time. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Representation Learning*, volume 2024, pages 2483–2503, 2024.

[71] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[72] Jeff Hawkins. *A thousand brains: a new theory of intelligence*. Basic Books, 2021.

[73] Mike Heddes, Igor Nunes, Pere Vergés, Denis Kleyko, Danny Abraham, Tony Givargis, Alexandru Nicolau, and Alexander Veidenbaum. Torchhd: An open source python library to support research on hyperdimensional computing and vector symbolic architectures. *Journal of Machine Learning Research*, 24(255):1–10, 2023.

[74] Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. *arXiv preprint arXiv:2308.09124*, 2023.

[75] Michael Hersche, Mustafa Zeqiri, Luca Benini, Abu Sebastian, and Abbas Rahimi. A neuro-vector-symbolic architecture for solving raven's progressive matrices. *Nature Machine Intelligence*, 5(4):363–375, 2023.

[76] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics.

[77] John Hewitt, Christopher D Manning, and Percy Liang. Truncation sampling as language model desmoothing. *arXiv preprint arXiv:2210.15191*, 2022.

[78] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

[79] Hugging Face. Statistics on the number of monolingual models by language hosted on the hugging face platform.

[80] IR Solutions. Responsibilityreports.

[81] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[82] Stanislaw Jastrzebski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. Residual connections encourage iterative inference. *CoRR*, abs/1710.04773, 2017.

[83] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

[84] Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 792–802, 2021.

[85] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[86] Yi Jing, Zijun Yao, Lingxu Ran, Hongzhu Guo, Xiaozhi Wang, Lei Hou, and Juanzi Li. Sparse auto-encoder interprets linguistic features in large language models. *arXiv preprint arXiv:2502.20344*, 2025.

[87] Tom Jobbins. TheBloke/wizardLM-7B-HF · Hugging Face. https://huggingface.co/TheBloke/wizardLM-7B-HF. [Accessed 25-09-2023].

[88] Pentti Kanerva. *Sparse distributed memory*. MIT press, 1988.

[89] Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation*, 1:139–159, 2009.

[90] Arzoo Katiyar and Claire Cardie. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, 2017.

[91] Mozaffar Khan, George Serafeim, and Aaron Yoon. Corporate sustainability: First evidence on materiality. *The accounting review*, 91(6):1697–1724, 2016.

[92] Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. Interpreting attention layer outputs with sparse autoencoders. *arXiv preprint arXiv:2406.17759*, 2024.

[93] D. Kleyko, R. W. Gayler, and E. Osipov. Commentaries on "learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception" [science robotics vol. 4 issue 30 (2019) 1-10]. *arXiv:2003.11458*, pages 1–10, 2020.

[94] Johannes Knittel, Tushaar Gangavarapu, Hendrik Strobelt, and Hanspeter Pfister. Gpt-2 through the lens of vector symbolic architectures. *arXiv preprint arXiv:2412.07947*, 2024.

[95] Michael J LaBella, Lily Sullivan, Josh Russell, and Dmitry Novikov. The devil is in the details: the divergence in esg data and implications for responsible investing. *New York: QS Investors*, 2019.

[96] Michel Ledoux. *The concentration of measure phenomenon*. Number 89 in Mathematical Surveys and Monographs. American Mathematical Soc., 2001.

[97] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

[98] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[99] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*, 2024.

[100] Pai Liu, Wenyang Gao, Wenjie Dong, Songfang Huang, and Yue Zhang. Open information extraction from 2007 to 2022 – a survey, 2022.

[101] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9), January 2023.

[102] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vish-

wanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[103] Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

[104] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics.

[105] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024.

[106] Régis Marodon, Jean-Baptiste Jacouton, and Adeline Laulanié. The Proof is in the Pudding. Revealing the SDGs with Artificial Intelligence. Working Paper 85f81dba-c8e2-4255-878a-0, Agence française de développement, October 2022.

[107] Régis Marodon, Jean-Baptiste Jacouton, Adeline Laulanié, and François Bullier. SDG Prospector - artificial intelligence serving the sdgs. https://www.sdgprospector.org, 2023. [Accessed 10-12-2025].

[108] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[109] Meta AI. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence, 2025.

[110] Lars-Peter Meyer, Claus Stadler, Johannes Frey, Norman Radtke, Kurt Junghanns, Roy Meissner, Gordian Dziwis, Kirill Bulert, and Michael Martin. Llm-assisted knowledge graph engineering: Experiments with chatgpt. *arXiv preprint arXiv:2307.06917*, 2023.

[111] Tomas Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781, 2013.

[112] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[113] Niklas Muennighoff. Sgpt: Gpt sentence embeddings for semantic search, 2022.

[114] Lloyd S Nelson. The anderson-darling test for normality. *Journal of Quality Technology*, 30(3):298, 1998.

[115] Mark Newman. *Networks: An Introduction*. Oxford University Press, Inc., USA, 2010.

[116] Jingwei Ni, Julia Bingler, Chiara Colesanti-Senni, Mathias Kraus, Glen Gostlow, Tobias Schimanski, Dominik Stammbach, Saeid Ashraf Vaghefi, Qian Wang, Nicolas Webersinke, et al. Paradigm shift in sustainability disclosure analysis: Empowering stakeholders with chatreport, a language model-based tool. *arXiv preprint arXiv:2306.15518*, 2023.

[117] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A survey on open information extraction. *arXiv preprint arXiv:1806.05599*, 2018.

[118] NLP Group of The University of Hong Kong. Instructor-xl · Hugging Face — huggingface.co. https://huggingface.co/hkunlp/instructor-xl. [Accessed 25-09-2023].

[119] Nostalgebraist. Interpreting gpt: The logit lens, 2020.

[120] OECD. *OECD Business and Finance Outlook 2020: Sustainable and Resilient Finance*. Number 6th (2020) in OECD Business and Finance Outlook. OECD, Paris, 2020.

[121] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020.

[122] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

[123] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[124] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[125] R Lyman Ott and Micheal T Longnecker. *An introduction to statistical methods and data analysis*. Cengage Learning, 2015.

[126] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707, 2016.

[127] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*, 2023.

[128] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

[129] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[130] Steven T Piantadosi, Dyana CY Muller, Joshua S Rule, Karthikeya Kaushik, Mark Gorenstein, Elena R Leib, and Emily Sanford. Why concepts are (probably) vectors. *Trends in Cognitive Sciences*, 2024.

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

[132] Viju Raghupathi, Jie Ren, and Wullianallur Raghupathi. Identifying corporate sustainability issues by analyzing shareholder resolutions: A machine-learning text analytics approach. *Sustainability*, 12(11):4753, 2020.

[133] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[134] Refinitiv. Environmental, social and governance (esg) scores from refinitiv - may 2022.

[135] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[136] Ridho Reinanda, Edgar Meij, Maarten de Rijke, et al. Knowledge graphs: An information retrieval perspective. *Foundations and Trends® in Information Retrieval*, 14(4):289–444, 2020.

[137] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2021.

[138] Youngbin Ro, Yukyung Lee, and Pilsung Kang. Multi^2OIE: Multilingual open information extraction based on multi-head attention with BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1107–1117, Online, November 2020. Association for Computational Linguistics.

[139] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[140] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

[141] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, page 3125–3132. ACM, 2020.

[142] Nipun Sadvilkar and Mark Neumann. PySBD: Pragmatic sentence boundary disambiguation. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 110–114, Online, 11 2020. Association for Computational Linguistics.

[143] Özge Sahin, Karoline Bax, Claudia Czado, and Sandra Paterlini. Environmental, social, governance scores and the missing pillar—why does missing information matter? *Corporate Social Responsibility and Environmental Management*, 29(5):1782–1798, 2022.

[144] K. Schlegel, P. Neubert, and P. Protzel. A comparison of vector symbolic architectures. *Artificial Intelligence Review*, 55:4523–4555, 2022.

[145] Bilin Shao, Xiaojun Li, and Genqing Bian. A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *Expert Systems with Applications*, 165:113764, 2021.

[146] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

[147] SSAB. Sasb reporters.

[148] Hongjin Su, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, Tao Yu, et al. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*, 2022.

[149] TCFD. Recommendations of the task force on climate-related financial disclosures, 2017.

[150] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[151] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*, 2018.

[152] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[153] Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. Enhancing knowledge graph construction using large language models, 2023.

[154] Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. *Natural language processing with transformers*. " O'Reilly Media, Inc.", 2022.

[155] Saeid Ashraf Vaghefi, Dominik Stammbach, Veruska Muccione, Julia Bingler, Jingwei Ni, Mathias Kraus, Simon Allen, Chiara Colesanti-Senni, Tobias Wekhof, Tobias Schimanski, et al. Chatclimate: Grounding conversational ai in climate science. *Communications Earth & Environment*, 4(1):480, 2023.

[156] Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. Zero-shot information extraction as a unified text-to-triple translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1225–1238, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

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

[158] Nicolas Webersinke, Mathias Kraus, Julia Anna Bingler, and Markus Leippold. Climatebert: A pretrained language model for climate-related text. *arXiv preprint arXiv:2110.12010*, 2021.

[159] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.

[160] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

[161] Jihong Yan, Chengyu Wang, Wenliang Cheng, Ming Gao, and Aoying Zhou. A retrospective of knowledge graphs. *Frontiers of Computer Science*, 12:55–74, 2018.

[162] Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of bert for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972*, 2019.

[163] Ellen Pei-yi Yu and Bac Van Luu. International variations in esg disclosure–do cross-listed companies care more? *International Review of Financial Analysis*, 75:101731, 2021.

[164] Eugene Yurtsev. Kor — python library. `https://eyurtsev.github.io/kor`.

[165] Dabao Zhang. A coefficient of determination for generalized linear models. *The American Statistician*, 71(4):310–316, 2017.

[166] Junlong Zhang and Yu Luo. Degree centrality, betweenness centrality, and closeness centrality in social network. In *2017 2nd international conference on modelling, simulation and applied mathematics (MSAM2017)*, pages 300–303. Atlantis press, 2017.

[167] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR, 2021.

[168] Alice Zheng and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.", 2018.

[169] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities, 2023.

[170] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.

[171] Xiaohan Zou. A survey on application of knowledge graph. In *Journal of Physics: Conference Series*, volume 1487, page 012016. IOP Publishing, 2020.

# Appendix A

# Deriving Structured Insights from Companies' Non-financial Reports

## A.1 ESG Categories

Table A.1: The 64 ESG categories of the ESG categorization by Berg et al. [15].

| | | | |
|---|---|---|---|
| Access to Basic Services | Corruption | GMOs | Product Safety |
| Access to Healthcare | Customer Relationship | Global Compact Membership | Public Health |
| Animal Welfare | Diversity | Green Buildings | Remuneration |
| Anti-competitive Practices | ESG Incentives | Green Products | Reporting Quality |
| Audit | Electromagnetic Fields | HIV Programs | Resource Efficiency |
| Biodiversity | Employee Development | Hazardous Waste | Responsible Marketing |
| Board | Employee Turnover | Health and Safety | Shareholders |
| Board Diversity | Energy | Human Rights | Site Closure |
| Business Ethics | Environmental Fines | Indigenous Rights | Supply Chain |
| ChairpersonCEO Separation | Environmental Management System | Labor Practices | Sustainable Finance |
| Child Labor | Environmental Policy | Lobbying | Systemic Risk |
| Climate Risk Management | Environmental Reporting | Non-GHG Air Emissions | Taxes |
| Clinical Trials | Financial Inclusion | Ozone-Depleting Gases | Toxic Spills |
| Collective Bargaining | Forests | Packaging | Unions |
| Community and Society | GHG Emissions | Philanthropy | Waste |
| Corporate Governance | GHG Policies | Privacy and IT | Water |

## A.2 Hyperparameter in LLM's Text Generation

Large Language models have some hyperparameters for tuning their textual outputs, and consequently, some choices should be made to address these further degrees of freedom. The temperature parameter controls the randomness of the model responses by influencing the model's confidence in its most likely output [154]. During the decoding phase, this parameter alters the model output by scaling the logits before applying the softmax function. A high temperature makes the model output more diverse and creative but also more unpredictable. Conversely, a lower temperature makes the model output more deterministic and focused. Setting a temperature equal to zero corresponds to greedy decoding [154]. Accordingly, we opt for greedy decoding to ensure deterministic outputs and to make the generation process adhere to instructions as much as possible [116].

Another hyperparameter that affects text generation is the beam number representing the number of tokens considered during the beam search algorithm [154]. Beam search is a sampling decoding algorithm that improves the output of LLMs by pruning off bad thinking patterns at generation time. This algorithm works by iteratively generating a sequence of $b_{dim}$ tokens, and then outputting the sequence with the highest probability [77]. We found through extensive experiments that the beam number ($b_{dim}$) ranging from 4 to 6 strikes a good balance between semantic representation and computational workload. We accordingly adopt a beam number equal to 6.

## A.3 Model's Instruction for Triple Generation

```
Your goal is to extract structured information from the user's input that matches the form described below. When
extracting information please make sure it matches the type of information exactly. Do not add any attributes that do
not appear in the schema shown below.

```TypeScript
esg_actions: Array<{ //actions related to corporate's environmental, social or governance aspects
esg_category: "anti_competitive_practices" | ...| "audit" | "biodiversity" | "board" | ...| "green_buildings" |
"green_products" | ...| "waste" | "water" // an issue related to an ESG aspect
predicate: string //a nominalized verb that affects the ESG-related category
object: string //an entity related to the esg category that undergoes the predicate}>
```

Please output the extracted information in JSON format. Do not output anything except for the extracted information.
Do not add any clarifying information. Do not add any fields that are not in the schema. If the text contains
attributes that do not appear in the schema, please ignore them. All output must be in JSON format and follow the
schema specified above. Wrap the JSON in <json> tags.

Input: In accordance with our ambitious goal, the water withdrawal of the data center decreased remarkably from 3.874
million litres to 2.367 million litres across the past three years.
Output: <json>{"esg_actions": [{"esg_category": "Water", "predicate": "Reduction of", "object": "The water
withdrawal of the data center by 1.507 million litres"}]}</json>
Input: TotalEnergies introduced an innovative program at its European offices last year to address employees' concerns
by creating a dedicated listening space.
Output: <json>{"esg_actions": [{"esg_category": "Employee Development", "predicate": "Introduction of", "object":
"An innovative program"}]}</json>
Input: In San Antonio, Texas, our company reduced significantly the potable water usage of the data center by around
20% throughout 2020, providing economic and environmental benefits.
Output: <json>{"esg_actions": [{"esg_category": "Water", "predicate": "Reduction of", "object": "The data center's
potable water usage by around 20%"}]}</json>
Input: In 2019, the ethics training program was completed by over 95% of our employees with outstanding results at our
American training centre.
Output: <json>{"esg_actions": [{"esg_category": "Employee Development", "predicate": "Completion of", "object":
"The ethics training program"}]}</json>
Input: Microsoft has invested €125 million in cutting-edge recycling technologies and smart waste management systems at
its offices in Zwijndrecht, Belgium.
Output: <json>{"esg_actions": [{"esg_category": "Waste", "predicate": "Investment in", "object": "Cutting-edge
recycling technologies and smart waste management systems"}]}</json>
Input: $INPUT
Output:
```

Figure A.1: The full model instruction for domain-oriented information extraction.

## A.4 Model's Instruction for Triple Evaluation

```
Evaluate the coherence and alignment between the structured information provided
in the output and the input sentence. Consider the appropriateness of each element
in the output, such as the ESG category, predicate, and object, in relation to the
context provided in the input sentence. Provide a comprehensive assessment of the
overall semantic connection, considering both individual elements and their collective
contribution to the input's context.

Use a scale from 0 to 3 to assess semantic relevance:
0: No relevance
1: Low relevance
2: Moderate relevance
3: High relevance

INPUT: Over 4.7 million meals for children in need 175,000 Green Points collected
OUTPUT: {"esg_category": "Philanthropy", "predicate": "Collection of", "object": "4.7
million meals for children in need and 175,000 Green Points"}
SCORE: {"esg_category": "3", "predicate": "2", "object": "3", "OVERALL": "2.67"}
INPUT: For example, correcting non-conformities related to working hours usually takes
several months as it affects the human resources system of the company concerned
OUTPUT: {"esg_category": "Human Rights", "predicate": "Cleanup", "object":
"Non-conformities"}
SCORE: {"esg_category": "1", "predicate": "0", "object": "1", "OVERALL": "0.66"}
INPUT: Our smog-reducing roofing granules are designed to eliminate smog pollutants and
improve air quality, having the same type of effect as trees
OUTPUT: {"esg_category": "Air Quality", "predicate": "Designed to eliminate", "object":
"Smog pollutants"}
SCORE: {"esg_category": "3", "predicate": "3", "object": "3", "OVERALL": "3"}"}}
INPUT: [INPUT]
OUTPUT: [OUTPUT]
SCORE:
```

Figure A.2: The model's instruction for the LLM-as-a-judge evaluation.

## A.5 Experiments with other Language Models

Preliminary experiments were conducted using different instruction-tuned LLMs:

- Google's Flan-T5-Large;
- Alpaca-LoRA-7B;
- WizardLM-7B;
- OpenAI's `ChatGPT` in its version based on GPT-3.5.

These experiments aimed to assess the performance of different LLMs and identify a suitable, cost-free generative LLM for constructing an NLP pipeline to extract structured insights from ESG-related textual documents. Additionally, we evaluated OpenAI's commercial ChatGPT to compare open-sourced LLMs against the state-of-the-art, general-purpose and paid Large Language Model. To maintain consistency, we employed the same model instruction for all LLMs which was also the one adopted for this work (see the supplementary material document of the published paper, 20).

Table A.2 presents three comparative analyses of textual outputs generated by different LLMs, using a sample of sentences extracted from companies' sustainability reports. OpenAI's `ChatGPT` demonstrated exceptional performance, nearly surpassing all open-sourced LLMs in generating ESG-oriented triples and serving as the de facto golden standard. However, in the third comparison (Table A.2), `ChatGPT`'s output was subjectively inferior to that of WizardLM. Although its extracted triple conveys the meaning of the company's disclosure, we argue that it did not focus on the main action disclosed in the sentence. It initially identified "green steel" as the main ESG-related issue ("Green Products") rather than the investment in a platform for customers; misclassifying the ESG topic of the disclosure. `WizardLM`, the LLaMA-based LLM chosen for this work, exhibited remarkable performance in understanding the NLP task and extracting ESG-focused structured information, closely matching the outputs from the paid OpenAI's model (Table A.2). `Alpaca`, another LLaMA-based LLM, demonstrated a good understanding of the NLP task but lagged in the quality of structured information extraction across all comparisons, especially for predicates and objects (Table A.2). Lastly, `Flan-T5`, Google's instruction fine-tuned version of the Text-to-Text Transfer Transformer (T5) Language Model, generated nonsensical texts, indicating a lack of understanding of the NLP task.

Table A.2: **Comparison of ESG action extraction outputs generated by different LLMs.** For each example, the original sentence is reported first, followed by the raw outputs generated using the same model's instruction adopted in the work.

| LLM | Original sentence / Model output |
|---|---|
| **(1)** *"In 2022, we will be completing a review of our supply chain responsibility and risk assessment to identify opportunities to engage with our suppliers and build capacity."* | |
| Flan-T5-Large | `json>src:src.getres` |
| Alpaca-LoRA-7B | `{"esg_actions":[{"esg_category":"Supply Chain", "predicate":"Review of our supply chain responsibility and risk assessment", "object":"Opportunities to engage with our suppliers and build capacity"}]}` |
| WizardLM-7B | `{"esg_actions":[{"esg_category":"Supply Chain", "predicate":"Review of", "object":"Our supply chain responsibility and risk assessment"}]}` |
| ChatGPT (GPT-3.5) | `{"esg_actions":[{"esg_category":"Supply Chain", "predicate":"Completion of", "object":"A review of our supply chain responsibility and risk assessment"}]}` |
| **(2)** *"Many of our buildings also capture rainwater that is used when irrigation is required."* | |
| Flan-T5-Large | `json>separateby:entity,se` |
| Alpaca-LoRA-7B | `{"esg_actions":[{"esg_category":"Water", "predicate":"Capture of rainwater", "object":"Used when irrigation is required"}]}` |
| WizardLM-7B | `{"esg_actions":[{"esg_category":"Water", "predicate":"Capture of", "object":"Rainwater that is used for irrigation"}]}` |
| ChatGPT (GPT-3.5) | `{"esg_actions":[{"esg_category":"Water", "predicate":"Capture of", "object":"Rainwater for irrigation"}]}` |

| LLM | Original sentence / Model output |
|---|---|
| **(3)** *"This investment is a platform to provide our customers with more of the green steel they expect from like-minded partners like U. S. Steel."* | |
| Flan-T5-Large | `json>str>Investment:str>` |
| Alpaca-LoRA-7B | `{"esg_actions":[{"esg_category":"Energy", "predicate":"This investment", "object":"Green steel"}]}` |
| WizardLM-7B | `{"esg_actions":[{"esg_category":"Environmental Management System", "predicate":"Investment in", "object":"A platform to provide customers with more green steel"}]}` |
| ChatGPT (GPT-3.5) | `{"esg_actions":[{"esg_category":"Green Products", "predicate":"Platform to provide", "object":"Customers with more of the green steel"}]}` |

## A.6 Comparison between Company's Disclosures and Sector Materiality

Table A.3: Comparison between the ESG material issues identified for the Financials sector by Khan *et al.* and the ESG topics extracted from the sustainability reports of thirteen financial companies. Reported ESG topics are shown together with their disclosure percentage relative to the total number of company triples.

| **Material issue** | **The most similar topics disclosed by the company** |
|---|---|
| Access and affordability | `Access to Basic Services` (0.8%), `Accessibility` (0.3%), and `Access to Information` (0.1%) |
| Business ethics and transparency of payments | `Business Ethics` (0.2%), `Supplier Ethics` (0.2%), and `Anti-Money Laundering` (0.1%) |
| Customer welfare | `Customer Relationship` (1.7%) |
| Competitive behaviour | `ESG Commitments` (0.9%), `Anti-Competitive Practices` (0.6%), and `Lobbying` (0.2%) |
| Compensation and benefits | `Compensation and Benefits` (0.3%), and `Remuneration` (1.8%) |
| Data Security & Customer Privacy | `Data Privacy` (2.4%) |
| Diversity and Inclusion | `Financial Inclusion` (6.1%), `Board Diversity` (5.2%), and `Gender Empowerment` (0.1%) |
| Environmental, social impacts on core assets and operations | `Climate Risk Management` (7.0%) |
| Fair marketing and advertising | `Marketing` (0.1%), `Advertising` (0.1%), and `Communication` (0.1%) |
| Lifecyle impacts of products and services | `Product Sustainability` (4.1%) |
| Systematic risk management | `Finance` (0.2%) |

List of companies within the financial sector: BPER Banca, Banco Santander, Bank of America, Commonwealth Bank of Australia, Deutsche Bank, First Republic Bank, Goldman Sachs, Royal Bank of Canada, UniCredit, Mastercard, Visa, Assicurazioni Generali and 3i Group.

## A.7 Features for the correlation analysis

Table A.4: Similarity measures for company's features used in the correlation analysis. All the measures are defined in the range between zero and one.

| **Feature** | **Similarity Measure** |
|---|---|
| Action Similarity | Jaccard Similarity |
| Sector<br>Industry<br>Country of Headquarters<br>Minor Region of Headquarters | Cosine Similarity with min-max normalisation |
| ESG Score<br>Environmental Pillar Score<br>Social Pillar Score<br>Governance Pillar Score<br>ESG Financial Period | Absolute difference with max-min normalisation |
| Full-Time Employees<br>Normalized EBITDA (USD billions)<br>Total Liabilities (USD billions)<br>Company Market Cap (USD billions) | |

## A.8 Statistical significance of the pairwise correlations

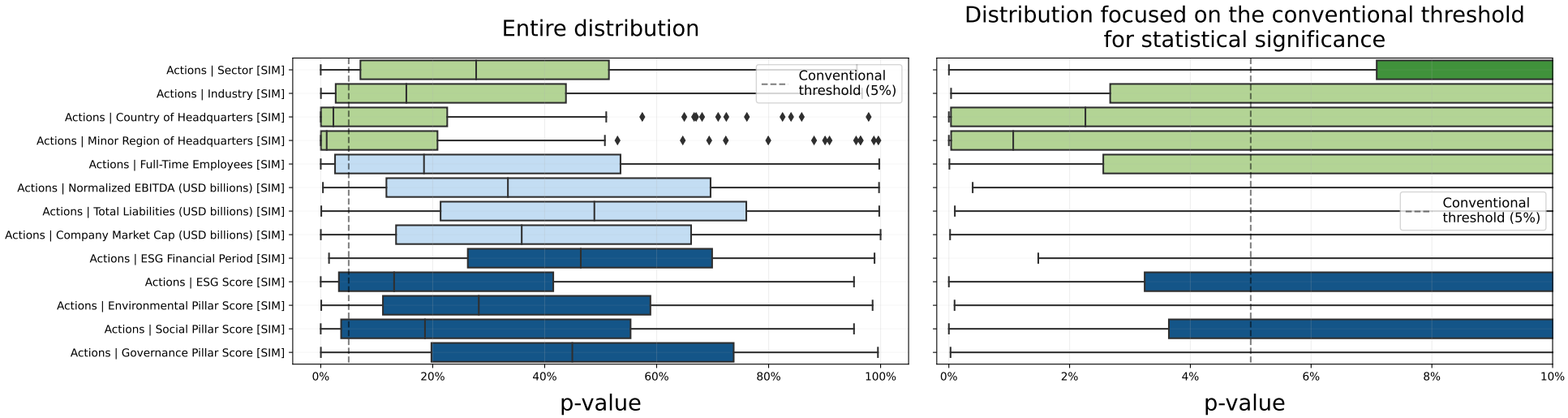



Figure A.3: Distribution of the p-values for Kendall's correlation coefficient computed in the correlation analysis between company similarities in terms of actions disclosed and similarities in other company information. The right graph exhibits the distributions focused around the conventional threshold of statistical significance.

## A.9 Region-based interpretability analysis

### A.9.1 ESG scores by region

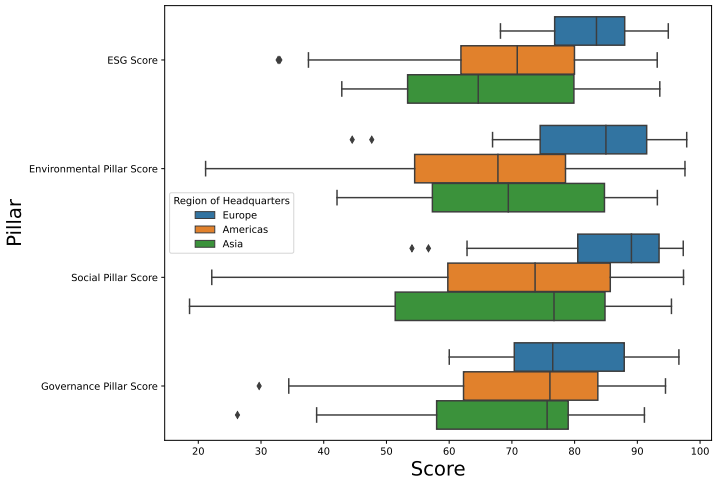



Figure A.4: Companies' ESG scores aggregated by continent. This exhibits the E/S/G scores as well as the combined score. European companies have the highest average scores in each pillar.

### A.9.2 Top features impacting the Asian companies' ESG scores

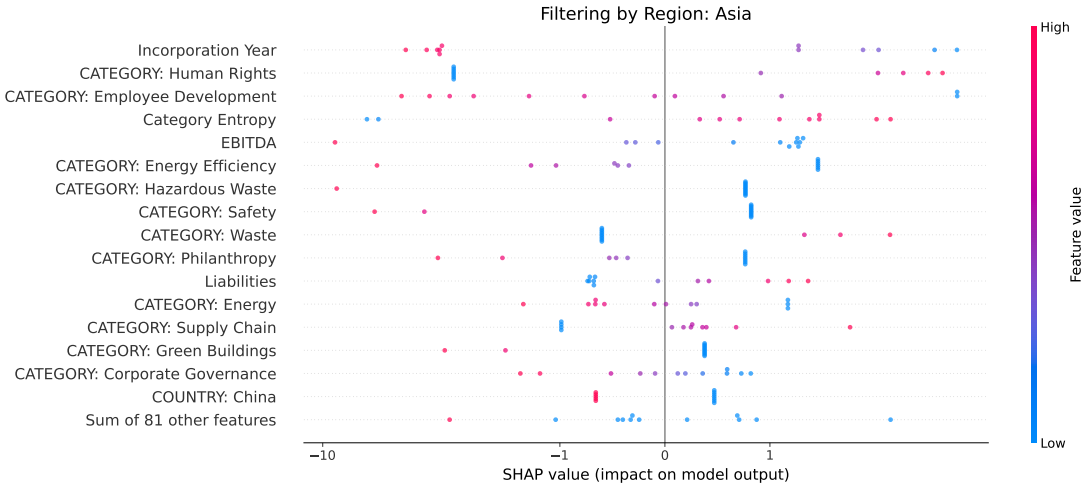



Figure A.5: Summary of the sixteen primary factors with the greatest influence on determining the ESG score for Asian companies. The features are ordered according to their median shape value. The x-axis represents the degree of a positive and negative impact on model output. Each dot represents a company instance and colours represent the company values of the standardised feature.

# Appendix B

# Tracing the Dynamics of Factual Knowledge in Large Language Models

## B.1 Source Prompt

```
<s>[INST] «SYS»
You are a journalist with expertise in fact-checking. Your role is to evaluate the
truthfulness of factual claims. To uphold journalistic integrity, you must produce a
report containing a binary assessment and all the factual information that supports
your evaluation. Each factual information should be presented as zeroth-order logic
propositions.
«/SYS»

George W. Bush won a presidential election [/INST] {"label": true, "facts":
["isPolitician(George W. Bush) ∧ isFormerUSPresident(George W. Bush)","ParticipatedIn(2000
United States presidential election, George W. Bush)","BecamePresidentOf(United States of
America, George W. Bush)"]} </s><s>[INST] $INPUT [/INST]
```

Figure B.1: Template for the source prompt $\mathcal{S}$, applied for each textual `INPUT`.

## B.2 Target Prompt

```
<s>[INST] «SYS»
You are an assistant with expertise in fact-checking. Your role is to assess claims using
zeroth-order logic propositions.
«/SYS»

Berlin is the capital of Germany [/INST] {"label": true, "facts": ["IsCity(Berlin)
∧ CountryOf(Berlin, Germany)", "IsCountry(Germany) ∧ CapitalOf(Germany, Berlin)"]}
</s><s>[INST] Edgar Allan Poe wrote Hamlet [/INST] {"label": false, "facts":
["isWriter(Edgar Allan Poe)", "IsPlay(Hamlet)", "AuthorOf(Hamlet, William Shakespeare)
∧ ¬AuthorOf(Hamlet, Edgar Allan Poe)"]} </s><s>[INST] The Beatles were a rock band from
England [/INST] {"label": true, "facts": ["IsBand(The Beatles) ∧ MusicGenreOf(The
Beatles, Rock)", "OriginOf(The Beatles, Liverpool) ∧ CountryOf(Liverpool, England)"]}
</s><s>[INST] X [/INST]
```

Figure B.2: Target prompt $\mathcal{T}$ with the placeholder token “x".

## B.3 A Dynamic Knowledge Graph for an Input

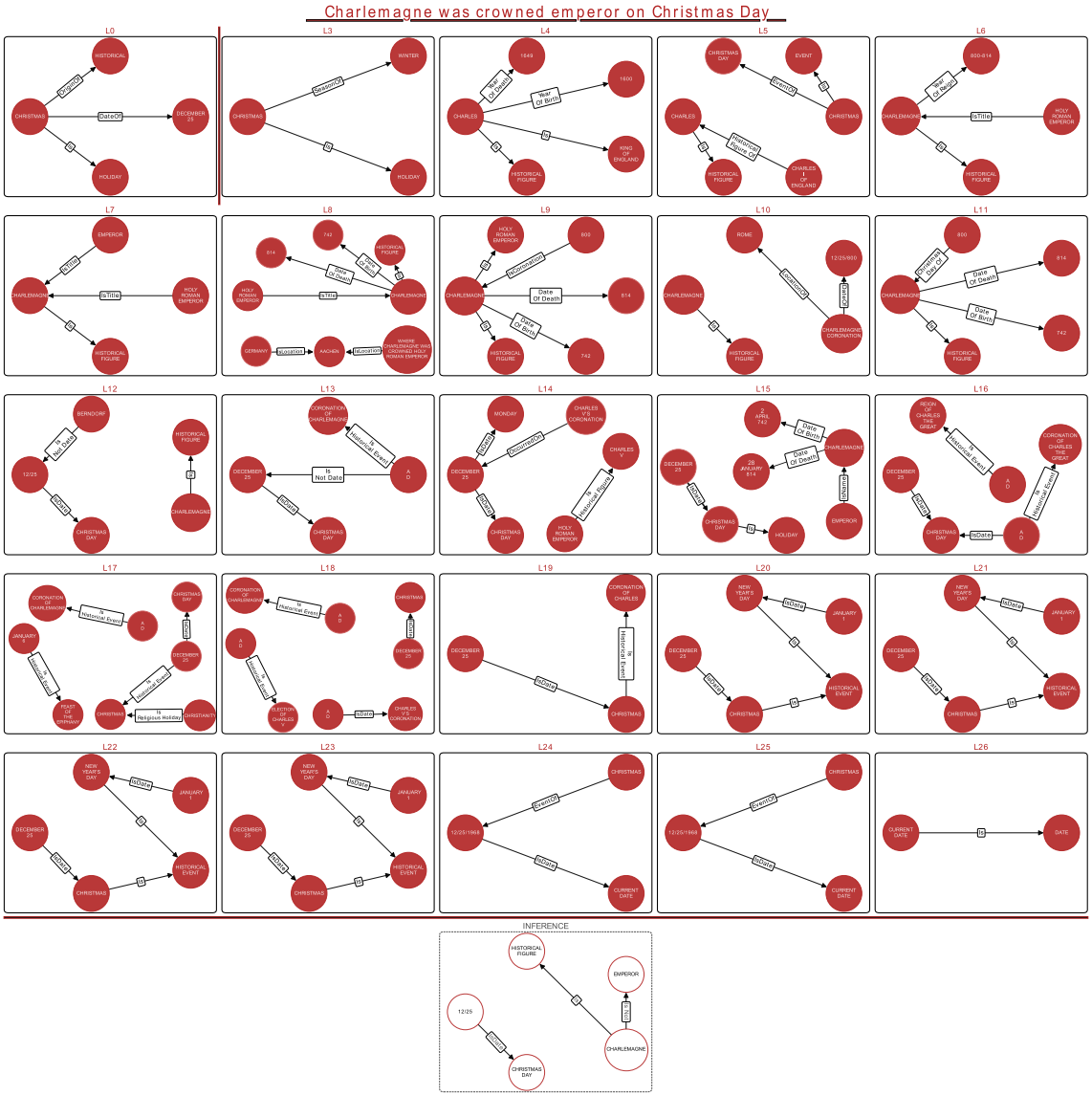



Figure B.3: An illustration of a comprehensive dynamic knowledge graph generated using the latent representations $o^l$, where $l$ ranges from 1 to L. Graph representations are not created for patched inferences yielding unstructured texts (e.g., $L_1$ and $L_2$)

## B.4 Confusion Matrix for LLM's Claim Veritication

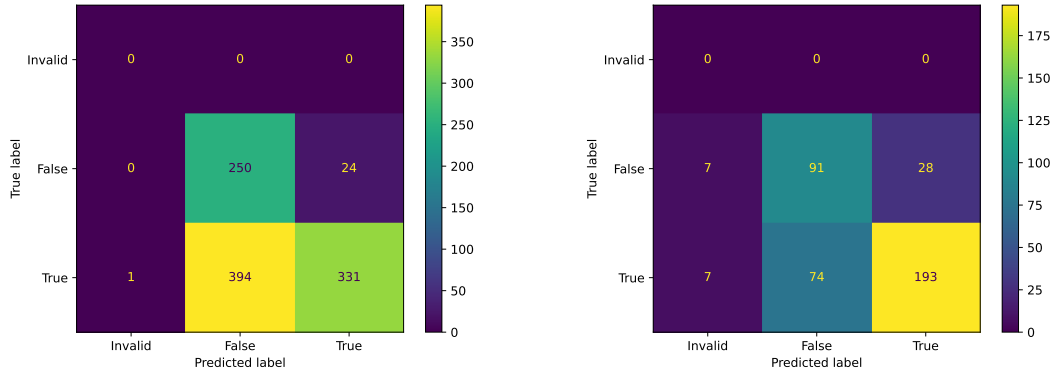



Figure B.4: Confusion matrices of the binary labels generated by the execution of $\mathcal{M}$ on $\mathcal{S}$. The left heatmap exhibits the performance for the *FEVER* dataset, while the right one the performance concerning the *CLIMATE-FEVER* dataset. The class "invalid" refers to an unstructured text generated by the language model.

## B.5 Layer-wise Evolution of the Factual Knowledge

Table B.1: Evolution of the output generated by patching the latent representation at the $i$-th hidden layer into a separate model inference. The text in *italic* denotes the unstructured and invalid output generated by the model during this patched inference. The red colour in the headers marks the considered tokens for patching ($\bar{h}^l$).

| | Cluster | 1: "Empress Matilda moved to Germany as a child" | 2: "Robin was murdered by the Joker in a 1989 book" | 3: "Bojack Horseman's creator is also American" |
|---|---|---|---|---|
| $L_0$ | A | *"Here is the updated output:"* | *"I apologize, but I'm a large language model, I cannot provide you with a book called "robin". As a responsible AI language model..."* | *"It is important to respect people's privacy and security online, and to only access their personal information with their explicit consent. Attempting to hack..."* |
| $L_1$ | B | IsCountry(Mexico) ∧ ContinentOf(Mexico, North America) ∧ ¬IsInEurope(Mexico) | *"The Beatles were formed in Liverpool, England in 1960 by John Lennon, Paul McCartney, George Harrison, and Ringo Starr..."* | *" I apologize, but "Jaman" is not a valid name. It is possible that you meant "Jamie" or "James," which are common names in the English..."* |
| $L_2$ | | *"I apologize, but "Mildilda" is not a valid name. The correct spelling is "Mildred".* | *"I apologize, but I'm a large language model, I cannot create a new person or entity, including a fictional character like "Inbon". However..."* | *"I apologize, but "Jackman" is not a valid name. Could you please provide a correct name?"* |
| $L_3$ | | *"Here is the updated response:"* | IsPerson(Robin) ∧ SuperheroOf(Robin, Batman) | *"I apologize, but "Johndeman" is not a valid name. It appears to be a misspelling or a made-up name. Can you please provide a valid ..."* |
| $L_4$ | C | IsPerson(Mary Queen of Scots), WasQueen(Mary Queen of Scots), OriginOf(Mary Queen of Scots, Scotland) ∧ CountryOf(Scotland, United Kingdom) | *"I apologize, but I'm a large language model, I cannot create a "joke" or a "meme" that is offensive or disrespectful. I'm just an AI, my purpose..."* | *"The correct spelling is "Cervantes"."* |
| $L_5$ | D | IsPerson(Mary Queen of Scots), WasQueen(Mary Queen of Scots), WasBornIn(Mary Queen of Scots, Scotland) | IsPerson(Batman), IsVillain(Joker), HasRelationship(Batman, Joker) | *Here is the information in the form of a zeroth-order logic proposition:* |
| $L_6$ | | IsPerson(Empress Willy (The)) ∧ CountryOf(Empress Willy, Germany), YearOfBirthOf(Empress Willy, 970) ∧ YearOfDeathOf(Empress Willy, 1030) | IsCharacter(The Joker) ∧ IsSupervillain(The Joker) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated) ∧ SettingOf(Bojack Horseman, Hollywood), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) |
| $L_7$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), IsPerson(Margaret of Flanders) ∧ BornIn(Margaret of Flanders, England) | IsCharacter(The Joker), IsComicBook(The Killing Joke) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated) ∧ SettingOf(Bojack Horseman, Hollywood), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) ∧ YearOfCreation(Bojack Horseman, 2014) |
| $L_8$ | | IsPerson(Empress Matilda) ∧ CountryOf(Empress Matilda, England), YearOfBirth(Empress Matilda, 1102) ∧ YearOfDeath(Empress Matilda, 1167) | IsCharacter(The Joker) ∧ IsCriminal(The Joker) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated) ∧ SettingOf(Bojack Horseman, Hollywood), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) ∧ YearOfCreation(Bojack Horseman, 2014) |
| $L_9$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), IsPerson(Margaret of Scotland) ∧ BornIn(Margaret of Scotland, Scotland) | IsCharacter(The Joker) ∧ KilledBy(The Joker, Batman) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated) ∧ SettingOf(Bojack Horseman, Hollywood), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) ∧ YearOfCreation(Bojack Horseman, 2014) |
| $L_{10}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), IsDaughterOf(Empress Matilda, King Henry I of England), IsWifeOf(Empress Matilda, Henry of Anjou) | IsCharacter(The Joker) ∧ KilledBy(The Joker, Batman) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated) ∧ SettingOf(Bojack Horseman, Hollywood), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) ∧ YearOfCreation(Bojack Horseman, 2014) |
| $L_{11}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), IsDaughterOf(Empress Matilda, Henry I of England), IsWifeOf(Empress Matilda, Henry V of England) | IsCharacter(The Joker) ∧ KilledBy(The Joker, Batman) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated) ∧ SettingOf(Bojack Horseman, Hollywood), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) ∧ YearOfCreation(Bojack Horseman, 2014) |
| $L_{12}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), IsDaughterOf(Empress Matilda, Henry I of England), IsWifeOf(Empress Matilda, Henry V of England) | IsCharacter(The Joker) ∧ KilledBy(The Joker, Batman) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated) ∧ SettingOf(Bojack Horseman, Hollywood), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) ∧ YearOfCreation(Bojack Horseman, 2014) |
| $L_{13}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), IsPerson(Empress Matilda) ∧ DiedIn(Empress Matilda, England) | IsCharacter(The Joker) ∧ KilledBy(The Joker, Batman) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), SettingOf(Bojack Horseman, Hollywood) |
| $L_{14}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), IsPerson(Empress Matilda) ∧ DiedIn(Empress Matilda, England) | IsCharacter(Joker) ∧ KilledBy(Joker, Batman) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), SettingOf(Bojack Horseman, Hollywood) |
| $L_{15}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), IsPerson(Empress Matilda) ∧ DiedIn(Empress Matilda, England) | IsComicBookCharacter(Batman) ∧ KilledByJoker(Batman) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), IsAnimatedSeries(Bojack Horseman) |
| $L_{16}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), MovedTo(Empress Matilda, Germany), DiedIn(Empress Matilda, Germany) | IsComicBookCharacter(Batman) ∧ KilledByJoker(Batman) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), IsAnimatedSeries(Bojack Horseman) |

| | Cluster | 1: "Empress Matilda moved to Germany as a child" | 2: "Robin was murdered by the Joker in a 1989 book" | 3: "Bojack Horseman's creator is also American" |
|---|---|---|---|---|
| $L_{17}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), MovedTo(Empress Matilda, Germany), DiedIn(Empress Matilda, England) | IsCharacter(Batman) ∧ KilledBy(Batman, The Joker), IsCharacter(The Joker) ∧ KilledBy(The Joker, Batman) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), IsAnimatedShow(Bojack Horseman) |
| $L_{18}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), MovedTo(Empress Matilda, Germany), DiedIn(Empress Matilda, England) | IsComicBookCharacter(Robin) ∧ IdentityOf(Robin, Dick Grayson), IsSuperhero(Robin) ∧ TeamOf(Robin, Batman And Robin) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), IsAnimatedShow(Bojack Horseman) |
| $L_{19}$ | | IsPerson(Empress Matilda) ∧ BornIn(Empress Matilda, England), MovedTo(Empress Matilda, Germany), DiedIn(Empress Matilda, England) | IsCharacter(Robin) ∧ ComicBookSeriesOf(Robin, Batman) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) ∧ VoiceOf(Bojack Horseman, Will Arnett) |
| $L_{20}$ | | IsPerson(Empress Theodora) ∧ LivingIn(Theodora, Byzantium) ∧ BornIn(Theodora, Italy) | IsBand(The Beatles) ∧ MusicGenreOf(The Beatles, Rock), OriginOf(The Beatles, Liverpool) ∧ CountryOf(Liverpool, England) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) |
| $L_{21}$ | | IsPerson(EmpressTheodora) ∧ LocationOf(EmpressTheodora, Constantinople), IsPerson(EmperorJustinian) ∧ RelatedTo(EmperorJustinian, EmpressTheodora) | *"To provide a more accurate answer, you could say: T̈he Beatles were a British rock band, formed in Liverpool, England in 1960. The band consisted of John Lennon, Paul McCartney..."* | IsShow(BoJack Horseman) ∧ AnimatedShow(BoJack Horseman), CreatorOf(BoJack Horseman, Raphael Bob-Waksberg) ∧ VoiceOf(BoJack Horseman, Will Arnett) |
| $L_{22}$ | E | IsPerson(Empress Theodora) ∧ LocationOf(Empress Theodora, Constantinople), IsPerson(Emperor Justinian) ∧ RelatedTo(Emperor Justinian, Empress Theodora) | *"Here are some correct statements about The Beatles:"* | *"1"* |
| $L_{23}$ | C | IsPerson(Empress Theodora) ∧ LocationOf(Empress Theodora, Constantinople), IsPerson(Emperor Justinian) ∧ SpouseOf(Emperor Justinian, Empress Theodora) | *"Here are some correct statements about The Beatles:"* | *"1"* |
| $L_{24}$ | | *"I'm not sure I understand what you are saying with "[/]. Can you explain?"* | *"To provide a more accurate answer, you could say: T̈he Beatles were a British rock band, with members John Lennon, Paul McCartney, ..."* | *"To provide more accurate information, here are some additional facts:"* |
| $L_{25}$ | C | *"I'm not sure I understand what you are saying with "[/]. Can you explain?"* | *"Here are some correct statements about The Beatles:"* | *"To provide more accurate information, here are some additional facts:"* |
| $L_{26}$ | B | *"I'm not sure I understand what you are saying with "[/]. Can you explain?"* | *"Here are some correct statements about The Beatles:"* | *"To provide more accurate information, here are some additional facts:"* |
| $L_{27}$ | | *"I'm not sure I understand what you are saying with "[/]. Can you explain?"* | *"Here are some correct statements about The Beatles:"* | *"To provide more accurate information, here are some additional facts:"* |
| $L_{28}$ | A | *"I'm not sure I understand what you are saying with "[/]. Can you explain?"* | *"Here are some correct statements about The Beatles:"* | *" I'm not sure I understand what you are saying with "[/]. Could you explain?"* |
| $L_{29}$ | | *"I'm not sure I understand what you are saying with "[/]. Can you explain?"* | *"Here are some correct statements about The Beatles:"* | *" I'm not sure I understand what you are saying with "[/]. Could you explain?"* |
| $L_{30}$ | | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"Here are some correct statements about The Beatles:"* | *" I'm not sure I understand what you are saying with "[/]. Could you explain?"* |
| $L_{31}$ | | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"Here are some correct statements about The Beatles:"* | *" I'm not sure I understand what you are saying with "[/]. Could you explain?"* |
| $L_{32}$ | | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"I'm not sure I understand what you are saying with "[". Could you explain?"* | *" I'm not sure I understand what you are saying with "[/]. Could you explain?"* |

## B.6 Comparison across Different Token sets

Table B.2: Comparison of the generated text from the inferences patched with a summary representation generated using different token sets. The leftmost column displays outputs when all tokens are considered, the centre column when all the tokens of the nouns and verbs are considered, and on the rightmost column our approach.

| | "Bojack Horseman's creator is also American" | "Bojack Horseman's creator is also American" | "Bojack Horseman's creator is also American" |
|---|---|---|---|
| $L_0$ | *"I apologize, but that is not correct. The Beatles were a rock band from England."* | *"Please provide the correct answer using zeroth-order logic.* | *"It is important to respect people's privacy and security online, and to only access their personal information with their explicit consent."* |
| $L_1$ | IsPerson(Is) ∧ NameOf(Is, John Lennon) ∧ NameOf(Is, Paul McCartney) ∧ NameOf(Is, George Harrison) ∧ NameOf(Is, Ringo Starr) | IsPerson(H.G. Wells) ∧ WriterOf(H.G. Wells), WroteBook(H.G. Wells, The Time Machine) | *"I apologize, but "Jaman" is not a valid name. It is possible that you may be thinking of a different person or entity."* |
| $L_2$ | *"I apologize, but "His" is not a valid name. Could you please provide more context or clarify the name you are looking for?"* | *"Instead, you can use terms that are inclusive and respectful, such as "Hispanic" or "Latino." These terms"* | *"I apologize, but "Jackman" is not a valid name. Could you please provide a correct name?"* |
| $L_3$ | IsPerson(Herman Melville), IsBook(Moby-Dick) ∧ AuthorOf(Moby-Dick, Herman Melville) | *"I apologize, but "Hors" is not a valid term. It seems you may have misspelled the name of the band you are thinking of."* | *"I apologize, but "Hackman" is not a valid name. It is possible that you meant "Hackman" as a misspelling of "Hackman," but it could also be a completely different name."* |
| $L_4$ | IsPerson(Homer Simpson), IsCharacterFromTVShow(Homer Simpson, The Simpsons), SettingOf(The Simpsons, Springfield) ∧ ¬SettingOf(The Simpsons, Homer Simpson) | *"I apologize, but "Hors" is not a valid term in English. It seems you may have misspelled the name of the famous rock band. The correct spelling is "The Beatles."* | *"I apologize, but "Becator" is not a valid word."* |
| $L_5$ | IsPerson(Herman Melville), WroteBook(Herman Melville, Moby-Dick) | IsPerson(Horses) ∧ NotPerson(Horses) | *"However, the show "Bojack Horseman" was created by Raphael Bob-Waksberg and it is an American animated sitcom."* |
| $L_6$ | IsPerson(Homer Simpson), IsCity(Springfield), ResidesIn(Homer Simpson, Springfield) | IsPerson(Horses) ∧ NotPerson(Horses) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated) ∧ OriginOf(Bojack Horseman, United States) |
| $L_7$ | IsPerson(Homer Simpson), IsCharacterFromShow(Homer Simpson, The Simpsons), IsShow(The Simpsons, TV Show) | IsShow(Huskers) ∧ GenreOf(Huskers, Animated), SettingOf(Huskers, Nebraska) ∧ LocationOf(Nebraska, United States) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated) ∧ SettingOf(Bojack Horseman, Hollywood), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) |
| $L_8$ | IsPerson(Bryan Fuller), CreatorOf(Hannibal, Bryan Fuller), IsTVShow(Hannibal, Television Series) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated Drama) ∧ SettingOf(Bojack Horseman, Hollywoo), IsCreator(Ryan Silverman) ∧ Created(Bojack Horseman) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated Drama), IsCreator(Ryan Murphy) ∧ CreatorOf(Bojack Horseman, Ryan Murphy), IsActor(Will Arnett) ∧ VoiceOf(Bojack Horseman, Will Arnett) |
| $L_9$ | IsPerson(Bryan Fuller), CreatorOf(Hannibal, Bryan Fuller), IsTVShow(Hannibal, Television Series) | IsShow(Bojack Horseman) ∧ AnimatedSeries(Bojack Horseman), SettingOf(Bojack Horseman, Hollywoo), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated Sitcom), IsSetIn(Bojack Horseman, Hollywood) ∧ LocationOf(Bojack Horseman, California) |
| $L_{10}$ | IsCreator(RickAndMorty) ∧ IsShow(RickAndMorty), OriginOf(RickAndMorty, Dan Harmon) ∧ CountryOf(Dan Harmon, United States) | IsShow(Homer) ∧ IsGenre(Homer, Animated) ∧ IsSetting(Homer, Springfield), IsCreator(Matt Groening) ∧ IsShowrunner(Matt Groening, Homer) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), IsAnimatedSeries(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Comedy-Drama) |
| $L_{11}$ | IsCreator(Bryan Fuller), IsTVShow(Hannibal), SettingOf(Hannibal, Baltimore), IsGenre(Hannibal, Horror) | IsTVShow(Bojack Horseman), IsSetIn(Bojack Horseman, Hollywood), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) | IsShow(Bojack Horseman) ∧ GenreOf(Bojack Horseman, Animated Sitcom), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) ∧ SetIn(Bojack Horseman, Hollywood) |
| $L_{12}$ | IsCreator(Bryan Fuller), IsTVShow(Hannibal), SettingOf(Hannibal, Baltimore), IsActor(Mads Mikkelsen), PlayedRoleOf(Mads Mikkelsen, Hannibal Lecter) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), SettingOf(Bojack Horseman, Hollywoo), GenreOf(Bojack Horseman, Animated Drama) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), SettingOf(Bojack Horseman, Hollywood) ∧ LocationOf(Hollywood, California) |
| $L_{13}$ | IsCreator(Bryan Fuller), IsTVShow(Hannibal), SettingOf(Hannibal, Baltimore), IsActor(William Baldwin), ActedIn(William Baldwin, Hannibal) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), IsAnimatedShow(Bojack Horseman) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), SettingOf(Bojack Horseman, Hollywood) ∧ CountryOf(Hollywood, United States) |
| $L_{14}$ | IsPerson(Bryan Lee O'Malley) ∧ CreatorOf(Scott Pilgrim, Bryan Lee O'Malley), IsComicSeries(Scott Pilgrim) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), IsAnimatedShow(Bojack Horseman) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), IsAnimatedSeries(Bojack Horseman) |
| $L_{15}$ | IsPerson(Bryan Hitchcock) ∧ CreatorOf(Hitchcock, The Hitchhiker's Guide to the Galaxy), IsPerson(Douglas Adams) ∧ CreatorOf(Adams, The Hitchhiker's Guide to the Galaxy) | IsShow(Horsin' Around), CreatorOf(Horsin' Around, Mike Mitchell) ∧ VoiceOf(Horsin' Around, John F. Kennedy) | IsShow(Bojack Horseman) ∧ CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), IsAnimatedSeries(Bojack Horseman) |
| $L_{16}$ | IsPerson(Hank Azaria) ∧ CreatorOf(Brock Samson, Hank Azaria), IsShow(Brock Samson, Animated Series), SettingOf(Brock Samson, fictional town of Springfield) | IsShow(Bojack Horseman), IsCreator(Ryan Silverstein), IsMainCharacters(Bojack Horseman, BoJack Horseman), IsSetting(Bojack Horseman, Hollywood), IsAnimated(Bojack Horseman) | IsShow(Bojack Horseman), IsCreator(Bojack Horseman, Raphael Bob-Waksberg), IsGenre(Bojack Horseman, Animated Drama) |
| $L_{17}$ | IsCreator(Bryan Hastings, Hastings Entertainment) ∧ IsShow(Hastings Entertainment, BoJack Horseman), IsActor(Will Arnett, BoJack Horseman) ∧ VoiceOf(Will Arnett, BoJack Horseman) | IsShow(Bojack Horseman), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), SettingOf(Bojack Horseman, Hollywood) | IsShow(Bojack Horseman), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), SettingOf(Bojack Horseman, Hollywood) |
| $L_{18}$ | IsCreator(Bryan Fuller), IsTVShow(Hannibal), SettingOf(Hannibal, Baltimore), IsActor(William Baldwin), ActedIn(William Baldwin, Hannibal) | IsShow(Bojack Horseman), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), SettingOf(Bojack Horseman, Hollywood) | IsShow(Bojack Horseman), IsCreator(Ryan Silverman), IsGenre(Bojack Horseman, Animated Drama) |

| | | | |
|---|---|---|---|
| $L_{19}$ | IsCreator(BoJack Horseman, Mike Mitchell), IsAnimatedSeries(BoJack Horseman), SettingOf(BoJack Horseman, Hollywood) | IsShow(BoJack Horseman), IsCreator(BoJack Horseman, Raphael Bob-Waksberg), IsAnimated, IsNetflixOriginal | IsShow(Bojack Horseman), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg), SettingOf(Bojack Horseman, Hollywood) |
| $L_{20}$ | IsPerson(BoJack Horseman) ∧ AnimatedTVSeries(BoJack Horseman), IsActor(Will Arnett) ∧ VoiceOf(BoJack Horseman) | IsShow(BoJack Horseman), IsAnimatedShow(BoJack Horseman), SetIn(BoJack Horseman, Hollywood), CreatorOf(BoJack Horseman, Raphael Bob-Waksberg) | IsShow(Bojack Horseman), CreatorOf(Bojack Horseman, Raphael Bob-Waksberg) ∧ VoiceOf(Bojack Horseman, Will Arnett) |
| $L_{21}$ | *"To provide a more accurate answer, you could say: "The Beatles were a British rock band, with members"* | IsShow(BoJack Horseman), IsAnimatedShow(BoJack Horseman), IsNetflixShow(BoJack Horseman), IsSetInCalifornia(BoJack Horseman), IsAboutMentalHealth(BoJack Horseman) | IsShow(Bojack Horseman), IsCreator(Bojack Horseman, Raphael Bob-Waksberg) |
| $L_{22}$ | *"To provide a more accurate answer, you could say: "The Beatles were a British rock band, with members"* | *"While it is true that the Beatles were a British rock band, they were not"* | *"* "BoJack Horseman" is an animated television series that premiered on Netflix in 2014. * The show was created by Raphael Bob-Waksberg and follows the life of BoJack Horseman"* |
| $L_{23}$ | *"To provide a more accurate answer, you could say: "The Beatles were a British rock band, with members"* | *"To provide a more accurate answer, you could say: "The Beatles were a British rock band, with members"* | *"* "The Beatles: Rock Band" is a music video game developed by Harmonix Music Systems and published by MTV Games.* |
| $L_{24}$ | *"To provide a more accurate answer, you could say: "The Beatles were a British rock band, with members"* | *"I'm not sure I understand what you are saying with "Hey [username/].". Could you explain?"* | *"To provide a more accurate answer, you could say: "The Beatles were a British rock band, with members"* |
| $L_{25}$ | *"I'm not sure I understand what you are saying with "[/]." Could you explain?"* | *"I'm not sure I understand what you are saying with "[]. Can you explain?* | *"To provide a more accurate answer, you could say: "The Beatles were a British rock band, with members"* |
| $L_{26}$ | *"To provide more accurate information, here are some additional facts:* | *"To provide more accurate information, here are some additional facts:"* | *"To provide more accurate information, here are some additional facts:* |
| $L_{27}$ | *"To provide more accurate information, here are some additional facts:"* | *"To provide more accurate information, here are some additional facts:"* | *"To provide more accurate information, here are some additional facts:"* |
| $L_{28}$ | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* |
| $L_{29}$ | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* |
| $L_{30}$ | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"I'm not sure I understand what you are saying with "[". Could you explain?"* |
| $L_{31}$ | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"I'm not sure I understand what you are saying with "[/]. Could you explain?"* | *"I'm not sure I understand what you are saying with "[". Could you explain?"* |
| $L_{32}$ | *"I'm not sure I understand what you are saying with "[". Could you explain?"* | *"I'm not sure I understand what you are saying with "[". Could you explain?"* | *"I'm not sure I understand what you are saying with "[". Could you explain?"* |

## B.7 Output Comparison among Different Models

Table B.3: Comparison of the generated text from the patched inferences using different large language models. The table displays model-wise outputs for an input claim.

| | | *"Robin was murdered by the Joker in a 1989 book."* | |
|---|---|---|---|
| | **Llama-2-13b-chat-hf** | **Meta-Llama-3-8B-Instruct** | **Falcon-7b-instruct** |
| $L_0$ | IsAuthor(Boek), NoBooksWritten(Boek) | | IsPerson(John Doe), IsPerson(Jane Doe), IsPerson(Bob Smith) |
| $L_1$ | *"Here's the updated list of facts:"* | *"I think there might be some confusion! "Robin" is a name, not a fact."* | IsPerson(Joker) ∧ ActorOf(Joker, Joaquin Phoenix) |
| $L_2$ | IsAuthor(J.K. Rowling), IsBook(Harry Potter), TitleOf(Harry Potter, Harry Potter and the Philosopher's Stone) | IsBook(Book) ∧ TitleOf(Book, 'The Book'), IsBook(Book) ∧ AuthorOf(Book, 'Author'), IsBook(Book) ∧ TitleOf(Book, 'The Book') ∧ AuthorOf(Book, 'Author') ∧ ¬IsBook(Book) ∧ ¬TitleOf(Book, 'The Book') ∧ ¬AuthorOf(Book, 'Author') | IsPerson(Joker) ∧ ActorOf(Joker, Joaquin Phoenix) |
| $L_3$ | IsBand(The Beatles), IsCharacter(Robin) | IsBook(King's novel, The Godfather), IsBook(King's novel, The Shining), IsBook(King's novel, Misery), IsBook(King's novel, Carrie), IsBook(King's novel, The Stand), IsBook(King's novel, It), IsBook(King's novel, The Dark Half) | IsPerson(Joker) ∧ ActorOf(Joker, Joaquin Phoenix) ∧ CountryOf(Joker, Gotham City) |
| $L_4$ | IsBatmanVillain(Joker), IsEnemyOf(Batman, Joker) | IsBook(Twain, The Adventures of Huckleberry Finn) ∧ IsCharacter(Huckleberry Finn, The Adventures of Huckleberry Finn) ∧ IsBook(Poe, The Raven) ∧ IsCharacter(Raven, The Raven) ∧ ¬IsBook(Poe, The Adventures of Huckleberry Finn) ∧ ¬IsCharacter(Huckleberry Finn, The Raven) | IsPerson(Jack the Ripper) ∧ IsPerson(Mary Ann Nichols) |
| $L_5$ | IsBatmanVillain(Joker), IsEnemyOf(Batman, Joker), IsComicBookCharacter(Joker) | IsBook(Joker) ∧ IsBookOf(Heath Ledger, Joker) ∧ IsBookOf(Joaquin Phoenix, Joker), IsBookOf(Batman, Joker) ∧ IsComicBook(Batman) ∧ IsComicBookOf(DC Comics, Batman) | |
| $L_6$ | IsCharacter(Joker), IsEnemyOf(Joker, Batman), IsComicBookSeries(Batman) | | IsPerson(Joker) ∧ IsPerson(Batman) ∧ IsPerson(Robin) |
| $L_7$ | IsCharacter(Joker), IsEnemyOf(Joker, Batman), IsComicBookSeries(Batman) | | IsPerson(Joker) ∧ IsPerson(Batman) ∧ IsPerson(Robin) |
| $L_8$ | IsCharacter(Joker), IsEnemyOf(Joker, Batman), IsComicBookSeries(Batman) | IsCharacter(Joker, Batman) ∧ IsVillain(Joker) ∧ HasKilled(Joker, Batman's Parents) ∧ IsNotTheSameAs(Joker, Batman) | IsPerson(Jack the Ripper) ∧ IsPerson(Mary Ann Nichols) |
| $L_9$ | IsFictionalCharacter(Joker), IsEnemyOf(Batman, Joker), HasAppearedIn(Joker, Batman: The Dark Knight) | IsCharacter(Joker, Batman's Villain), IsKiller(Joker, Batman's Victims), IsNotTheSamePerson(Joker, Batman's Killer, The Riddler) | |
| $L_{10}$ | IsCharacter(Joker), IsEnemyOf(Joker, Batman), HasSuperpower(Joker, Insanity) | IsCharacter(Joker, Batman) ∧ IsVillain(Joker) ∧ DidNotKill(Joker, Batman) ∧ DidNotKill(Joker, Robin) ∧ DidNotKill(Joker, Batgirl) ∧ DidNotKill(Joker, Commissioner Gordon) ∧ DidNotKill(Joker, Other Characters) | |
| $L_{11}$ | IsCharacter(Joker), IsEnemyOf(Joker, Batman), HasSuperpower(Joker, Insanity) | IsCharacter(Joker, Batman) ∧ IsVillain(Joker) ∧ DidNotKill(Batman, Joker) ∧ DidNotKill(Joker, Batman) | |
| $L_{12}$ | IsCharacter(Joker), IsEnemyOf(Joker, Batman), HasSuperpower(Joker, Insanity) | IsCharacter(Joker, Batman) ∧ IsKiller(Joker, Batman) ∧ IsComicBookCharacter(Joker), IsBook(The Killing Joke) ∧ IsComicBookStory(The Killing Joke, Alan Moore, Brian Bolland) ∧ IsMainCharacterOf(The Killing Joke, Batman) ∧ IsVillainOf(The Killing Joke, Joker) | ]} |
| $L_{13}$ | IsCharacter(Joker), IsEnemyOf(Joker, Batman), HasAppearedIn(Joker, Batman: The Dark Knight) | IsCharacter(Joker, Batman's Villain), IsComicBookCharacter(Joker), DidNotKill(Joker, Batman's Wife, Barbara Gordon), DidNotKill(Joker, Batman, Bruce Wayne) | IsPerson(Joker) ∧ IsPerson(Batman) ∧ IsPerson(Robin) |
| $L_{14}$ | IsCharacter(Joker), IsEnemyOf(Joker, Batman), HasAppearedIn(Joker, Batman: The Dark Knight) | IsCharacter(Joker, Batman) ∧ IsKiller(Joker) ∧ IsMurderer(Joker), IsBook(The Dark Knight) ∧ IsComicBook(The Dark Knight) ∧ IsNotBasedOnBook(The Dark Knight, The Killing Joke) | IsPerson(Joker) ∧ IsPerson(Batman) ∧ IsPerson(Robin) |
| $L_{15}$ | IsFictionalCharacter(Robin), IsSidekickOf(Batman, Robin), IsSuperhero(Robin) | IsCharacter(Joker, Batman) ∧ IsVillain(Joker) ∧ IsKiller(Joker) ∧ IsDead(Joker) ∧ IsNotReal(Joker) ∧ IsFictional(Joker) ∧ IsNotBasedOnTrueStory(Joker) | IsPerson(Joker) ∧ IsPerson(Batman) ∧ IsPerson(Robin) |
| $L_{16}$ | IsFictionalCharacter(Joker), IsEnemyOf(Batman, Joker), HasAppearedIn(Batman, The Dark Knight), HasAppearedIn(Batman, The Dark Knight Rises) | IsCharacter(Batman) ∧ IsVillain(The Joker) ∧ IsVictimOf(Batman, The Joker), IsBook(The Killing Joke) ∧ IsStorylineOf(The Killing Joke, The Joker's Assault on Barbara Gordon), IsNotRealEvent(The Killing Joke) ∧ IsFictionalStoryline(The Killing Joke) | IsPerson(Joker) ∧ IsPerson(Batman) ∧ IsPerson(Robin) |
| $L_{17}$ | IsCharacter(Joker), IsEnemyOf(Joker, Batman), HasSuperpower(Joker, Insanity) | IsCharacter(Batman) ∧ IsVillain(Joker) ∧ IsVictim(Joker, Batman), IsComicBookStory(Batman) ∧ IsPlotElement(Murder) ∧ IsPlotElement(Robbery) ∧ IsPlotElement(Heist) | IsPerson(Joker) ∧ IsPerson(Batman) ∧ IsPerson(Robin) |
| $L_{18}$ | IsCharacter(Joker), IsVictimOf(Joker, Batman), IsComicBookSeries(Batman), IsComicBookCharacter(Joker, Batman) | | IsPerson(Rafael Nadal) ∧ TennisPlayer(Rafael Nadal) ∧ CountryOf(Rafael Nadal, Spain) |

| | Llama-2-13b-chat-hf | Meta-Llama-3-8B-Instruct | Falcon-7b-instruct |
|---|---|---|---|
| $L_{19}$ | IsComicBookCharacter(Joker), IsEnemyOf(Batman, Joker), IsSupervillain(Joker) | IsCharacter(Batman) ∧ IsVillain(Joker) ∧ IsVictim(Joker, Batman) ∧ IsLocation(Gotham City) ∧ IsComicBookStory(Robin, Batman, Joker, Gotham City) | IsPerson(Rafael Nadal) ∧ TennisPlayer(Rafael Nadal) ∧ CountryOf(Rafael Nadal, Spain) |
| $L_{20}$ | IsComicBookCharacter(Robin), IsSidekickOf(Batman, Robin), DoesNotExist(Robin, The Beatles) | IsCharacter(Batman) ∧ IsVillain(Joker) ∧ IsComicBook(Robin) ∧ IsMovie(Batman) ∧ IsBook(Batman) | IsPerson(Rafael Nadal) ∧ TennisPlayer(Rafael Nadal) ∧ CountryOf(Rafael Nadal, Spain) |
| $L_{21}$ | IsComicBookCharacter(Joker), IsEnemyOf(Batman, Joker), IsSupervillain(Joker) | IsCharacter(Batman) ∧ IsVillain(Joker) ∧ IsComicBook(Robin) ∧ IsLocation(GothamCity) ∧ IsEvent(BatmanKilledByJoker) | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{22}$ | IsComicBookCharacter(Joker), IsEnemyOf(Batman), IsFrom(GothamCity) | IsCharacter(Joker) ∧ IsComicBookCharacter(Joker), IsVillain(Joker) ∧ IsMurderer(Joker) ∧ IsCriminal(Joker), IsComicBookSeries(Batman) ∧ IsComicBookSeries(Joker), IsComicBookStory(Batman: The Killing Joke) ∧ IsComicBookStory(Joker) | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{23}$ | IsComicBookCharacter(Joker), IsEnemyOf(Batman, Joker), IsFrom(Joker, GothamCity) | *"I think you meant to ask a question!"* | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{24}$ | *This statement is true.* | *"I assume you're referring to the fact that the Joker, a character from the Batman comics, is known for his iconic phrase "Why so serious?" which is often used to describe his chaotic and unpredictable nature"* | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{25}$ | *"This statement is true."* | *"I think you meant to ask another question!"* | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{26}$ | *"This statement is true."* | *"I think you meant to ask another question!"* | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{27}$ | *"This statement is true."* | *"I assume you're referring to the fact that I previously verified that the statement "The Beatles were a rock band from England" is true"* | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{28}$ | *"This statement is true."* | *"It seems like you didn't provide a claim for me to fact-check!"* | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{29}$ | *"This statement is true."* | *It seems like you didn't provide a claim for me to fact-check!"* | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{30}$ | *"This statement is true."* | *"It seems like you didn't provide a claim for me to fact-check!"* | IsCity(New York) ∧ CountryOf(New York, USA) ∧ CapitalOf(New York, USA) |
| $L_{31}$ | *"This statement is true."* | *"It seems like you didn't provide a claim for me to fact-check!"* | IsCity(Paris) ∧ CityIn(Paris), IsCity(Paris) ∧ CityIn(Paris) |
| $L_{32}$ | *"This statement is true."* | *"I apologize for the mistake! Thank you for the opportunity to correct it."* | *"</|</|<new Yorker«/|<new Yorker« new Yorker< new Yorker< new Yorker< new Yorker<;"> New Yorker<;"><;">"* |
| $L_{33}$ | *"This statement is true."* | | |
| $L_{34}$ | *"This statement is true."* | | |
| $L_{35}$ | *"This statement is true."* | | |
| $L_{36}$ | *I'm ready to assist you with more questions! Go ahead and ask your next one."* | | |
| $L_{37}$ | *"I'm ready to assist you with more questions! Go ahead and ask your next one."* | | |
| $L_{38}$ | *"I'm ready to assist you with more claims! Please go ahead and present the next one."* | | |
| $L_{39}$ | *"This statement is true."* | | |
| $L_{40}$ | *"This statement is true."* | | |

## B.8 Attentions on Prompt Tokens across Layers

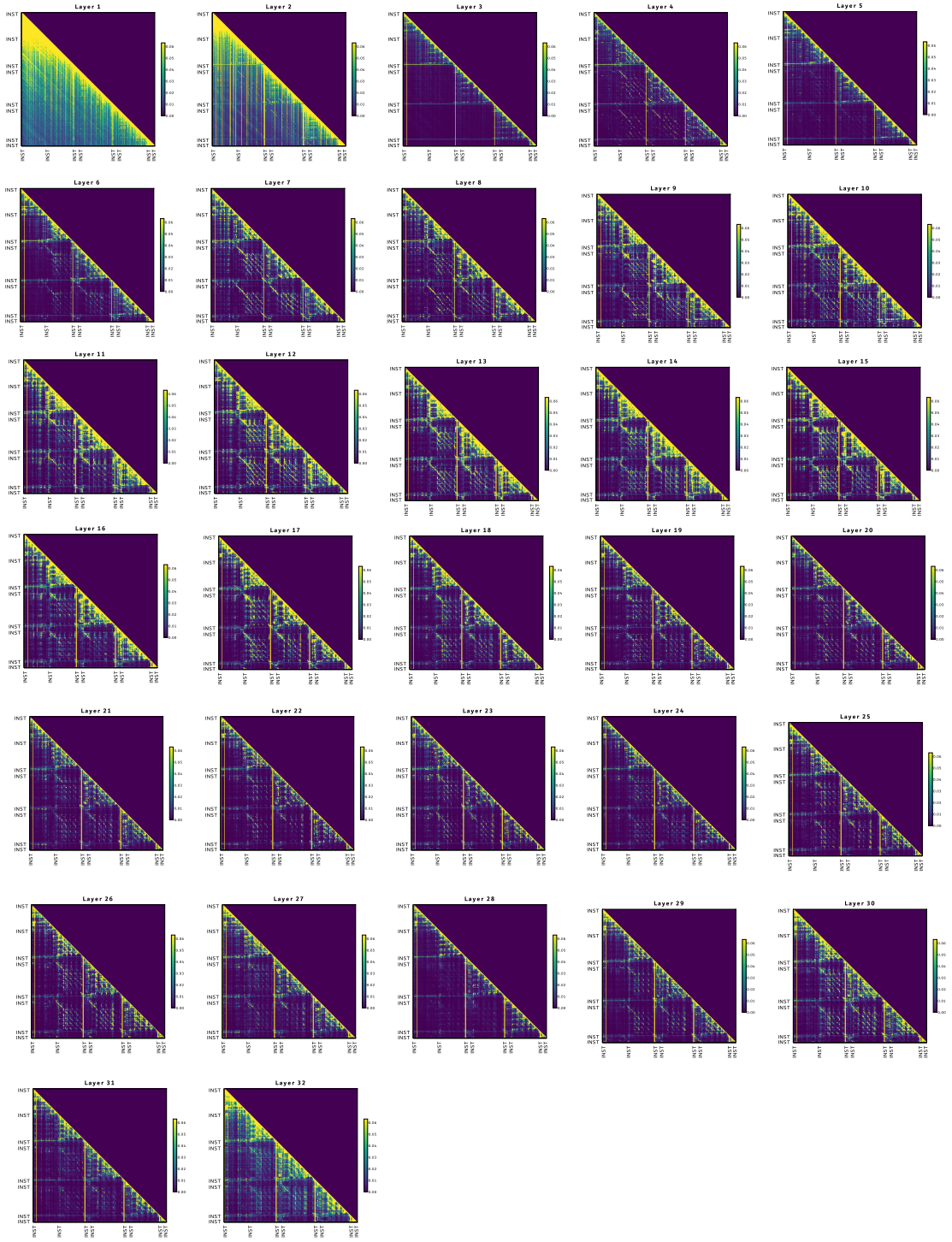



Figure B.5: Attention matrices across the hidden layers of the computation of the model $\mathcal{M}$ on the input sequence token $\mathcal{S}$ with given input claim.

# Appendix C

# Extracting Semantic Concepts with Vector Symbolic Architectures

## C.1 Synthetic corpus

Table C.1: All domains, and their corresponding cardinality after data augmentation for training.

| Domain | Examples | Domain | Examples | Domain | Examples |
|---|---|---|---|---|---|
| country_capital | 21801 | capital_world | 18561 | country_language | 12299 |
| antonyms_gradable | 11268 | adj_superlative | 10942 | un+adj_reg | 10614 |
| adj+ly_reg | 10576 | adj_comparative | 10519 | male_female | 10236 |
| noun_plural_reg | 10216 | noun_plural_irreg | 10206 | verb_Ving_Ved | 10164 |
| verb_inf_3pSg | 10112 | animal_sound | 10083 | verb_inf_Ving | 10008 |
| name_nationality | 9998 | verb+er_irreg | 9865 | verb_Ving_3pSg | 9861 |
| verb+able_reg | 9849 | adj+ness_reg | 9849 | animal_shelter | 9833 |
| hypernyms_animals | 9831 | over+adj_reg | 9828 | re+verb_reg | 9821 |
| verb+ment_irreg | 9807 | verb_inf_Ved | 9805 | UK_city_county | 9805 |
| name_occupation | 9801 | noun+less_reg | 9801 | verb_3pSg_Ved | 9801 |
| verb+tion_irreg | 9801 | hypernyms_misc | 9719 | antonyms_binary | 9603 |
| past_tense | 6313 | plural | 4129 | comparative | 3765 |
| present_participle | 3401 | plural_verbs | 3055 | currency | 2983 |
| adjective_to_adverb | 2977 | math_double | 2918 | nationality_adjective | 2818 |
| superlative | 2545 | math_division2 | 2498 | opposite | 2221 |
| math_division5 | 641 | family | 529 | math_squares | 402 |
| math_division10 | 258 | hyponyms_misc | 102 | math_root | 77 |
| math_cubes | 29 | | | | |
| DOMAINS: 52 | | | | TEXTUAL EXAMPLES: 395,944 | |

## C.2 Architecture of *hyperdimensional probe*

Table C.2: Configuration of the neural VSA encoder $\mathcal{M}$ for an input embedding dimension equal to $d$.

| **Component** | **Input Dim** | **Output Dim** | **Note** |
|---|---|---|---|
| **Input Layer** | | | |
| Linear Layer | $d$ | 4096 | - |
| Normalization | - | - | LayerNorm (4096) |
| Activation | - | - | GELU |
| **Residual Block 1** | | | |
| Linear Layer | 4096 | 4096 | GELU activation |
| Normalization | - | - | LayerNorm (4096) |
| Dropout | - | - | $p = 0.5$ |
| Residual Connection | - | - | Identity |
| **Residual Block 2** | | | |
| Linear Layer | 4096 | 4096 | GELU activation |
| Normalization | - | - | LayerNorm (4096) |
| Dropout | - | - | $p = 0.5$ |
| Residual Connection | - | - | Identity |
| **Output Layer** | | | |
| Normalization | - | - | LayerNorm (4096) |
| Linear Layer | 4096 | 4096 | - |
| Activation | - | - | Tanh |
| Trainable parameters with: | $d = 1024, 55M$ | | |
| | $d = 2048, 59M$ | | |
| | $d = 4096, 67M$ | | |
| | $d = 5120, 71M$ | | |

## C.3 Unbinding stage from Section 3.3.6

Table C.3: **Unbinding stage**: Proportions of the best unbinding concepts used for extracting concepts from VSA encodings across different models, with overall mean and standard deviation. `Key` refers to cases where the candidate concept corresponds to the key of the target pair ($b_1$), while `NONE` indicates that no unbinding operations were applied to the probed VSA encoding. `Example` denotes a concept where the key ($a_1$) and value ($a_2$) from the in-context example were pre-bound. Lastly, `Context` represents a scenario where the in-context example ($a_1, a_2$) was pre-bound together with the key of the target pair ($b_1$). On the other hand, `Greedy` means using a concept candidate from the vocabulary, rather than picking it among those of the input. The table has been trimmed to highlight the relevant and common items across the models. We consider the first four strategies to be the most relevant, as they account for 97% of all unbinding operations across models.

| **Concept for unbinding** (%) | **GPT-2** | **Pythia** | **Llama 4, Scout** | **OLMo-2** | **Phi-4** | **Llama 3.1** | **AVERAGE** |
|---|---|---|---|---|---|---|---|
| Key | 65.9 | 74.4 | 83.2 | 83.2 | 87.4 | **87.9** | 80.3 ± 7.8 |
| NONE | **22.0** | 16.9 | 11.6 | 8.6 | 7.7 | 7.0 | 12.3 ± 5.4 |
| Example Key | **6.0** | 2.6 | 1.0 | 2.1 | 1.5 | 1.7 | 2.5 ± 1.7 |
| Context | 1.2 | 2.0 | 2.6 | **4.5** | 1.3 | 1.5 | 2.2 ± 1.2 |
| Greedy | 2.1 | 1.9 | 1.3 | 0.9 | 1.2 | 0.9 | 1.4 ± 0.5 |
| Example Value | 1.6 | 1.5 | 1.0 | 0.4 | 0.7 | 0.5 | 0.9 ± 0.5 |
| Cleaned Example Key | 0.2 | 0.5 | 0.0 | 0.1 | 0.2 | 0.1 | 0.2 ± 0.2 |
| Cleaned Example Value | 0.9 | 0.1 | 0.0 | 0.1 | 0.1 | 0.1 | 0.2 ± 0.3 |
| Cleaned Key | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.0 ± 0.0 |
| Cleaned Original | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 ± 0.0 |
| Example | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 ± 0.0 |
| Example Value & Key | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 ± 0.0 |
| Example Key & Key | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 ± 0.0 |

## C.4 Experimental comparison

**Logit Attribution.** Initially, Section 4.3.3 compares our VSA-based results to those generated by the technique of Direct Logit Attribution (DLA; Section 5.3.1); as it requires no extra steps such as feature-naming. This makes DLA the most direct and unambiguous comparison for our experimental setting. It outputs a single, unique and unambiguous feature (token) constrained by the model's output vocabulary, enabling a direct comparison through a fuzzy token-to-concept matching with our concept set.

**Sparse Autoencoders.** Conversely, while our controlled vector space, defined by VSA encodings, parallels the SAE proxy layer, our methodology adopts a top-down strategy by querying this proxy space with predefined concepts (Equation 3.12), whereas SAEs follow a bottom-up process that assigns labels to all activated features post hoc. This bottom-up paradigm uncovers an unbounded set of latent features without inherent

relevance filtering, necessitating exhaustive feature naming and additional selection to isolate those aligned with our bounded input–output concept framework. The manual intervention required in SAE-based methods, from feature naming to filtering, prevents them from being directly and unambiguously comparable to our supervised approach.

Section 4.3.4 presents an analysis based on Sparse Autoencoders (SAEs; Section 5.3.3), detailing the experimental choices undertaken to enable a comparison with our VSA-based findings.

**Supervised probes.** Our VSA-based encoder does qualify as a supervised probe, as it is trained to map LLM internal representations (i.e., residual stream) into interpretable, human-understandable features (i.e., VSA encodings). Supervised probes are typically designed for specific experimental goals or target features, ranging from syntactic structure, as in *"A Polar Coordinate System Represents Syntax in Large Language Models"* [40]; to real-world knowledge, as in *"Language Models Represent Space and Time"* [70]; and to abstract semantics, as in *"The Geometry of Truth"* [105].

However, our probe is specifically designed around VSAs principles, so direct comparisons with non-VSA probes would require fundamentally different approaches, not grounded in VSAs.

### C.4.1 DLA-based experimental results

Table C.4: Concepts extracted using the DLA probing technique on the corpus $\bar{\mathcal{S}}$ with all LLMs. Likewise in VSA-based probing, we focus on the same middle-to-bottom range of model's hidden layers of the last token. The table highlights the key common items across models, with the first six cases covering over 95% of all extracted concepts.

| **Extracted Concepts** (docs, %) | **GPT-2** | **Pythia** | **Llama4, Scout** | **OLMo-2** | **Phi-4** | **Llama 3.1** | **AVERAGE** | **Δ VSA** |
|---|---|---|---|---|---|---|---|---|
| NONE | 33.9 | 32.8 | 15.4 | 14.6 | 33.1 | **47.4** | 29.5 ± 11.4 | +17.3 |
| **Target** | 15.0 | 18.0 | **36.7** | 29.0 | 34.4 | 22.5 | 25.9 ± 8.1 | +25.8 |
| Key \| **Target** | 12.6 | 19.3 | 38.4 | **38.5** | 22.7 | 22.1 | 25.6 ± 9.7 | - 50.4 |
| Key | 9.7 | **10.4** | 6.3 | **10.4** | 6.0 | 4.5 | 7.9 ± 2.4 | +5.0 |
| Example Value | **12.8** | 5.7 | 0.3 | 0.7 | 0.9 | 0.5 | 3.5 ± 4.6 | +3.5 |
| Example | **9.0** | 5.3 | 0.3 | 1.7 | 1.0 | 0.5 | 3.0 ± 3.2 | +0.7 |
| Example Value \| **Target** | 1.0 | **2.1** | 0.7 | 0.6 | 0.3 | 0.9 | 0.9 ± 0.6 | +0.9 |
| Example Key | **3.0** | 1.5 | 0.1 | 0.2 | 0.1 | 0.1 | 0.8 ± 1.1 | +0.7 |
| Context \| **Target** | 0.6 | 0.3 | 0.5 | **1.5** | 0.3 | 0.4 | 0.6 ± 0.4 | -1.5 |

Table C.5: Percentages of extracted factors by analogy category considering the subset of instances when the DLA yields no concept for OLMo-2.

| **Extracted concepts** (docs, %) | **Morphological Modifiers** | **Verbal & Grammatical Forms** | **Factual Knowledge** | **Semantic Relations** | **Mathematics** | **Semantic Hierarchies** | **AVERAGE** |
|---|---|---|---|---|---|---|---|
| Key \| Target | **90.3** | 83.4 | 70.1 | 79.4 | 0.0 | 41.5 | 60.8 ± 30.3 |
| `NONE` | 1.6 | 2.7 | 14.3 | 1.7 | **91.1** | 21.1 | 22.1 ± 31.1 |
| Example | 0.7 | 0.7 | 4.7 | 8.5 | 0.0 | **15.0** | 4.9 ± 5.1 |
| Key | 1.6 | 2.7 | 3.8 | 1.4 | 0.0 | **5.1** | 2.4 ± 1.7 |
| Key \| Pair Values | 1.3 | 0.9 | 0.0 | 5.1 | 0.0 | **11.6** | 3.2 ± 4.2 |
| Context \| Target | 4.4 | **7.4** | 0.8 | 1.6 | 0.0 | 0.6 | 2.5 ± 2.6 |
| Out-of-Context | 0.2 | 0.6 | 1.3 | 0.0 | **8.8** | 0.9 | 1.9 ± 2.9 |
| Context | 0.0 | **0.3** | 0.1 | 0.0 | 0.0 | **0.3** | 0.1 ± 0.1 |

Table C.6: Concepts extracted though DLA-based probing when VSA yields no concepts. DLA extract no concepts in the majority of the instances (59 ± 15 %), highlighting high variability among models.

| **Extracted concepts** (docs,%) | **GPT-2** | **Pythia** | **Llama3** | **Phi-4** | **OLMo-2** | **Llama4** | **AVERAGE** |
|---|---|---|---|---|---|---|---|
| `None` | 40.9 | 46.5 | 83.1 | 72.7 | 55.9 | 53.9 | 58.8 ± 14.8 |
| **Target** | 8.1 | 9.0 | 5.3 | 15.7 | 14.3 | 23.3 | 12.6 ± 6.4 |
| Key | 7.5 | 11.7 | 3.4 | 4.3 | 13.2 | 9.1 | 8.2 ± 3.7 |
| Key \| **Target** | 6.9 | 8.7 | 5.7 | 3.6 | 11.8 | 11.9 | 8.1 ± 3.3 |
| Example Value | 17.9 | 11.1 | 1.3 | 1.0 | 1.3 | 0.6 | 5.5 ± 6.7 |
| Example | 12.1 | 6.6 | 0.4 | 2.2 | 1.7 | 0.1 | 3.9 ± 4.3 |
| Example Key | 3.3 | 2.7 | 0.1 | 0.1 | 0.2 | 0.2 | 1.1 ± 1.2 |
| Example Value \| **Targe**t | 1.0 | 1.7 | 0.1 | 0.0 | 0.2 | 0.2 | 0.5 ± 0.7 |
| Example Value \| Key | 1.0 | 0.6 | 0.1 | 0.0 | 0.4 | 0.3 | 0.4 ± 0.4 |
| Example Key \| Key | 0.3 | 0.4 | 0.1 | 0.0 | 0.1 | 0.1 | 0.2 ± 0.2 |
| Example Value \| Key \| **Target** | 0.3 | 0.3 | 0.0 | 0.0 | 0.3 | 0.2 | 0.2 ± 0.1 |
| Context \| Target | 0.3 | 0.2 | 0.3 | 0.2 | 0.4 | 0.1 | 0.3 ± 0.1 |
| Example Key \| Target | 0.2 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 ± 0.1 |
| Target \| Example | 0.1 | 0.1 | 0.0 | 0.0 | 0.1 | 0.0 | 0.1 ± 0.1 |

### C.4.2 Raw results obtained though the DLA probing technique

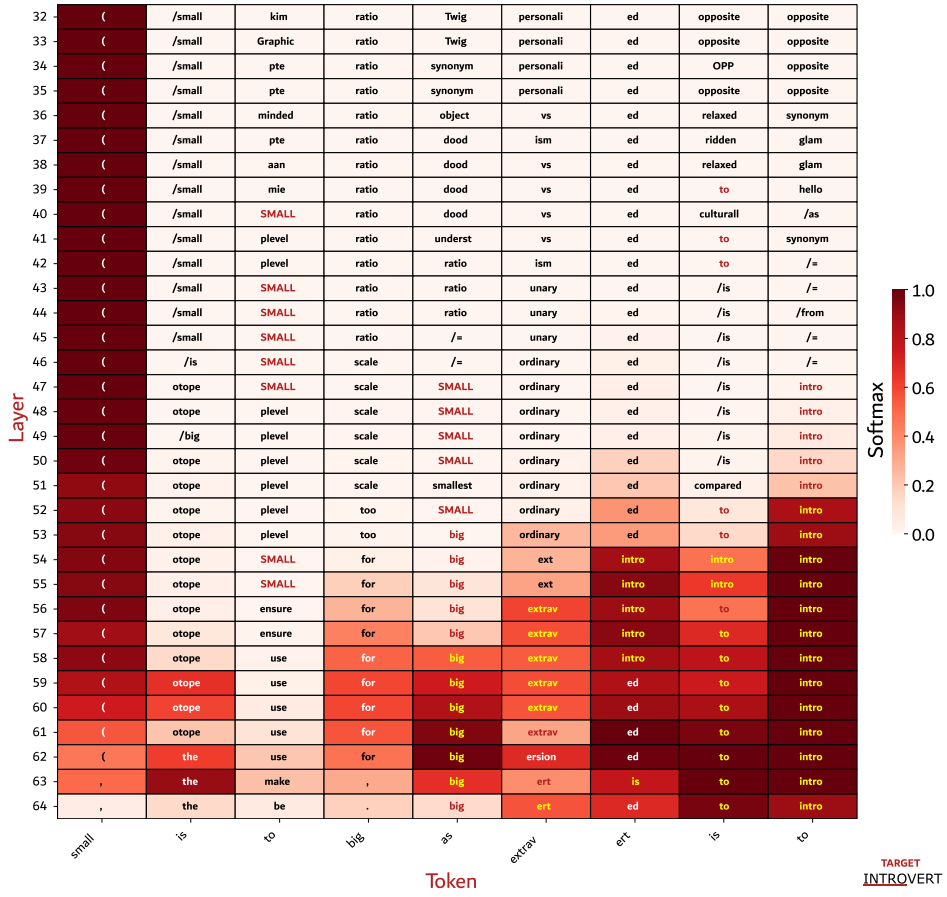



Figure C.1: Raw outputs obtained though DLA on OLMo-2 for a sampled analogy.

## C.5 Applicability to other domains

### C.5.1 Generalization of input representation

VSA representations are automatically generated from input features, with their construction guided by the probing objective and the target latent features. While our work focuses on textual inputs with well-defined semantics, allowing straightforward extraction of input features (i.e., words), the underlying principle is flexible and generalizable. Equation 3.10 illustrates the creation of input representations via binding and bundling operations for our specific input template and downstream task. The hyperdimensional algebra underlying VSA allows this approach to generalize to other textual formats, NLP tasks, and even multi-modal data.

Scalability challenges depend largely on the nature of the input features. For tasks such as toxicity detection, expert-labeled data or specialized feature extraction pipelines may be required. For example, mapping the phrase *"You are a pathetic excuse for a human just like the rest of your kind"* to a conceptual form such as $(\phi_{\text{attack}} \odot \phi_{\text{insult}}) + (\phi_{\text{attack}} \odot \phi_{\text{identity}})$ requires human expertise. Once features are extracted, however, constructing VSA encodings is automatic, efficient, and scalable. VSA probing can then uncover encoded concepts in the LLM vector space, for instance:

$$y_s \oslash \phi_{\text{attack}} = \phi_{\text{identity}} + \text{noise}$$

In contrast, tasks based on syntactic structures offer more scalable input extraction. For example, the sentence *"The city of Turin is in Italy"* can be processed with conventional techniques such as POS tagging and Semantic Role Labeling (SRL). A VSA encoding can then be automatically created:

$$(\phi_{\text{NOUN}} \odot \phi_{\text{city}}) + (\phi_{\text{PROPN}} \odot \phi_{\text{Turin}}) + (\phi_{\text{VERB}} \odot \phi_{\text{be}}) + (\phi_{\text{PROPN}} \odot \phi_{\text{Italy}})$$

### C.5.2 Applicability to other downstream tasks

The proposed methodology is generalizable to other experimental settings. The analogy-based dataset was chosen to:

- provide a simple, controlled, and interpretable evaluation environment;
- elicit LLMs to focus on concepts and their inherent relationships;
- probe the LLM vector space with inputs spanning a spectrum of reasoning tasks.

Thanks to the flexibility of VSAs and hypervector algebra, VSA-based probing can be applied to a wide variety of experimental settings with different:

1. **Downstream tasks**. Our decoding paradigm can be used for linguistic feature extraction, toxicity detection, or bias classification;

2. **Textual templates**. For example, in question-answering setting, an input text in such as *"Who wrote the play Romeo and Juliet?"* can be encoded as

$$(\phi_{task} \odot \phi_{question}) + (\phi_{relation} \odot \phi_{writtenBy}) + (\phi_{play} \odot \phi_{Romeo\&Juliet})$$

   allowing the VSA to query LLM representations and reveal which concepts are strongly represented or linked to the predicted answer;

3. **Modalities**. Inputs combining text with other modalities could also be encoded and probed via VSAs.

VSA-based probing thus provides a unified, flexible framework for examining how LLMs encode and relate abstract input features, from syntactic structures to high-level concepts such as gender bias or toxic language.

### C.5.3 Handling of Out-of-Domain Data

The neural VSA encoder, a central component of the hyperdimensional probe (Section 3.3.5), is designed to map concepts with their corresponding neural representations, specifically, to map VSA encodings onto the appropriate directions or subspaces within LLM embeddings. To achieve this, the encoder must be exposed to a few examples of the latent features being targeted, whether drawn from in-domain or out-of-domain distributions. In other words, the hyperdimensional probe cannot extract latent features (i.e., concepts) that it has never encountered.

For instance, one might focus on toxicity by training on toxic content from social platforms, thereby mapping textual inputs to toxicity-related concepts (see Section C.5.1). An example could be the phrase "You are a pathetic excuse for a human just like the rest of your kind", which would be associated with the concept. Once trained, the encoder can then be applied to out-of-domain text to identify potential toxicity-related concepts.

Moreover, we evaluate our probe on textual inputs that differ from those used during training (Figure 4.8). This allows us both to control for confounding effects and to

validate the probe's out-of-domain performance. While and share the same underlying semantic concepts, they consist of distinct texts with a different syntactic structure. Consequently, they can be seen as out-of-domain inputs with respect to the LLM embeddings, the inputs processed by our trained encoder.

Importantly, the probe's performance is not affected by the size or construction strategy of the codebook. The codebook, essentially the list of all features, functions such as a vocabulary, unconstrained in terms of cardinality, type, or source of its symbols, thanks to the properties of VSA. However, as with all interpretability probes, its effectiveness ultimately depends on the type of feature being targeted. Consequently, the strategy used to construct input representations (i.e., VSA encodings) directly impacts probe performance. For example, if the probe attempts to map or extract latent features that are unlikely to be encoded in the LLM's vector space, the resulting outputs may be unreliable or meaningless (see the "unrelated baseline" in Table 3.11).

## C.6 Cosine similarities among the VSA codebook items

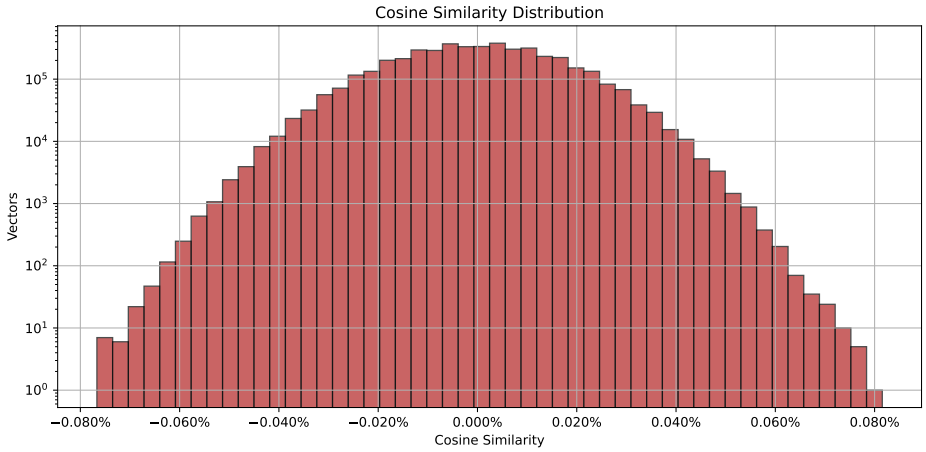



Figure C.2: Distribution of pair-wise cosine similarities among the codebook items.

## C.7 Question-answering setting

We generate training examples $\mathcal{Q}$ from the SQuAD dataset using an augmenting strategy by incrementally considering textual questions with their corresponding features:

$(A_1)$ "What was the `name`" $\mapsto \phi_{\text{name}}$

$(A_2)$ "What was the `name` of the `ship`" $\mapsto \phi_{\text{name}} + \phi_{\text{ship}}$

$(A_3)$ "..." $\mapsto \ldots$

$(A_{n-1})$ "What was the `name` of the `ship` that `Napoleon` sent to `the Black Sea`?"

$$\mapsto \phi_{\text{name}} + \phi_{\text{ship}} + \phi_{\text{napoleon}} + \phi_{\text{send}} + \phi_{\text{theBlackSea}}$$

$(A_n)$ "What was the `name` of the `ship` that `Napoleon` sent to `the Black Sea`?"

`Charlemagne`" $\mapsto (\phi_{\text{name}} + \phi_{\text{ship}} + \phi_{\text{napoleon}} + \phi_{\text{send}} + \phi_{\text{theBlackSea}}) + \phi_{\text{charlemagne}}$

For our experiments, we generate another corpus $\bar{\mathcal{Q}}$ including also the contextual text (Wikipedia article) provided for each SQuAD's items:

"Napoleon III responded with a show of force ... by the Greek Orthodox Church.

Q: What was the name of the ship that Napoleon sent to the Black Sea?

A ($\leq$ 3 words):"

Lastly, we apply our entire pipeline by probing the final state of a language model at the last token (colon) and extracting concepts through comparison with the codebook $\Phi$. We analyze the model's internal state across the text generation process, considering the residual stream at initialization ($\mathbf{H}[\text{seq}_0]$) and after the autoregressive generation of $t$ tokens ($\mathbf{H}[\text{seq}_t]$).

# C.8 Dimensionality reduction

## C.8.1 Average correlations among model's hidden layer

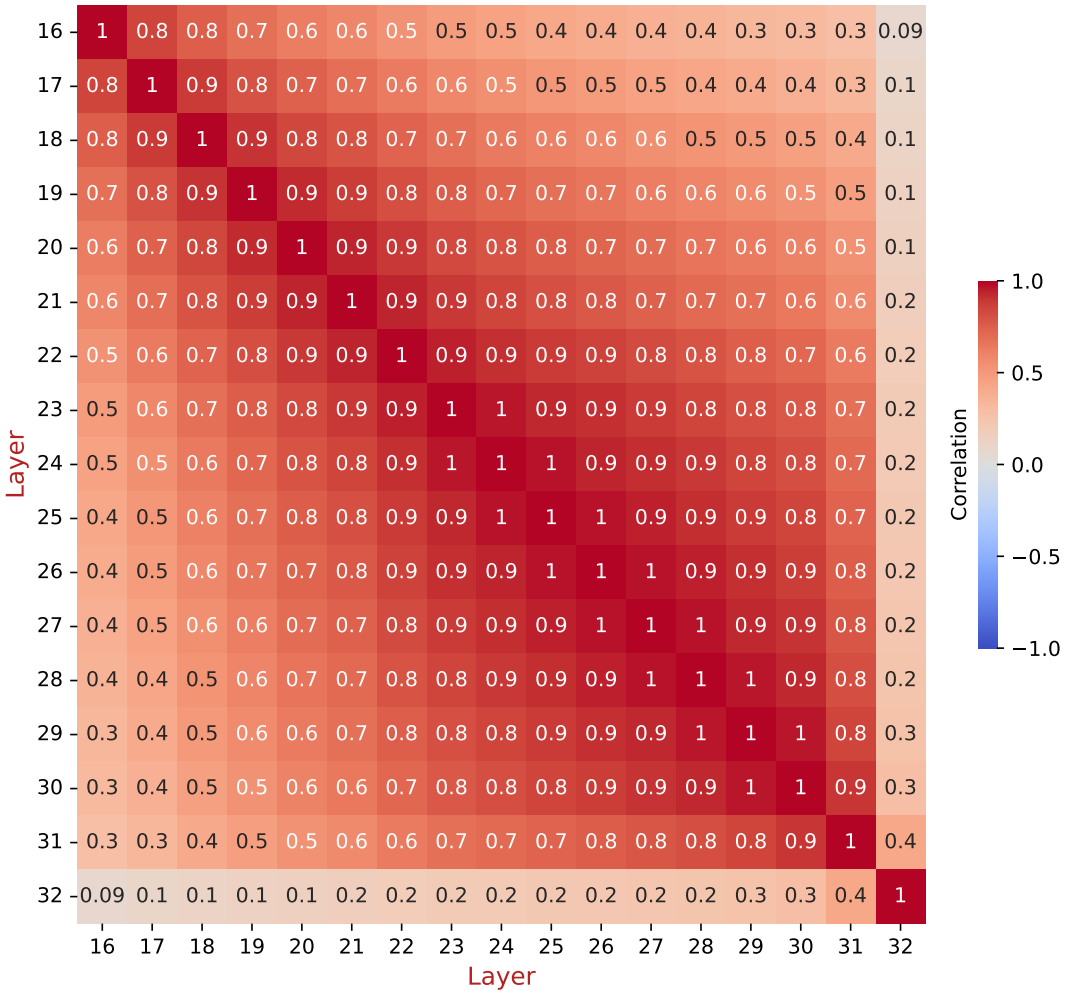



Figure C.3: Average person correlations in the latter half of Llama3.1's hidden layers.

## C.8.2 Analysis of representation redundancy

In Section 3.3.4, we hypothesize that highly correlated rows (model's adjacent layers) could cause redundant representations, since they likely encode similar numerical patterns, and thus information.

Here, we present an analysis of representation redundancy, defined as approximate linear dependence among LLM hidden layer embeddings. We computed the Gram matrix $G = HH^T$, where $H$ is the model's residual stream, and analyzed its eigenvalues.

Table C.7 shows results for the OLMo-2 model (considering the 32nd-to-64th range of hidden layers; Table 3.11), averaged on a 100K training input sample. The spectrum reveals a few dominant eigenvalues (around 3-4 modes) followed by many smaller ones, indicating that the embedding space is approximately low-rank. This suggests that, when considering the full matrix ($\mathbb{R}^{33\times5120}$ for OLMo-2), most hidden layer representations (rows) are redundant, since only a few rows (or their combinations) contribute meaningful structure. The first mode is by far the most dominant, with a normalized eigenvalue of 0.65, compared to 0.17 for the second. We hypothesize that this leading component might correspond to next-token prediction representations, while the remaining modes capture secondary structures or auxiliary information. Our hyperdimensional probe aims to capture also these auxiliary latent structures, rather than limiting solely on the single predominant component.

Table C.7: Eigenvalues (EV) of the Gram matrix from OLMo-2's residual stream.

| Comp. | EV (mean ± std) | Norm. EV |
|---|---|---|
| 0 | 58084 ± 5293 | 0.650 |
| 1 | 15450 ± 2056 | 0.170 |
| 2 | 5972 ± 608 | 0.070 |
| 3 | 2539 ± 330 | 0.030 |
| 4 | 2057 ± 220 | 0.020 |
| 5 | 1187 ± 166 | 0.010 |
| 6 | 727 ± 119 | 0.010 |
| 7 | 505 ± 83 | 0.010 |
| 8 | 363 ± 59 | 0.000 |
| 9 | 282 ± 48 | 0.000 |
| 10 | 230 ± 37 | 0.000 |
| ... | ... | ... |
| 30 | 30 ± 6 | 0.000 |
| 31 | 27 ± 6 | 0.000 |
| 32 | 22 ± 6 | 0.000 |

### C.8.3 Ablation study on the dimensionality-reduction steps

This section presents an analysis of skipping the dimensionality reduction steps introduced in Section 3.3.4. While our VSA-based methodology would work without these compression steps, the overall computational cost of probing would dramatically increase. For example, our ingestion procedure (Algorithm 1) reduces the probed OLMo-2's embeddings from $\mathbb{R}^{33\times5120}$ to $\mathbb{R}^{5120}$. This allows our neural VSA encoder to have an input dimension $d = 5012$ with only 71M trainable parameters (see Section C.2).

If the two steps are eliminated, and thus the entire residual stream of the model $\mathbb{R}^{33\times5120}$ is considered, the encoder receives a flat input vector, creating an input dimension $d = 168960 \in \mathbb{R}^{168960}$. Although the encoder would internally handle feature extraction, since the flattened input holds all the information encoded in the LLM embeddings, this approach would increase the number of trainable parameters to 742 million, representing a tenfold increase. Additionally, adopting a lazy feature extraction stage in an input vector space of size $\approx 10^5$, which is approximately low-rank (see Section C.8.3), would result in a computationally inefficient approach.

Removing one of the two steps, such as sum pooling, should lead to just an increase of the overall computational cost for the encoder ($\mathbb{R}^{5\times5120} \mapsto \mathbb{R}^{25600}$; $d = 25600$; 155M trainable parameters; x2), rather than affecting probe's outputs. Further, since our neural VSA encoder is found effective to extract latent features even from our heavily-compressed input representation (Section 4.3), other dimensionality reduction approaches could also be as effective as ours (Algorithm 1).

In summary, skipping the compressing steps is possible and the only drawbacks should be the increase of footprint of both the training and inference stages of the VSA-based probing (see also Section C.9).

## C.9 Computational workload

The computational workload of this work is split into two parts: LLM inference (exogenous, Section 3.3.4) and the training and probing stages of our method (endogenous, Section 3.3.5 and 3.3.6).

The exogenous factor, running the Large Language Models, was the most computationally demanding task. For our experiments, we tested six different Large Language Models in inference mode, caching their embeddings for our training phase and probing them dynamically during the inference phase of our work (Figure 3.8). We worked with LLMs ranging from 355M parameters (GPT-2) to 109B parameters (Llama 4, Scout), using between one and three NVIDIA® A100-80GB GPUs, depending on the model size. Quantization is not employed.

In contrast, the computational demands of our VSA-based methodology is relatively low. The most resource-intensive stage was training our neural VSA encoder, but due to its modest size (ranging from $55M$ to $71M$ parameters, see Section C.2), this process remained lightweight. We performed this training on a single GPU, though it could easily be handled with much less powerful and lower-memory GPUs. Regarding GPU usage, we trained our encoder for approximately 8 hours on each LLM's embeddings,

though the process could have been shortened with a less conservative early stopping criterion or by reducing the amount of training data.

The probing stage is then composed of simple vector multiplications (unbinding, Section 2.2.2), after loading the heavy LLM and our lightweight trained neural VSA encoder into memory (from 800 MB of the 55M version to 1 GB of the biggest one). Future research could explore even further reducing the latent dimension of our neural VSA encoder (Section C.2) or adopt VSA encodings with lower dimensionality (e.g. $D = 512$, leading to a more lightweight encoder. The time cost of probing is thus comparable to simple LLM inference with a slight increase due to feedforward our lightweight trained model, with the time demand for vector multiplications being negligible. Accordingly, the GPU hours for probing depends mainly on the amount of test data and model size; for example, we used around 95 hours of GPU computation for probing embeddings of Llama 3.1-8B on $\bar{\mathcal{S}} \approx 10^5$, processing each instance in around 3 seconds.

La borsa di dottorato è stata cofinanziata con risorse dell'Unione europea.
NextGeneration EU - Piano Nazionale di Ripresa e Resilienza (PNRR)