Evaluation and Integration of Structural Knowledge for Pre-trained Language Models

Jesus Enrique Lovon Melgarejo

RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à l'Université Toulouse III - Paul Sabatier

Évaluation et intégration des connaissances structurelles pour modèles de langue pré-entraînés

---

Thèse présentée et soutenue, le 28 mars 2024 par

Jesus Enrique LOVON MELGAREJO

École doctorale
EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité
Informatique et Télécommunications

Unité de recherche
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Lynda TAMINE EPOUSE LECHANI et José MORENO

Composition du jury
M. Yvon FRANÇOIS, Président, CNRS Paris Centre
Mme Aurélie NÉVÉOL, Rapporteure, CNRS Île-de-France
M. Julien VELCIN, Rapporteur, Université Lumière Lyon 2
M. Enrique AMIGO, Examinateur, Universidad Nacional de Educación a Distancia
Mme Lynda TAMINE EPOUSE LECHANI, Directrice de thèse, Université Toulouse III - Paul Sabatier
M. José G. MORENO, Co-directeur de thèse, Université Toulouse III - Paul Sabatier

Membres invités
M. Olivier Ferret, Université Paris-Saclay, CEA, List
M. Romaric Besançon, Université Paris-Saclay, CEA, List

EVALUATION AND INTEGRATION OF STRUCTURAL KNOWLEDGE FOR PRETRAINED LANGUAGE MODELS

JESÚS ENRIQUE LOVÓN MELGAREJO

Institut de Recherche en Informatique de Toulouse
Université Toulouse III – Paul Sabatier

Co-supervisor: José G. Moreno
Supervisor: Lynda Tamine

Contact me for any comments and corrections: jesus.lovon@irit.fr

ACKNOWLEDGMENTS

The journey through this academic adventure, from its beginning to its culmination, has been marked by the invaluable support of an exceptional human group. I have been fortunate to share with them significant moments that have left me with pleasant experiences and transcendental lessons.

Firstly, I wish to express my deepest gratitude to my supervisors, José and Lynda. I thank them for their trust in starting this project together. Their guidance, advice, and constant support in facing the challenges throughout these years, along with their demand and academic rigor, were crucial for achieving the personal goals I had set. Their contribution and assistance have been essential in dealing with life's inevitable highs and lows.

My thanks also extend to Olivier and Romaric for their kindness, for their significant involvement in the progress of this thesis, and for the enriching discussions that helped me advance during the most challenging moments. I also take this opportunity to extend my appreciation to all the other members of the MEERQAT project for the exciting discussions and shared experiences. I want to address special acknowledgment to Hervé Le Borgne for the project coordination and to Paul Lerner for the collaborations and discussions.

I am also thankful to the thesis jury, to the evaluators, Aurélie Névéol and Julien Velcin, for dedicating their time to a detailed analysis of my work, and to the examiners, Enrique Amigó and François Yvon, for their interest in my research and for agreeing to participate in my thesis defense.

Moreover, I would like to thank Gilles Hubert for welcoming me into the IRIS team and all its members for the shared moments and pleasant discussions. A special thanks to Karen and Laure for their invaluable guidance during my master's internships, which stayed with me as I worked on this thesis. I am also grateful to Manuel Castillo-Cara for sparking my interest in research way back in my undergraduate days in Perú.

Thanks also to all the doctoral students with whom I have shared these last years: Maxime, Rafik, Alexandre (*por me ajudar a melhorar meu português*), Maël, Antoine, Nishchal, Mira, Alex, Damien, Malik, Luis, Nicolas, Raphael, among others. A particular shoutout to Nihed and Hanane for their motivating discussions, friendship and their invaluable support in the final phase of this thesis. The same goes for

Khalil, Jules, and Thouria for their hard work during their master's internships and for giving me the experience of mentoring them, which was rewarding. I wish them success in their future endeavors.

Finally, on a more personal level, I want to express my gratitude to Gabi for being my unwavering partner on this journey and supporting me at every moment, both in good times and in difficult ones. Last but not least, from the bottom of my heart, I would like to thank to my friends and family, *especialmente a mis padres, por su amor, apoyo incondicional y confianza en mis decisiones. Por estar presentes, a pesar de la distancia y los momentos importantes que hemos tenido que vivir separados. Su confianza en mí ha sido el pilar de mi fortaleza*.

RÉSUMÉ

Le domaine de la représentation des connaissances est en constante évolution. Grâce aux récents progrès dans les réseaux neuronaux profonds, en particulier l'architecture *transformer*, le domaine du traitement automatique du langage naturel (TALN) a été doté d'outils révolutionnaires conduisant à des performances améliorées sur de multiples tâches de TALN. Les modèles de langue pré-entraînés (PLM), tels que BERT et GPT, qui sont des modèles basés sur des *transformers* entraînés sur d'importantes quantités de données textuelles, ont joué un rôle significatif dans ces avancées.

Les PLMs peuvent produire des représentations contextualisées intégrant des motifs syntaxiques et sémantiques riches du langage. Cependant, ils ne fournissent pas de représentations structurées et factuelles des connaissances, essentielles pour une meilleure compréhension du langage. Pour remédier à ces problèmes, les chercheurs ont exploré la combinaison de PLMs classiques avec des ressources de connaissances externes, telles que les bases de connaissances (KB). Cette approche vise à compléter les PLMs en fournissant les composants structurels et factuels manquants inhérents aux KBs. En résulte l'émergence d'une nouvelle famille de PLM renforcés par la connaissance (KEPLM).

Dans cette thèse, nous nous concentrons sur l'intégration des KBs dans les PLMs, avec un intérêt particulier pour leur structure ou hiérarchie. Nous explorons différentes orientations de recherche visant à améliorer ces PLMs, notamment (i) l'exploration des limitations et des méthodes pour intégrer implicitement les KBs et leur impact sur les tâches basées sur le raisonnement et (ii) la définition de méthodologies d'évaluation pour les signaux hiérarchiques explicites des PLMs et leur transférabilité à d'autres tâches de TALN.

Dans une première contribution, nous proposons de revisiter les méthodes d'entraînement des PLMs pour les tâches basées sur le raisonnement. Les méthodes actuelles se limitent à généraliser cette tâche à différents niveaux de difficulté, traitant chaque niveau comme une tâche différente. Au lieu de cela, nous suggérons une approche incrémentielle d'apprentissage du raisonnement, où le raisonnement est appris progressivement, passant des niveaux de difficulté simples aux niveaux complexes. Cette approche tire parti de composants précédemment négligés qui ne participent pas à la chaîne de raisonnement principale, et nous évaluons si cela améliore la généralisation

de cette tâche. Nous utilisons une méthodologie implicite qui transforme l'information structurée en texte non structuré avec un contenu taxonomique riche. Nous avons également mené des expériences sur des tâches liées au raisonnement, telles que la compréhension de lecture et la réponse aux questions, pour évaluer la pertinence de notre proposition.

Pour notre deuxième contribution, nous visons à améliorer les performances des PLMs en incorporant des signaux hiérarchiques explicites en eux. Alors que diverses approches d'évaluation et d'intégration ont été développées pour les plongements lexicaux statiques, il y a une exploration limitée de ces méthodes pour les plongements lexicaux contextualisés. Les méthodes d'évaluation actuelles pour les PLMs héritent des limitations des évaluations des plongements statiques, telles que les biais des ensembles de données et les signaux hiérarchiques superficiels. Par conséquent, nous proposons une nouvelle méthodologie d'évaluation pour les PLMs qui prend en compte de multiples signaux hiérarchiques. Notre travail caractérise la représentation hiérarchique en la décomposant en distributions hiérarchiques de base que nous appelons propriétés hiérarchiques. Nous évaluons les connaissances hiérarchiques présentes dans les PLMs de pointe en utilisant ces propriétés et analysons si leur apprentissage vise à améliorer les représentations hiérarchiques internes des modèles et leur applicabilité aux tâches de TALN connexes.

ABSTRACT

The field of knowledge representation is a constantly evolving domain. Thanks to recent advancements in deep neural networks, particularly the Transformer architecture, the natural language processing (NLP) field has been provided with groundbreaking tools leading to improved performance across multiple NLP tasks. Pre-trained language models (PLMs), such as BERT and GPT, which are Transformer-based models trained on extensive amounts of textual data, have played a significant role in this progress.

PLMs can produce contextualized representations embedding rich syntactic and semantic patterns of language. However, they do not provide structured and factual knowledge representations, essential for a better understanding of language. To alleviate these issues, researchers explored combining classical PLMs with external knowledge resources, such as knowledge bases (KBs). This approach aims to complement PLMs by providing the missing structural and factual components inherently present in KBs. As a result, this approach has given rise to a new family of knowledge-enhanced PLMs (KEPLMs).

In this thesis, we focus on integrating KBs into PLMs, with a particular interest in their structure or hierarchy. We explore different research directions towards enhancing these PLMs, which include (i) exploring the limitations and methods to implicitly integrate KBs and their impact on reasoning-based tasks and (ii) defining evaluation methodologies for explicit hierarchical signals for PLMs and their transferability to other NLP tasks.

In a first contribution, we propose to revisit the training methods of PLMs for reasoning-based tasks. Current methods are limited to generalizing this task to different difficulty levels, treating each level as a separate task. Instead, we suggest an incremental learning reasoning approach, where reasoning is learned gradually from simple to complex difficulty levels. This approach takes advantage of previously overlooked components that do not participate in the main reasoning chain, and we evaluate whether it improves the generalization of this task. We use an implicit methodology that transforms structured information into unstructured text with rich hierarchical content. We further conduct experiments on reasoning-related tasks such as reading comprehension and question answering to assess the pertinence of our proposal.

For our second contribution, we aim to improve the performance of PLMs by incorporating explicit hierarchical signals into them. While various evaluation and integration approaches have been developed for static word embeddings, there is limited exploration of these methods for contextualized word embeddings. The current evaluation methods for PLMs inherit limitations from static embedding evaluations, such as dataset biases and superficial hierarchical signals. Therefore, we propose a new evaluation methodology for PLMs that considers multiple hierarchy signals. Our work characterizes the hierarchical representation by decomposing it into basic hierarchical distributions that we call hierarchy properties. We evaluate the hierarchical knowledge present in state-of-the-art PLMs using these properties and analyze if learning them aims to improve inner hierarchical representations of the models and their applicability to related NLP tasks.

PUBLICATIONS

International Conference Paper

- Jesus Lovon-Melgarejo, Jose G. Moreno, Romaric Besançon, Olivier Ferret, and Lynda Tamine. 2024. Probing Pretrained Language Models with Hierarchy Properties. In *Advances in Information Retrieval: 46th European Conference on IR Research, Findings of ECIR 2024, Glasgow*. Springer International Publishing.

- Jesus Lovon-Melgarejo, Jose G. Moreno, Romaric Besançon, Olivier Ferret, and Lynda Tamine. 2022. Can We Guide a Multi-Hop Reasoning Language Model to Incrementally Learn at Each Single-Hop?. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING), pages 1455–1466, Gyeongju, Republic of Korea*. International Committee on Computational Linguistics

National Conference Paper

- Jesús Lovón Melgarejo, Jose Moreno, Romaric Besançon, Olivier Ferret, and Lynda Tamine. 2023. Reconnaissance d'Entités Nommées fondée sur des Modèles de Langue Enrichis avec des Définitions des Types d'Entités. In *Actes de CORIA-TALN 2023. Actes de la 18e Conférence en Recherche d'Information et Applications (CORIA), pages 185–194, Paris, France*. ATALA.

International Workshop Conference Paper

- Jesus Lovon-Melgarejo, Jose G. Moreno, Romaric Besançon, Olivier Ferret, and Lynda Lechani. 2023. MEERQAT-IRIT at SemEval-2023 Task 2: Leveraging Contextualized Tag Descriptors for Multilingual Named Entity Recognition. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 878–884, Toronto, Canada*. Association for Computational Linguistics.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

1 INTRODUCTION

1.1 CONTEXT

Grouping things into categories is a fundamental aspect of human thinking that helps us understand the world around us (Plebe, 2011). The idea of classifying things into groups, known as a *taxonomy*, has been around for a long time. It can be traced back to Aristotle, who wrote about how living beings can be classified based on shared characteristics in his works "History of Animals" and "Parts of Animals".

Taxonomies have been useful resources in various fields, including computer science. With the rapid advancements in artificial intelligence and information systems, computational taxonomies have become essential tools for organizing and managing knowledge. Examples of early computational taxonomies include bibliographic retrieval systems like *Medline*, *Mesh* (Medical Subject Heading), and *Roget's Thesaurus* (Mawson, 1911). Later, using taxonomies to structure information retrieval systems, search engines, and knowledge representation models became crucial in improving the efficiency of different applications.

In the field of Natural Language Processing (NLP), structured knowledge sources play a significant role in determining the relationship between different concepts. These sources rely on the notions of *semantic relatedness* or its inverse *semantic distance*, which are crucial for various NLP applications such as text structure determination, word sense disambiguation, summarization, annotation, and information extraction and retrieval (Budanitsky, 1999).

The definition of *semantic relatedness* is often compared with *semantic similarity*. It is important to understand that *semantic relatedness* is a broader concept than *semantic similarity*. It includes not just entities that have similar meanings but also dissimilar entities that can have semantic relatednesses, such as through lexical relationships like meronymy (e.g., car - wheel) and antonymy (e.g., hot - cold) or functional relationships and frequent associations (e.g., pencil - paper, rain - flood). In computational applications, the use of semantic relatedness is more relevant than similarity[1].

There are various ways to measure the level of relatedness between two concepts in NLP. One of the earliest methods was the Bag of

1 Examples are taken from Budanitsky and Hirst (2006)

Words (BoW) approach, which uses dictionary-based resources to determine the level of similarity. Another method is based on edges, particularly based on the number of edges as a distance that separates the concepts when working with taxonomical resources. The Mesh hierarchy introduced an initial edge-based approach by Rada et al. (1989). Later, a different definition of this edge-based distance was proposed, considering factors such as depth (Wu, 1995; Sussna, 1993), and path (Leacock, 1998). In addition to edge-based distances, Resnik (1995) introduced the concept of information-based similarity measures that evaluate the extent to which two concepts share common information (Jiang and Conrath, 1997; Dekang, 1998).

With the rise of the internet and the abundance of information available, taxonomies have become more integrated into computer science. However, creating taxonomies requires expert knowledge and agreement, making it a resource-intensive task. To address this, researchers have developed methods to derive taxonomies directly from textual sources (Wang, He, and Zhou, 2017).

The work from Hearst (1992) observed that organized information, including relationships like *hypernymy*, can be extracted from recurring *patterns* in text. This led to the creation of several taxonomies from large datasets of web pages, such as the Probase dataset (Wu et al., 2012).

The advent of machine learning has brought a significant change in the taxonomy reconstruction from text, moving from *pattern-based* approaches to *distributional-based* methodologies. These new methods determine semantic relatedness by analyzing information extracted from large corpora based on the "distributional hypothesis" (DIH) (Harris, 1954). The DIH assumes that words that are similar in meaning tend to occur together. However, one limitation of the initial approaches is that they consider isolated word representations, ignoring the actual meaning of the words.

The emergence of deep neural networks proposed tools to develop enriched word representation (Chandrasekaran and Mago, 2021). Word embeddings are at the heart of this evolution. They convert words into vector representations that encapsulate the inherent linguistic relationships between them.

Various architectural propositions have been developed based on the representations of linguistic relationships. These include Convolutional Neural Networks (CNN) (Khan et al., 2020), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM) (Yu et al., 2019), and, most recently, Transformers (Vaswani et al., 2017), which include Pre-trained Language Models (PLMs). These models have proven highly effective in deciphering complex

linguistic relationships and can extract knowledge from diverse and extensive plain-data sources. Transformers, in particular, have demonstrated remarkable capabilities in this regard.

Embedding representations are utilized by most hybrid semantic relatedness and deep learning approaches to convert text into high-dimensional vectors. Hence, the efficacy of these embeddings is crucial for the performance of semantic similarity methods (Chandrasekaran and Mago, 2021).

1.2 MOTIVATION

In recent years, Transformer-based models have contributed significantly to the improvement of NLP tasks, especially in natural language understanding tasks such as reading comprehension, question-answering, and semantic similarity tasks. Models like BERT (Devlin et al., 2019) and GPT (Radford et al., 2018), among others, leverage the Transformer architecture and pre-trained language models to encode contextualized vector representations of vast amounts of data. This makes them the basis for developing many current state-of-the-art (SOTA) models in NLP and knowledge representation.

According to current research, it has been found that PLMs encode knowledge in the form of a knowledge base (KB) (Petroni et al., 2019). This allows us to retrieve factual information under certain restrictions, similar to a knowledge base retriever. Particularly, creating input queries using text patterns as an extraction mechanism has proven to be a promising approach toward this goal (Bouraoui, Camacho-Collados, and Schockaert, 2020). This approach is commonly called "prompt engineering" and has been proven helpful in extracting specific information and behaviors from PLMs.

Most research on PLMs has focused on their factual component, assuming they behave as knowledge bases. Therefore, injecting knowledge into PLMs is widely explored to target different applications, such as correcting inaccurate factual information (also known as "hallucinations"). However, the structural component is often overlooked. In this dissertation, we explore different approaches to extracting and improving structural information by injecting it into PLMs using KB information. We analyze different methodologies, explicitly and implicitly injecting structure.

1.3 RESEARCH OBJECTIVES AND CONTRIBUTIONS

This dissertation delves into the emerging topic of knowledge integration for PLMs and its applications in related NLP tasks. As previously mentioned, PLMs are the leading resource for SOTA NLP models. However, they have limitations, as they are not capable of capturing factual and structured knowledge. On the other hand, knowledge resources like KBs explicitly store factual knowledge in a structured form (Pan et al., 2023). Therefore, using KBs to provide this missing information is a promising approach to improving PLM representations.

The literature has widely explored different knowledge integration methods. In our work, we mainly focus on the hierarchical structural characteristics of KBs, the different methods to evaluate them, and their integration in the context of PLMs. Furthermore, we investigate the transferability of the hierarchy knowledge into related NLP tasks, such as multi-hop reasoning, question answering, and taxonomy reconstruction.

Throughout this dissertation, we will address the following questions:

1. Do PLMs naturally encode hierarchical knowledge from pretraining?
2. Can learning hierarchical-based signals help PLMs enhance their representations?
3. Is improving hierarchical representation beneficial for performing better in NLP downstream tasks?

For this dissertation, we have chosen to distinguish between two injection techniques: *implicit* and *explicit* integration methodologies. We aim to address the previous questions by independently applying both techniques, taking into account the benefits and limitations of each.

IMPLICIT INTEGRATION OF KNOWLEDGE.

Implicit integration methodologies typically use templates to transform structural information into text, such as triplets (two entities linked by one relationship). This approach considers transforming a KB into a specialized corpus of taxonomic knowledge that reinforces the structural relations between entities, such as hypernyms or siblings. The textual taxonomic corpus strengthens the co-occurrence

of taxonomic relations of terms found less frequently in typical text resources. Incorporating this taxonomic knowledge into PLMs has been effective in improving the performance in reasoning-based tasks (Richardson et al., 2020), leveraging the default emergent capabilities of PLMs for reasoning (Kassner, Krojer, and Schütze, 2020).

One of the key features of reasoning tasks is the varying difficulty levels of inference, which can be identified by the depth of the reasoning chain. Common approaches conceive different depth reasoning tasks as separated tasks, and models are developed using a multi-task training approach to generalize reasoning. Results using this approach showed that PLMs can handle multiple-depth reasoning simultaneously. However, their ability to generalize on this task is still questionable (Richardson and Sabharwal, 2020). Therefore, better training strategies are required for a better understanding of reasoning.

To handle reasoning generalization, we envision an incremental reasoning framework that is inspired by how humans learn. More specifically, we propose that a model should first abstract more superficial reasoning levels and then gradually learn complex tasks to comprehend deeper reasoning levels.

Our work proposes an incremental reasoning approach that implicitly encodes the reasoning chain from a KB adapted to PLMs. First, we examine the capabilities of PLMs to generalize reasoning tasks using SOTA methods. Then, we explore different training strategies and data generation algorithms to simulate incremental learning. Our approach relies on the contextual information given to PLMs, which is divided into relevant phrases and irrelevant phrases (known as *distractors*). Opposite to other methods that neglect the impact of *distractors* in learning, we propose to use them to improve reasoning skills. Finally, we evaluate whether endowing PLMs with incremental reasoning skills contributes to better performance in related NLP tasks, namely reading comprehension and question answering.

Overall, our results confirm that the context information, particularly the distractors, plays an essential role in the learning process of reasoning in PLMs. In addition, in terms of computing resources, our incremental reasoning strategy proved to require fewer learning steps compared to SOTA models relying on fine-tuning. Also, the experiments demonstrated that our improved models perform better in related NLP tasks, such as reading comprehension and question answering, compared to default PLMs.

EXPLICIT INTEGRATION OF KNOWLEDGE.

Explicit integration methods typically include hierarchical features that implicit integration methods often miss when transformed into text, such as the notion of hierarchy in terms of granularity or directionality .

The explicit integration of hierarchical structure into language models is an active area of research, with most SOTA methods developed in the context of static word embeddings (Chandrasekaran and Mago, 2021). Examples of these approaches involve developing vector spaces similar to a hierarchy or learning taxonomical embedding representations to capture the hierarchical distribution of these representations.

Classical evaluations to assess the effectiveness of these integration methods rely on the performance of tasks such as taxonomy reconstruction and hypernym discovery. These tasks are chosen based on the expected output, which obeys a taxonomic structure representation. However, using these tasks as proxy evaluators does not contribute to elucidating the model understanding of hierarchical knowledge. This is due to the limited superficial features of hierarchy that these tasks require and potential biases hidden in the evaluation process.

For instance, the approach to solving the previously mentioned tasks relies mainly on identifying only one type of taxonomical relationship: hypernymy (Camacho-Collados, 2017). Moreover, by the automatic generation nature of dataset evaluators, some biases analysis showed that models learn to memorize recurrent hypernym classes instead of learning the relationship itself (Levy et al., 2015).

Therefore, there is a latent need for more comprehensive evaluation datasets and methods for hierarchical representations, particularly for language models. We focus on PLMs as they are an important SOTA tool for different NLP applications. Therefore, it is crucial to understand in a comprehensive way insights into their strengths and limitations. Our proposal is to diagnose the hierarchical representations of concepts by characterizing the hierarchy distribution. We do this by decomposing its structure into relevant elements, which we refer to as *hierarchy properties*. These properties are used to assess the hierarchical distribution of models, providing a deeper insight into different taxonomic relationships beyond the hypernymy. Particularly, the properties we suggest extend the explored relationships to hypernymy (parent), ancestor, sibling, and far relationships.

Our evaluation methodology includes a setup proposition for PLMs to perform this evaluation and analyze their representations. Additionally, we investigate whether learning these properties can

enhance the PLM's hierarchical representation. We also examine the impact of these properties not only in the distribution of the model representations but also on NLP classical evaluations tasks such as taxonomy reconstruction and hypernymy discovery.

Our experiments show that PLMs possess some hierarchical knowledge representation, which is particularly useful in distinguishing different granularities of hypernymy-like relationships. However, differentiating between taxonomical relations, such as hypernyms and sibling relationships, suggests a more challenging evaluation. Our results also indicate that it is possible to learn PLM hierarchy properties, which promotes hierarchical representations. Finally, our evaluations on the impact of these representations are beneficial for hierarchy-related tasks but not as helpful for other tasks less related to hierarchy.

1.4 THESIS OUTLINE

This dissertation is structured into four parts, comprising two main parts, a conclusion and an appendix. The first part, titled "Background and State of the Art", introduces fundamental concepts, notions, and important scientific contributions related to different knowledge injection strategies used to enhance language models. The second part, titled "Contributions", provides a detailed explanation of the contributions made by this thesis.

- Part I: Background and State of the Art

  This part provides an overview of the current methods used to incorporate external knowledge sources into language models. To achieve this, we have divided these methods into two chapters.

  Chapter 2 reviews the techniques for integrating knowledge into PLMs, including evaluation approaches. This chapter begins by introducing various knowledge sources and their classification. Similarly, we introduce some basic concepts regarding PLMs and the pre-training tasks. Then, it surveys different methodologies for integrating these knowledge sources. This survey is mainly categorized into implicit and explicit methods. Finally, we review some of the primary evaluation approaches employed by the community and introduce some of the recent work in the realm of Large Language Models.

  Chapter 3, on the other hand, focuses on methods oriented to integrating the hierarchical structure of knowledge into language

models. In this chapter, we review some of the leading works that concentrate on taxonomic knowledge and other hierarchical features from knowledge bases. We start by reviewing some of the primary works that use static word embeddings, and then we move on to current approaches that use PLMs.

- Part II: Contributions

  This part presents our contributions. We present this part divided into two chapters, each corresponding to a specific contribution that emerged in the context of this dissertation (Lovón-Melgarejo et al., 2022; Lovón-Melgarejo et al., 2024). Each of the following chapters is an extended version of our original publications.

  Chapter 4 presents our first contribution exploring the implicit integration methodologies in the context of reasoning tasks. This chapter first introduces the context of our proposition and its contributions. Then, we present our methodology, including a formal representation of multi-hop reasoning and the algorithms used for our approach. Next, we introduce the experimental setup to present our results and discussions.

  Chapter 5 introduces our second contribution in the context of explicit integration methodologies. The chapter starts by introducing the motivation and contributions of our study. We then describe the methodology used to evaluate and integrate our approach. This is followed by a detailed explanation of the experimental setup, the results obtained, and the ensuing discussions.

- Conclusion

  As a conclusion to our dissertation, we summarize our findings, revisit our contributions, and propose future directions for further exploration of this work.

- Appendix

  We have included three appendices in our work. Appendix A presents technical details about the Bi-encoders model that we used in our work. Appendix B details the computing infrastructure and budget from our first contribution. Appendix C shows complementary results from our second contribution.

Part I

BACKGROUND AND STATE OF THE ART

2

KNOWLEDGE INTEGRATION IN LANGUAGE MODELS

2.1 INTRODUCTION

Over the years, language model development has been primarily focused on enhancing semantic embeddings. Initially, models like Skip-Gram (Mikolov et al., 2013) and GloVe (Pennington, Socher, and Manning, 2014) were used, but their performance was limited by the shallow networks they employed. With the advent of deep learning models like BERT (Devlin et al., 2019), a significant shift arrived in the domain due to the impressive performance of these approaches on different Natural Language Processing (NLP) tasks. These models rely on the Transformer architecture (Vaswani et al., 2017) using an attention mechanism, removing the recurrence and convolutions that previous architectures, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) used. BERT leveraged the power of self-supervised learning to move beyond traditional supervised learning methods and towards Pre-trained Language Models (PLMs).

PLMs stack multiple Transformers layers and are pre-trained on large textual datasets to learn contextualized word representations. This setup helps capture rich semantic relationships and linguistic nuances. The pre-training process involves exposing the model to various language tasks, enabling it to develop a deep understanding of language structures and patterns. For example, in the case of BERT, these tasks consist of Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

Despite their success in various NLP tasks, PLM's learning approach relies explicitly on encoding language-level features and overlooks knowledge-level features (Zhang et al., 2019). At a language level, PLMs encode word representations based on token representations and positional encodings. However, at a knowledge level, PLMs are not explicitly aware of different levels of language expressions. For example, they do not consider different token representations for entities (such as referring to the city of Paris as "Paris" or "the capital of France") and the underlying entities' interactions. This missing feature may generate factual errors during inference over this information, reflecting their inability to recall and reason accurately.

To address these limitations, the research community has introduced the notion of Knowledge-Enhanced Pre-trained Language Models (KEPLMs). These models incorporate external knowledge, such as that derived from Knowledge Bases (KBs), to enhance their interpretability, robustness, and reasoning capabilities. Section 2.2 introduces the most common external knowledge resources in this domain and its components.

The realm of KEPLMs is constantly evolving, with new methods emerging to leverage different features and applications of knowledge. In Section 2.3, we first introduce PLMs, the base of KEPLMs. Then in Section 2.4, we survey some of the primary methodologies employed in injecting knowledge into PLMs. This survey considers two main injection approaches based on how the KB resources are used: implicit and explicit knowledge incorporation. Implicit methodologies leverage the factual information provided by KB, whereas explicit methodologies leverage the structural relationships between entities.

Although these methods have shown state-of-the-art (SOTA) performance on different applications, it is still important to assess the effectiveness of the knowledge injection methodology. Evaluations for KEPLMs mainly aim to answer the following questions: "What information did my model learn?" and "How is this information helpful in NLP applications?". Section 2.5 introduces two main categories of evaluation approaches: knowledge capacity and task diagnostic. To answer the first question, knowledge capacity methods aim to verify and measure the amount of knowledge injected into the model. Task diagnostics evaluation decomposes a target task into measurable elements to answer the second question.

Finally, with the current trend of developing larger models by increasing the number of layers and parameters, Section 2.6 briefly presents some of the latest works leveraging external knowledge in Large Language Models (LLMs) to enhance their performance.

2.2 KNOWLEDGE AS RESOURCE FOR LANGUAGE MODELS

2.2.1 Knowledge

The definition of "knowledge" refers to familiarity, awareness, or understanding of a person or thing, such as facts, skills, or objects (Yang et al., 2021). According to Krathwohl (2002), there are four types of knowledge: factual, conceptual, procedural, and metacognitive. More-

over, knowledge is often represented using various methods, such as first-order predicate logic, frame representation (Minsky, 1974), script representation (Tomkins, 1978), semantic network representation (Quillian, 1966), and ontology representation (Elkan and Greiner, 1993). In this dissertation, we will mainly focus on ontology knowledge representation; more specifically, we will use a knowledge graph as an adaptable resource of this representation for different NLP applications.

KNOWLEDGE GRAPH (KG). We consider a KG as a formal and structured representation of knowledge containing information about entities, their attributes, and their relationships. In a KG, entities are nodes, attributes or properties are associated with nodes, and relationships are edges that connect two nodes. A *triplet* represents three elements of the KG consisting of two nodes and one relationship, commonly represented as (head, relationship, tail), where head and tail are nodes, and the relationship is an edge. The set of triplets forms the KG that captures the semantic relationships and attributes between entities in a structured manner. It is worth emphasizing that most of the existing KGs only contain conceptual and factual knowledge, as procedural and metacognitive knowledge is more complex to represent adequately[1].

2.2.2 Knowledge Sources

According to previous research (Pan et al., 2023; Yang et al., 2021), knowledge resources used to improve PLMs are classified based on their content and integration mechanisms. Based on the format of their content, we can identify two different types: *multi-modal* knowledge sources and *only text-based*, which is our primary exploited resource type in this dissertation.

Multi-modal knowledge graphs are a powerful tool for representing facts across various modalities, including images, sounds, and videos. Due to the integration of textual information with other forms of data, it enables a more comprehensive understanding of complex phenomena (Pan et al., 2023). Some examples of multi-modal knowledge graphs are Richpedia (Wang et al., 2020), WebQA (Chang et al., 2022), and ViQuAE (Lerner et al., 2022), which integrate both text and image information.

Only text-based resources can further be classified into three main categories based on the stored information: *encyclopedic*, *commonsense*,

1 The literature used the terms KBs and KGs interchangeably. In this dissertation, we adopt the same use to refer to this definition of knowledge graph.

and *domain-specific*. In the following, we outline the principal knowledge resources used in the literature to enhance language models according to their information.

Encyclopedic Knowledge

Encyclopedic knowledge is a shared resource representing general knowledge about the world. It is created by integrating data from different sources, including human experts, encyclopedias, and databases. Typically, encyclopedic knowledge is stored in textual formats or using *triplets*. One of the most notable text-based resources is Wikipedia[2], a multilingual encyclopedia featuring millions of pages in over 330 languages. Wikipedia pages are dedicated to different concepts and include textual details, tables, and infoboxes related to an unambiguous concept.

A widely used example of the KG resources is Wikidata (Vrandečić and Krötzsch, 2014), which is a project operated directly by the Wikimedia Foundation, aiming to fully structure the information on Wikipedia. Wikidata is a document-oriented semantic database based on items representing a topic with a unique identifier. Similarly, the Wikidata5M dataset (Wang et al., 2021b), which is a variant of Wikidata, includes *triplets* and high-quality *entity* and *relation descriptors* commonly used to enhance their representations. Another KG is SHINRA (Sekine, Kobayashi, and Nakayama, 2018), which proposes to create a structured knowledge base of Wikipedia, including the items and the attributes, aligning both elements on the ENE ontology (Sekine, 2008). Figure 2.1 shows examples of SHINRA. There are also other KGs such as Freebase (Bollacker et al., 2008), Dbpedia (Bizer et al., 2009), CN-DBpedia (Xu et al., 2017), and YAGO (Suchanek, Kasneci, and Weikum, 2007) that rely on Wikipedia. From these options, CN-DBpedia is available in English and Chinese.

Commmosense Knowledge

Commonsense knowledge relates to people's everyday understanding of the world and their surroundings, including the objects, events, and relationships between them. This knowledge has been helpful in training models for commonsense question-answering tasks, among others (Hu et al., 2023). Unlike encyclopedic knowledge, commonsense knowledge comprises tacit knowledge extracted from text. For instance, a *triplet* from a commonsense KB might be <having no food, CauseDesire, go to a store>, while an example of encyclopedic knowledge might be <Emmanuel Macron, president-of, France>.

2 https://www.wikipedia.org/

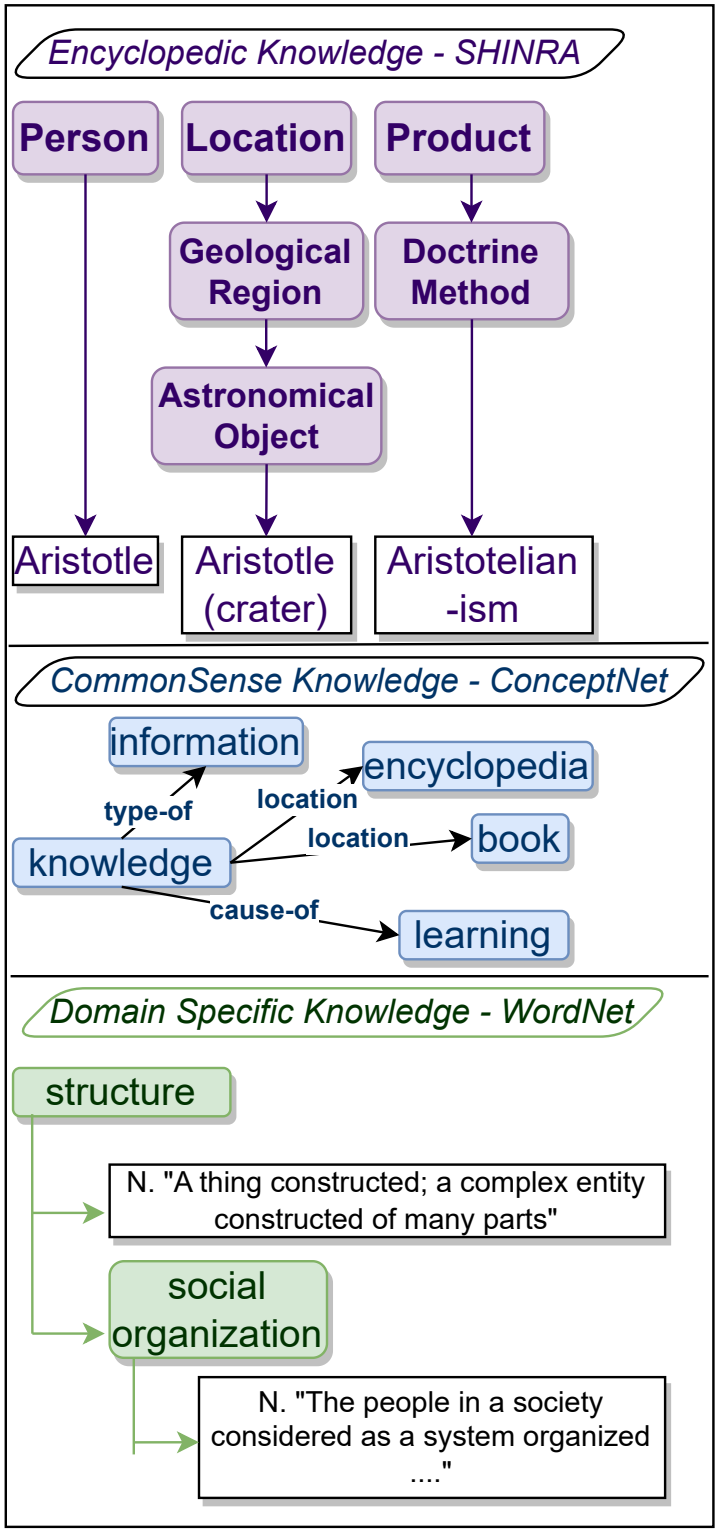


Figure 2.1: Examples of Encyclopedic Knowledge, Commonsense Knowledge, and Domain specific knowledge from the KB SHINRA, ConceptNet, and WordNet, respectively.

Two commonsense KGs that are frequently referenced in the literature are ConceptNet (Speer, Chin, and Havasi, 2017) and the ATOMIC KG (Sap et al., 2019). ConceptNet is an open-domain commonsense KG that includes 34 relations, including but not limited to RelatedTo, IsA, and Causes. Figure 2.1 shows a sample of ConceptNet. In contrast, ATOMIC diverges from ConceptNet by emphasizing inferential knowledge structured in an "if-then" structure, such as "If X pays Y a compliment, then Y will likely return the compliment".

There exist other KGs, such as ATOMIC$^{20}_{20}$ (Hwang et al., 2021) and ASER (Zhang et al., 2020), which cover more accurate and diverse commonsense knowledge, capturing more complex commonsense knowledge relations. Additionally, there are knowledge graphs like TransOMCS (Zhang et al., 2021) and CausalBank (Li et al., 2021) that are automatically constructed.

Domain specific Knowledge

In contrast to the non-specificity of encyclopedic knowledge, domain-specific knowledge refers to specialized knowledge in a particular field, such as sentiment analysis, semantics, biomedical, e-commerce, and finance. Although domain-specific knowledge graphs are typically smaller in scale, they are more precise and dependable.

For instance, WordNet (Miller, 1995), a domain-specific knowledge graph, is utilized for semantics, more precisely for *word sense disambiguation*. The *synset*, which is the core component of WordNet, represents a unique concept that can be expressed through nouns, verbs, adjectives, or adverbs and is composed of one or more lexicalizations, i.e., synonyms used to express the concept. Consequently, a word can belong to multiple *synsets*, denoting its different meanings. A sample in Figure 2.1. Similarly, UMLS (Bodenreider, 2004) is a domain-specific knowledge graph in the medical field, which contains biomedical concepts and their relationships.

2.2.3 Structural Knowledge Components

In addition to the information type, the element component used to work with knowledge sources is another crucial dimension. A knowledge graph is a big data structure that can be decomposed in different ways. From prior work, three formats for encoding emerge: entity, relation, and subgraph.

Entity Knowledge

Entities are crucial in KG resources as they are directly connected to other entities through relationships and indirectly through information gathered from their neighbors. There are two possible methods to encode an entity: *entity embeddings* and *entity descriptors*.

Entity embeddings are obtained by KG embedding methods such as TransE (Bordes et al., 2013), TransH (Wang et al., 2014), or RoTate (Sun et al., 2018). These embeddings are suitable for encoding the neighboring nodes of entities in the KG but are typically incompatible with text. Additionally, a significant challenge of this method is updating the vector spaces each time the KG is updated.

On the other hand, *entity descriptors* represent an entity by a textual definition. Then, a language model can provide a vector representation of this entity through this definition. However, given that we pick and convert into text isolated elements from the KG, we risk to lose critical structural information about entities in their original format.

Relation Knowledge

In a KG, the relation knowledge is represented by the link between two nodes. As previously stated, the interaction between the relation and the two nodes is represented as a *triplet*. Similar to entity embeddings, we find embedding and textual encodings for this element.

Different KG embedding methods have expanded the representation space by defining dedicated spaces for relations and entities to model their interactions better, like TransR (Lin et al., 2015).

In language models, *triplets* are commonly transformed into coherent sentences using text templates. These templates integrate KG sources with an underlying structure in the form of text. The templates can be manually (Petroni et al., 2019; Talmor et al., 2020a) or semi-automatically generated (Bouraoui, Camacho-Collados, and Schockaert, 2020) and are discussed in Section 2.5.1.1.

Subgraph Knowledge

Knowledge graphs consist of subgraphs that are integral components of the graphs. Subgraphs are formed by selecting a group of entities as nodes and using relations as edges. The subgraphs help to prevent the loss of structured information that can occur when dealing with isolated *triplets*. The subgraph includes immediate relations between nodes (1-hop) and extends beyond to cover more distant connections (k-hop).

Subgraphs are typically encoded using Graph Neural Networks (GNNs) (Hamilton, Ying, and Leskovec, 2017; Battaglia et al., 2018) that naturally captures the inner graph structure of these representation. On the contrary, textual-based approaches typically encode subgraphs relying on the *triplet* representation, concatenating them or using random-walk through the subgraph to partially capture the structure.

2.3 PRE-TRAINED LANGUAGE MODELS

In this section, we briefly recall some basic definitions of PLMs that will be used later in this work.

A Pre-trained Language Model (PLM) is a deep learning model for natural language processing tasks. Its main characteristic is that it undergoes a *pre-training* phase on a large corpus of diverse textual data before being *fine-tuned* for specific language-related applications. PLMs are often built using transformer architectures and can learn contextualized word representations that capture complex linguistic patterns, semantic relationships, and syntactic structures.

In the following, we will provide a brief overview of the MLM and NSP pre-training tasks and some of the most common adaptation setups for NLP tasks.

2.3.1 Pre-training tasks

Pre-training tasks for PLMs involve training them on diverse textual data before fine-tuning them for specific tasks. The main objective of these pre-training tasks is to equip the model with a deep understanding of language structures, context, and relationships. This provides the model with versatile representations that can be used for various downstream natural language processing (NLP) applications.

In the following, we introduce two widely used pre-training tasks: MLM and NSP.

Masked Language Modeling (MLM)

During the pre-training phase, many traditional PLMs incorporate the MLM task (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019), which uses unannotated text from large corpora. In this task, the PLM is exposed to multiple sentences, and a randomly chosen subset of tokens is masked in each sentence.

The MLM training task is similar to *denoising auto-encoders* (Vincent et al., 2008), and its objective is to predict the original input by

focusing on the masked tokens, leveraging the PLM token representations. The learning process is guided by a cross-entropy loss, which influences all parameters within the model. It is important to note that although all input tokens contribute to the process, the learning process only utilizes the masked tokens.

More specifically, given a sequence of tokens $X = (x_1, x_2, ..., x_n)$, a PLM prepares first an input sequence. This input sequence is typically the aggregation of token embedding, positional embedding, and segment embedding (in the case of BERT (Devlin et al., 2019)). The model then computes L layers of d-dimensional contextual representations by successively applying non-linear functions, such as multi-headed self-attention layers. Finally, the last layer outputs contextualized representations of the tokens denoted as $Y = \{y_1, y_2, ..., y_n\}$ with $Y \in \mathbb{R}^{n \times d}$.

For illustration purposes, we define the step-by-step process of the MLM task:

1. The token x_i is projected from the one-hot representation into the PLM representation space using an external mapping matrix, $W \in \mathbb{R}^{d \times D_w}$, where d is the dimension of the PLM representation space, and D_w the vocabulary size.

2. An inverted mapping matrix is obtained to transform the contextualized represention into a word score vector for a masked word: $w_i = W^T \times y_i$.

3. The model learns the task comparing "w_i" with the one-hot vector of "x_i". The training objective of the MLM task is to have "w_i" representations as close as possible to the one-hot vector corresponding to the original input. We achieve this by using the cross-entropy loss between the softmax of the word score and the one-hot vector.

Finally, the MLM loss function is formally expressed as:

$$P(y_t|Y) = \frac{\exp(y_t)}{\Sigma_{i=1}^{n} \exp(y_i)} \qquad \text{(softmax function)} \tag{2.1}$$

$$\mathcal{L}_{MLM} = -\sum_{y_t \in M} \log P(y_t|Y) \tag{2.2}$$

where M is a subset of randomly selected tokens from the input to be masked out.

Next Sentence Prediction (NSP)

Next Sentence Prediction (NSP) is a binary classification task employed to predict whether a given pair of sentences in a document are contiguous or not. This task helps the model capture relationships between sentences and understand document-level context.

Formally, we represent two token sequences as $X^1 = (x_1^1, x_2^1, ..., x_n^1)$, and $X^2 = (x_1^2, x_2^2, ..., x_n^2)$. Then, the model PLM with a classification layer NSP on top represented as $\texttt{NSP}((\texttt{PLM}(X^1, X^2))$ will determine whether or not X^1 and X^2 are contiguous.

The model uses cross-entropy loss to calculate the NSP loss for each sentence pair:

$$\begin{aligned}\mathcal{L}_{\text{NSP}} = &- (y \times \log P(\text{Next}|X^1, X^2) \\ &+ (1 - y) \times \log(1 - P(\text{Next}|X^1, X^2)))\end{aligned} \tag{2.3}$$

where Next is a binary indicator variable obtained from the classification layer, and y is the binary label indicating whether X^1 and X^2 are consecutive.

2.3.2 Adaptation to specific NLP tasks

We present four different ways to adapt PLMs for specific NLP tasks according to the literature.

FINE-TUNING. Fine-tuning further trains a PLM on a specific task or domain to enhance its performance on that particular objective.

FEW-SHOT. Few-shot relies on training models on only a few examples and with limited training steps to test the adaptability of the general representations of a model to a specific new task.

INOCULATION. "Model inoculation" refers to the process of training models on new challenging tasks with only a few examples. The aim is not to completely repurpose the model, as it is in ordinary fine-tuning, but to enhance specific phenomena that may deviate from the model's original training distribution (Liu, Schwartz, and Smith, 2019). The primary distinction between Inoculation and few-shot learning approaches is the subset samples' difficulty level. Few-shot learning has no specific guidelines to filter the subset of the training set. At the same time, Inoculation focuses on searching for challenging samples to rapidly adjust the model to these challenges and the standard task.

ZERO-SHOT. Zero-shot refers to the scenario where a PLM performs tasks that it has not been specifically trained on. In other words, the model is expected to use its learned knowledge to generalize to new classes or tasks not part of the pre-training phase.

The main difference between these approaches lies in the amount of data the model uses for specialization. For instance, in a dataset consisting of 1000 training samples, a fine-tuning approach would use the complete dataset. On the other hand, the inoculation and few-shot techniques would only utilize a certain percentage, such as 10%. Inoculation will prioritize the "challenging" samples. The zero-shot method, however, would not use any of the training samples.

2.4 KNOWLEDGE ENHANCED PRE-TRAINED LANGUAGE MODELS

In the previous section, we introduced PLMs mainly trained on large amounts of unstructured text data such as Wikipedia and BookCorpus (Zhu et al., 2015). These models commonly use the pre-training MLM task to refine their token representations by capturing the contextual and semantic information embedded within these corpora. The MLM task uses the token information extracted from the input as a basis. This approach considers all tokens of the same type without explicitly distinguishing the language features, overlooking the valuable information in entities and phrases within the source text.

To overcome this limitation, researchers have started integrating complementary knowledge sources, such as knowledge bases, leading to the development of enhanced PLMs, referred to as KEPLMs.

A common feature among KEPLMs is the use of injection methodology to incorporate KBs into PLMs. This injection methodology seeks to integrate the explicit and structured knowledge from a KB into a PLM to enhance the knowledge representations. The realm of KEPLMs has quickly expanded with multiple works proposing different approaches to inject knowledge sources information into PLMs (Wei et al., 2021; Pan et al., 2023; Yang et al., 2021; Hu et al., 2023). This section will survey some of the most relevant works in this domain.

We review these works, classifying the methods into two approaches: *implicit* and *explicit knowledge incorporation*. *Implicit knowledge incorporation* leverages contextual information by exploiting text-based details within KBs, explicitly focusing on entities and relations as factual elements, essentially treating the KB as an extension of the training corpus. While these *implicit methods*

effectively capitalize on contextual signals, they risk overlooking the structural nuances within the KB.

On the other hand, *explicit knowledge incorporation* takes a more direct and structured approach by injecting information directly from KBs. These methods establish explicit associations between entities, fostering a more informed and contextually relevant understanding. Moreover, the *explicit* approach helps to avoid any potential issues that may arise from ignoring important structures in implicit methods.

2.4.1 Implicit KEPLMs

In the following, we present the relevant works mainly categorized on the integration level used: at a token level, span level, triplets in a text format, and mixing embeddings.

2.4.1.1 *Token-level Integration*

The literature has explored integrating different language features using the token representation to enhance the knowledge present in PLMs. Some of these language features are part-of-speech (POS), sentiment labels at sentence and word levels, word sense, entity mentions and spans, and relations.

The early work from Ke et al. (2020) introduces the SentiLare model, which integrates the POS elements into PLMs for sentiment classification. The integration uses a Label-Aware Masked Language Model (LA-MLM) task, a variation of the original MLM. The LA-MLM task predicts the sentiment polarity for each token from the input and the sentiment label from the global input sentence at the same time. This approach leverages the implicit relationships between the sentence label and the input.

Formally, SentiLare is trained using a combined loss $\mathcal{L} = \mathcal{L}_{\text{LA-MLM}} + \mathcal{L}_{\text{MLM}}$, where $\mathcal{L}_{\text{LA-MLM}}$ is defined as the sum of the loss predicting token-level sentiment polarity, $\mathcal{L}_{\text{token}}$, and the loss predicting the sentence sentiment, $\mathcal{L}_{\text{sent}}$.

$$
\begin{aligned}
\mathcal{L}_{\text{LA-MLM}} =& \mathcal{L}_{\text{token}} + \mathcal{L}_{\text{sent}} \\
\mathcal{L}_{\text{token}} =& -\sum_{t=1}^{n} m_t \cdot [\log P(y_t|\hat{X}, l) + \log P(\texttt{pos}_t|\hat{X}, l) + \\
& \log P(\text{polar}_t|\hat{X}, l)] \\
\mathcal{L}_{\text{sent}} =& -\log P(l|\hat{X}) - \sum_{t=1}^{n} m_t \cdot [\log P(y_t|\hat{X}) + \log P(\texttt{pos}_t|\hat{X}) + \\
& \log P(\text{polar}_t|\hat{X})]
\end{aligned}
\tag{2.4}
$$

where m_t is an indicator function and equals to 1 iff the input token x_t is masked, and $\hat{X}$ is the extended input X with the POS and word-polarity for the SentiLARE model. At the token-level $P(y_t|\hat{X}, l)$, $P(\texttt{pos}_t|\hat{X}, l)$, and $P(\text{polar}_t|\hat{X}, l)$ use the last hidden state, and conditioned to the sentiment label l. At the sentence-level, the probability $P(l|\hat{X})$ is based on the last output of the [CLS] token.

Similarly, the SenseBERT model (Levine et al., 2020) proposes incorporating word-sense information as knowledge into PLMs. The use of word-sense information is particularly useful to leverage the representation of low-frequency words, otherwise known as *rare words* (Schick and Schütze, 2020).

SenseBERT utilizes the ability of a PLM to predict the meanings of missing words, along with a sense-aware mechanism to extract more information from *rare words*. Sense-BERT captures sense embedding using a semantic-level training loss called $\mathcal{L}_{\text{SLM}}$ which is added to the regular training of BERT as $\mathcal{L} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{NSP}} + \mathcal{L}_{\text{SLM}}$. We formally define $\mathcal{L}_{\text{SLM}}$ as:

$$
\mathcal{L}_{\text{SLM}} = -\log \sum_{s \in A(w)} P(s|X) + \mathcal{L}_{\text{reg}} \tag{2.5}
$$

This loss corresponds to predict the correct meaning from a given word, where $A(w)$ denotes the possible meaning (*supersenses*) of a given word w, X the input context, $\mathcal{L}_{\text{NSP}}$ the next sentence prediction loss, , and $\mathcal{L}_{\text{reg}}$ is a regularization term to reduce noise for the high frequented supersenses labels.

Entities are another widely explored feature in the language. A limitation of the MLM task's default word-collocation prediction is that this task does not consider the relationships between entities. This results in PLMs struggling to extract entity and phrase knowledge solely from the context.

To illustrate, let us consider the sentence: "Harry Potter is a series of fantasy novels written by J.K. Rowling". A PLM typically represents

this sentence as "Harry UNK is a series of fantasy novels written by UNK UNK.". A PLM could easily predict the token "Potter" given "Harry" without using context. However, this approach fails to capture the high-level relationship between entities "Harry Potter" and "J.K. Rowling".

ERNIE-Baidu[3] (Sun et al., 2019), referred to as ERNIE 1.0 from now on, proposes a modified masking process. The training data for ERNIE 1.0 is prepared by first identifying the entity mentions in the input sentence and then masking the consecutive tokens of these mentions. Later models, namely ERNIE 2.0 (Sun et al., 2020b) and ERNIE 3.0 (Sun et al., 2021), were subsequently introduced to explore different training techniques to improve performance.

In the same context, ERNIE-Tingshua (Zhang et al., 2019) (referred to as ERNIE) introduces a novel approach to combine contextual and entity embeddings representation into BERT. The authors propose an MLM-based task-oriented to entities, called *denoising entity auto-encoder* (dEA). ERNIE first aligns and fuses entity embeddings and contextual embeddings based on the entity mentioned in the text. Then, based on these fusion representations, the dEA task will randomly mask some entity tokens to predict the corresponding entities later.

Similar to previous models, ERNIE incorporates the dEA task alongside the existing pre-training tasks, MLM, and NSP. For more details on the ERNIE architecture, please refer to Section 2.4.2.2.

2.4.1.2 *Span-representation Integration*

Representing an element in a KB through text can be challenging due to the need to find the correct textual representation. An entity element in a KB can be expressed in different ways, which can be encoded on single or multiple tokens. For example, the entity "Harry Potter" could be represented in the text as "Harry Potter", 'Harry James Potter" or "The Boy Who Lived". Despite the improved performance of previous injection methods on different NLP tasks, they still risk producing poor representations for *entity span representations* and *relation span representations*.

Recent work suggests integrating knowledge at a higher level to improve the representations obtained by these models. Several state-of-the-art (SOTA) models incorporate explicit span representations to capture different knowledge elements in PLMs. Generating informative span representations of entities and relations can benefit multi-

3 In the literature, there exist two homologous PLMs named ERNIE. To differentiate between them, the community refers to them by adding the research groups to which they belong: ERNIE-Baidu and ERNIE-Tinghua.

ple downstream tasks (Li et al., 2022). These span representations can take the form of aggregation methods for multi-token mentions or involve modeling the interaction between conceptual representations.

ENTITY SPAN REPRESENTATION.

We begin by considering the case where no entity representation is available. This case occurs when dealing with unseen entities from training which lack meaningful representations. To tackle this problem, the KEPLER model (Wang et al., 2021b) proposes a solution by encoding textual mentions of entities in a KB embedding space representation. KEPLER uses *entity descriptors* to obtain textual descriptions of entities, such as their definitions. The PLM encodes this conceptual representation of the entities and then adapts them to a KB embedding representation space.

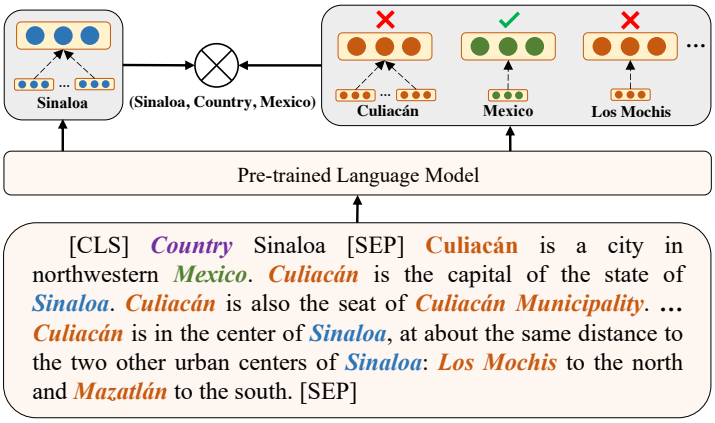
Specifically, KEPLER obtains embeddings from a *triplet* in a KB, $(h, r, t) \in G$, where h and t are entities, and r is the relation between them. The input is prepared as $(\mathbf{h}, \mathbf{r}, \mathbf{t})$, where each element is replaced by its corresponding *textual descriptor*. These embeddings are calculated as the [CLS] projection of the output layer of a PLM. Then, the loss function is formulated as: $\mathcal{L} = \mathcal{L}_{KE} + \mathcal{L}_{MLM}$, where $\mathcal{L}_{KE}$ denotes a knowledge embedding loss, inspired in the RotatE model (Sun et al., 2018).

Another limitation in entity representation concerns the need for multiple tokens to represent a single concept. The LUKE model (Yamada et al., 2020b) proposes a solution considering both words and entities are treated as independent types of tokens. The model's strength lies in its input representation, which includes an *entity type embedding* denoting a token, in addition to the traditional token and position embeddings, and an *entity-aware self-attention mechanism* managing both word and entity tokens. This enables the computation of intermediate and output representations for all tokens and entities.

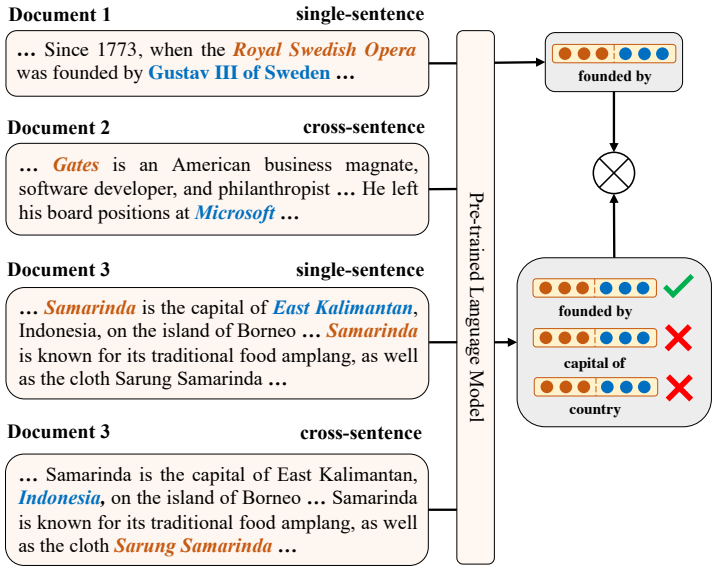
To pre-train LUKE, the model uses an extension of MLM, where entities are randomly masked, and the model is trained to predict the originals of these masked entities.

RELATION SPAN REPRESENTATION.

KEPLER and LUKE introduced additional factual knowledge into PLMs and improved entity representations. However, the complex and explicit relationships between entities remain challenging to encode due to their different mention possibilities (Yang et al., 2021).



(a)



(b)

Figure 2.2: Input examples for the pre-training tasks for ERICA. (a) *Entity Discrimination*. It represents the tail entity (*Mexico*) closer than other entities. (b) *Relation Discrimination*. Example for the relation "founded by" at sentence and paragraph (cross sentence) levels. Source (Qin et al., 2021)

Recent works aim to address this challenge by extending the encoding of information beyond the entity level in KBs to capture relations between entities and text at the sentence and paragraph levels. One of the earliest approaches in this domain is ERICA (Qin et al., 2021), which uses *contrastive learning* to enhance entity relations in PLMs. ERICA introduces two novel pre-training tasks to achieve this goal: *entity discrimination* and *relation discrimination*.

The *Entity Discrimination* (ED) task involves identifying which entity can be inferred from a given head entity and relation. This task enhances the understanding of each entity by considering its relations to other entities in the text. The task involves analyzing a *triplet* (e_i, r, e_j) with elements present in the token sequence $X = \{x_1, ..., x_n\}$ and its aligned entity sequence $\{e_1, e_2, .., e_m\}$. The goal is to maximize the posterior probability $P(e_j|e_i, r, X)$ through contrastive learning based on the work of (Hadsell, Chopra, and LeCun, 2006). The learning function seeks to obtain a closer representation between a positive pair, (e_i, e_j), compared to negative pairs. Formally, the ED loss can be expressed as:

$$\mathcal{L}_{ED} = -\sum_{(e_i,r,e_j)\in G} \log \frac{\exp(\cos(\mathbf{e}_i, \mathbf{e}_j)/\tau)}{\sum_{l=1,\, l\neq j}^{m} \exp(\cos(\mathbf{e}_i, \mathbf{e}_l)/\tau)}, \tag{2.6}$$

where $\cos(\cdot,\cdot)$ denotes the cosine similarity between two entity representations and τ (temperature) is a hyper-parameter. Figure 2.2a shows an example for the *triplet (Sinaloa, Country, Mexico)*.

The *Relation Discrimination* (RD) task aims to measure how closely related two distinct relations are using document-level distant supervision, which is different from the traditional sentence-level approaches. This approach improves relation representations by enhancing the capture of complex reasoning chains in real-world contexts.

In the RD approach, ERICA represents a relation as the concatenation of entity representations, $r_{ij} = [e_i, e_j]$. Similar to the ED approach, RD loss aims to bring closer positive relations (r_{ij}, r_{kl}), where $r_{ij} = r_{kl}$, and separate negative relations, where $r_{ij} \neq r_{kl}$. RD is formally described as:

$$\begin{aligned} \mathcal{L}_{RD} &= -\sum_{(\mathbf{r}_{ij},\mathbf{r}_{kl})\in\mathcal{T}^+} \log \frac{\exp(\cos(\mathbf{r}_{ij}, \mathbf{r}_{kl})/\tau)}{\mathcal{Z}}, \\ \mathcal{Z} &= \sum_{r_C\in\mathcal{T}/\{r_{ij}\}}^{N} \exp(\cos(\mathbf{r}_{ij}, \mathbf{r}_C)/\tau) \end{aligned} \tag{2.7}$$

where N is a hyper-parameter, $\mathcal{T}$ is the set of entity pairs with no relation in the dataset, and $\mathcal{T}^+$ the set of entity pairs with an annotated relation. Figure 2.2b shows an example for the relation "founded by" across entities at sentence and paragraph levels. Finally, ERICA is trained with the combined loss $\mathcal{L} = \mathcal{L}_{ED} + \mathcal{L}_{RD} + \mathcal{L}_{MLM}$.

A drawback in the ERICA approach lies in the use of average pooling from tokens as entity representation. This method is ineffective

in representing entities, especially for tasks requiring entity boundaries such as joint entity and relation extraction. SPOT (Li et al., 2022) addresses this limitation by enhancing the representation of entities and relationships. It achieves this improvement by focusing on token spans and span pairs within the text and with fewer parameters than conventional approaches. SPOT consists of three components: a *textual encoder*, a *span encoder*, and a *span pair encoder*. It uses three pre-training tasks to capture knowledge at three levels: token, entity, and relation.

Similarly to ERICA, SPOT uses the standard MLM approach at the token level and predicts entities based on the entity representation $\mathbf{e}_i$ generated by the *span encoder* at the entity level. However, due to the substantial number and imbalanced distribution of entities, the entity-level loss, denoted as $\mathcal{L}_{\text{ENT}}$, is computed using an adaptive softmax with log-likelihood during pre-training. This loss is defined as the negative log probability of gold entities e^* in the training documents $\mathcal{D}$.

At the relation level, the model predicts the relationships between entity i and entity j based on the entity pair representation $\mathbf{r}_{ij}$. Like tokens and entities, the log-likelihood calculates the probability of the relation between two entities. The loss for the relation, which is denoted as $\mathcal{L}_{\text{REL}}$, is defined as the negative log probability of ground-truth relations r^* in the training documents $\mathcal{D}$. Formally, we defined these losses as:

$$\begin{aligned} \mathcal{L}_{\text{ENT}} &= -\sum_{e^* \in \mathcal{D}} \log P(e^* | \mathbf{e}_i) \\ \mathcal{L}_{\text{REL}} &= -\sum_{r^* \in \mathcal{D}} \log P(r^* | \mathbf{r}_i) \end{aligned} \tag{2.8}$$

where e^* and r^* are the gold entities and relations in the training corpus $\mathcal{D}$, and $\mathbf{e}_i$ and $\mathbf{r}_i$ are the representations of entities and relations obtained with the span encoder.

The pre-training tasks in SPOT involve predicting entities and relations at two levels - entity and relation levels, respectively. These predictions are based on the representations generated by the *span encoder* and *span pair encoder*.

SPOT has proved to be more efficient in representing entities and relationships without requiring fine-tuning while using significantly fewer parameters than previous methods.

2.4.1.3 *Textual Triplet Integration*

Previous methods have focused on identifying and aligning entities or relations from KB in textual resources. However, the effectiveness of these methods depends on the ability to align these elements with the corresponding text. We now explore methods that translate the complete *triplet* element into text using text templates based on entity descriptors.

COMET (Bosselut et al., 2019) proposed a more general method to convert the relational *triplets* from KBs into textual sequences based on specific text templates. The approach involves generating a synthetic dataset by transforming *triplets*, (h, r, t), from a commonsense KB into text and training a GPT-based model for commonsense learning using this dataset. The model learns to generate the tail entity (object) phrase, t, of a knowledge tuple given the head entity (subject) phrase, h, and relation, r. The objective is to maximize the conditional likelihood of predicting the tail entity phrase tokens as follows:

$$\mathcal{L} = -\sum_{t=|h|+|r|}^{|h|+|r|+|t|} \log P(x_t | x_{\leqslant t}) \tag{2.9}$$

where $|h|$, $|r|$, and $|t|$ represent the number of tokens in each element's phrase, and $x_{\leqslant t}$ the contextualized representations before x_t.

Similar to previous models that rely on aligning *triplets* to text sources or creating synthetic text, KG-BERT (Yao, Mao, and Luo, 2019) proposes a different approach to knowledge graph completion tasks using *entity and relation text descriptors* from a *triplet*. KG-BERT concatenates the three elements of *triplets*, separated with special tokens, and aims to predict the *plausibility* of the given triplet and the *relation label*.

Suppose we have a *triplet* (h, r, t) with their corresponding *entity (and relation) descriptors*, $\text{text}_h = (h_1, ..., h_a)$, $\text{text}_r = (r_1, ..., r_b)$, and $\text{text}_t = (t_1, ..., t_c)$. In KG-BERT, we train the model by taking as input text the token sequences:

1. "[CLS]h_1...h_a[SEP]r_1...r_b[SEP]t_1...t_c[SEP]", for the *triplet plausibility* learning task, and
2. "[CLS]h_1...h_a[SEP]t_1...t_c[SEP]", for *predicting the relation*

In both cases, the model uses the final [CLS] token projection as an aggregate sequence representation to compute the classification scores, binary and multi-label, for the *triplet plausibility* and *relation classification*, respectively.

2.4.1.4 *Fusion Embeddings Integration*

Complementary, the E-BERT model (Poerner, Waltinger, and Schütze, 2020) investigates alternative methods of integrating entities into BERT at the input layer. E-BERT focuses on transforming and concatenating entity representations and proposes aligning entity vectors obtained from Wikipedia2Vec (Yamada et al., 2020a) with BERT's wordpiece vector space. This enables efficient injection of Wikipedia's entity knowledge into BERT.

To achieve this, the authors transform vectors from the entity vector space $\mathcal{E}_{\text{Wikipedia}}[\mathbb{L}_{\text{Ent}}]$ to vectors that resemble BERT's native wordpiece vector space $\mathcal{E}_{\text{BERT}}[\mathbb{L}_{\text{WP}}]$ using a linear mapping matrix $\mathbf{W} \in \mathbb{R}^{d_{\text{BERT}} \times d_{\text{Wikipedia}}}$. Once the matrix $\mathbf{W}$ is learned, it can project entities and tokens into the BERT space, creating a function as:

$$\mathcal{E}_{\text{E-BERT}} : \mathbb{L}_{\text{Ent}} \to \mathbb{R}^{d_{\text{BERT}}}$$
$$\mathcal{E}_{\text{E-BERT}}(a) = \mathbf{W}\mathcal{E}_{\text{Wikipedia}}(a)$$

for a given entity a, where d_{BERT} is the embedding dimension from BERT and $d_{\text{Wikipedia}}$ the embedding dimension from Wikipedia2Vec.

The authors integrated this information to the model by simply concatenating or replacing the aligned tokens with the default aligned input BERT tokens. This approach enables the model to effectively combine information from both sources with a light and efficient implementation.

2.4.2 Explicit KEPLMs

We categorize the models with explicit incorporation of knowledge into three groups: adding knowledge to input layer, adding a new module to fusion knowledge, or rely on retrieval-based approaches.

2.4.2.1 *Adding Knowledge to the Input Layer*

Another research line proposes explicitly using external knowledge to improve the performance of PLMs. This is achieved by integrating relevant knowledge from a KB at the input layer. The overall model architecture of these approaches is illustrated in Figure 2.3 considering the module (1).

Consider the sentence "Tim Cook is visiting Beijing now" and associate it with a KB. Once the entities are recognized and relevant triplets are extracted from the KG, we may obtain phrases such as "Tim Cook is the CEO of Apple.", "Beijing is a city in China." or "Beijing is the capital of China." If these phrases are taken without any

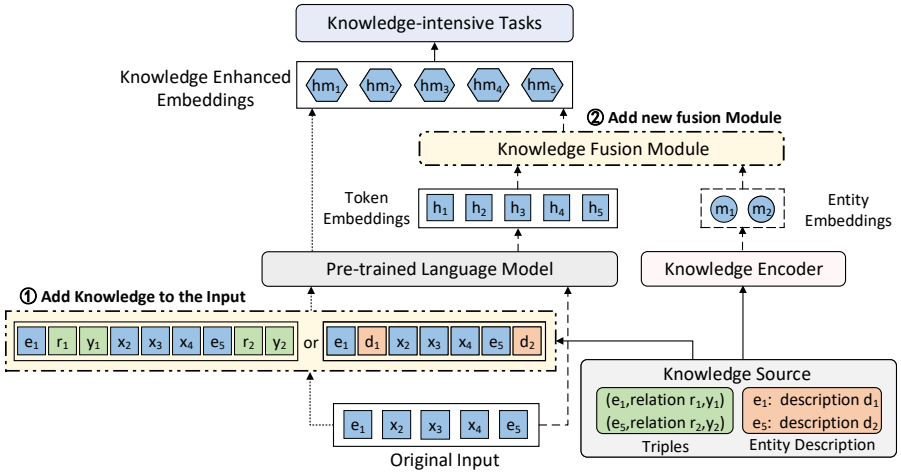


Figure 2.3: Explicit incorporation of knowledge into PLMs via modifying the model input or adding knowledge fusion modules. Source (Yang et al., 2021).

restrictions along with the original input sentence, then the original meaning of the enriched sentence may be altered.

A naive method to inject all these factual information into a model may accumulate too much knowledge, causing sentences to stray from their intended meaning. This issue is referred to as *knowledge noise* (KN) (Liu et al., 2020).

To tackle this problem, K-BERT (Liu et al., 2020) uses a matrix to limit interactions between textual knowledge and *triplet* information, thereby preventing any changes to the meaning of the original sentence. Additionally, K-BERT incorporates factual *triplets* by building a sentence tree that captures the encoded structure of a KB more effectively.

K-BERT identifies an entity in the input sentence and extends the relation and tail entity by querying a relevant *triplet* in a KB. This process converts the original sentence into a *sentence tree* (as illustrated in Figure 2.4), which is then fed into a PLM at the embeddings layer, along with a visible matrix. The visible matrix limits the impact of KN by controlling the interaction between entities identified in the original input and tokens in the original sentence through the self-attention module, acting as a masking layer.

The *triplet* information from the matrix is inserted through the input embeddings in addition to the token and position embeddings.

K-BERT uses *triplets* and sequences to analyze knowledge graphs, but it only looks at isolated *triplet* information from the graph and ignores other relations. In contrast, CoLAKE (Sun et al., 2020a) pro-

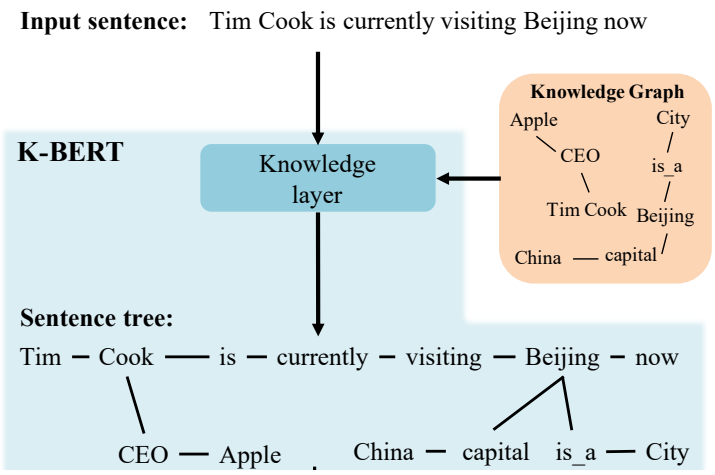


Figure 2.4: Illustration of the *sentence tree* proposed by (Liu et al., 2020) extending the original input with relevant KB triplets.

poses a *word-knowledge graph* (WK graph) that combines language and knowledge context for a more enriched representation.

CoLAKE first tokenizes the input sentence into a sequence of tokens and fully connects them as a word graph, which differs from the single-branched *sentence tree* to construct the *WK graph*. It then identifies and links entities in the sentence with a KB using an entity linker. The entity nodes in the initial graph are replaced by the linked entities from the KB, and the extended knowledge sub-graph is appended to the word graph. This results in a fully connected word graph with tokens aligned with the extracted knowledge sub-graphs from large KBs.

This knowledge is given into the model by concatenating the input sequence with the aligned node tokens from the KB and adapting the position embeddings to indicate the relation between entities, similar to the *sentence tree*. In addition, type embeddings are also considered at the input layer for the three types: word, entity, and relation. For the pre-training tasks, CoLAKE, similar to ERNIE 1.0, adapts the classic MLM task at the entity and relation level. This approach offers a nuanced solution to balance information injection while preserving the integrity of the sentence's intended meaning.

2.4.2.2 *Adding Knowledge Fusion Modules*

There are many models that propose dedicated modules to merge different modal spaces, with a focus on encoding text and knowledge sources. Given the different vector spaces from the source, these modules aim to perform an inter-modal fusion. The architecture of many

of these models follows the one in Figure 2.3 considering the "(2) Add new fusion Module" component and the Knowledge Encoder module.

FUSION ON TOP OF THE PLM.

One research line introduces a dual encoder architecture for Text and Knowledge called the *T-K module* (the PLM and Knowledge Encoder, respectively, in our generalized diagram from Figure 2.3). The first model to introduce this architecture was ERNIE (Zhang et al., 2019), which incorporated it on top of the PLM. In this structure, a `T-Encoder` is followed by a `K-Encoder`. The T-Encoder encodes the text corpus as a typical PLM, while the K-Encoder integrates entity embeddings from the knowledge space into the entity embeddings in the text space.

The `K-Encoder` consists of stacked `Aggregators` designed for encoding both tokens and entities, using two multi-head self-attentions (`MH-ATTs`) (Vaswani et al., 2017), and use as input the contextualized representations obtained from the `T-Encoder` and the entity embeddings. Subsequently, an additional *information fusion module* as a top aggregator is used to merge their heterogeneous features. The final output embeddings of the `K-Encoder` are determined by the output embeddings of both tokens and entities computed by the top aggregator.

Formally, the `K-Encoder` takes as input the token contextualized representations from a previous layer and KB embedding representation for entities aligned to the text, represented as $\{y_1^{(i-1)}, \ldots, y_n^{(i-1)}\}$ and $\{e_1^{(i-1)}, \ldots, e_m^{(i-1)}\}$, respectively:

$$\begin{aligned}\{y_1^{(i)}, \ldots, y_n^{(i)}\}, \{e_1^{(i)}, \ldots, e_m^{(i)}\} = \texttt{Aggregator}(\\ \{y_1^{(i-1)}, \ldots, y_n^{(i-1)}\}, \{e_1^{(i-1)}, \ldots, e_m^{(i-1)}\})\end{aligned} \tag{2.10}$$

The top integration of the token and entity sequence in the i-th aggregator is achieved through the *information fusion layer* defined as follows:

$$\begin{aligned} h_j &= \sigma(\tilde{W}_t^{(i)} \tilde{w}_j^{(i)} + \tilde{W}_e^{(i)} \tilde{e}_k^{(i)} + \tilde{b}^{(i)}), \\ w_j^{(i)} &= \sigma(W_t^{(i)} h_j + b_t^{(i)}), \\ e_k^{(i)} &= \sigma(W_e^{(i)} h_j + b_e^{(i)}) \end{aligned} \tag{2.11}$$

where h_j is the inner hidden state integrating the information of both the token and the entity intermediate representations ($\tilde{w}_j^{(i)}$ and

$\tilde{e}^{(i)}$ respectively). $\sigma(\cdot)$ is the non-linear activation function, usually the GELU function (Hendrycks and Gimpel, 2016). For tokens that lack aligned entities, $\tilde{e}_k^{(i)}$ and $e_k^{(i)}$ are set to 0.

ERNIE is built on BERT and employs TransE as the KG embedding. It is pre-trained using the dEA task introduced in Section 2.4.1.1.

It is worth noting that while TransE is a useful method for embedding KGs, it has some limitations, such as not being able to fully capture the structure and complex relations found in KGs, including 1-to-n, n-to-1, and n-to-n relations. To address these limitations, various approaches have been proposed after ERNIE.

Inspired by the ERNIE architecture, the BERT-MK model (He et al., 2020) proposes some improvements. BERT-MK uses a dual-encoder architecture (*T-K module*) similar to ERNIE. However, during pre-training, it enhances TransE embeddings by learning KG embeddings from scratch and incorporating additional information about neighboring entities in the knowledge encoder component.

To learn these KG embeddings, BERT-MK creates a subgraph by sampling triplets from the graph at 1-hop and 2-hop distance. We consider a x-hop node as an entity that is at a x-edges distance in a KG. BERT-MK uses this subgraph to create Graph-Contextualized Knowledge Embeddings (GCKE) based on a Transformer architecture, using a translation-based score as the loss function.

Once the enriched entity representations are obtained using the GCKE method, the embeddings are combined with the PLM using the same fusion module proposed by ERNIE. BERT-MK is specifically designed for the biomedical domain and was trained on the PubMed corpus and the UMLS meta thesaurus.

One of the main issues with the BERT-MK model is the presence of extra redundancy and noise in the subgraph creation process, presented before as the KN problem. Not all knowledge is helpful, and irrelevant or ambiguous knowledge can be injected into the model regardless of the textual context, leading to errors. To tackle this problem, the CokeBERT model (Su et al., 2021) has been proposed.

CokeBERT improves the K-Encoder of the original ERNIE by introducing a Dynamic Knowledge Context Encoder (DK-Encoder). This encoder uses a semantic-driven graph neural network (S-GNN) based on an attention mechanism to select relevant information based on the textual context. It can reach up to 2-hop of neighbor entities and compute an entity embedding based on the information aggregated on the subgraph. Finally, it fuses this entity embedding with the PLM representation with the typical information fusion module.

To filter out irrelevant subgraphs, the model considers the proximity of the meaning of the neighbor node to the text. In other words,

the closer the meaning of the neighbor node is to the text, the more its information is incorporated into the entity embedding.

FUSION ON THE MIDDLE OF THE PLM.

Other works involve retaining the output loss layers of a PLM while fine-tuning on unlabeled corpora to train the complementary modules. This approach helps recycle the high-level abstractions learned from the original model's top layers, considering the interactions between Transformer layers, which results in better knowledge transfer.

One of the first approaches in this direction is the KnowBERT model (Peters et al., 2019), which uses the Knowledge Attention and Recontextualized component (KAR) to connect Transformer Encoder blocks and a KB.

KnowBERT assimilates entity-related knowledge in sentences by leveraging *entity spans* within the input text. Unlike ERNIE-based models, KnowBERT dynamically detects entities from the input text through an Entity Linker instead of requiring entity alignment for pre-training.

The KAR component operates by projecting contextualized PLM representations into entity embeddings spaced according to *span mentions*. It introduces a *mention-span* self-attention layer between entities within the same sentence, enabling the computation of contextualized entity embeddings. An integrated entity linker then calculates the weighted average entity embeddings, enriching the *span representations* with knowledge from the KB. The enhanced entity representations are then projected back into the PLM vector space.

Formally, given the contextual representations, $\mathbf{H}_i$, at a specific layer i, and a set of *mention-span* representations for each entity candidate, denoted as $\mathcal{C}$ based on the KB, the resulting knowledge-enhanced representations are expressed as $\mathbf{H}'_i = \text{KAR}(\mathbf{H}_i, \mathcal{C})$.

These enhanced representations undergo recontextualization and are projected back into the subsequent PLM layer, represented as:

$$\mathbf{H}_{i+1} = \text{TransformerBlock}(\mathbf{H}'_i) \tag{2.12}$$

After this step, the PLM procedure goes on. More specifically, the KAR component initiates the process by projecting the contextual representations, $\mathbf{H}_i$, into the entity dimension (E) through a linear projection:

$$\begin{aligned}
\mathbf{H}_i^{\text{proj}} &= \mathbf{H}_i \mathbf{W}_1^{\text{proj}} + \mathbf{b}_1^{\text{proj}} \\
\mathbf{H}_i^{'\text{proj}} &= \text{MLP}(\text{MultiHeadAttn}(\mathbf{H}_i^{\text{proj}}, \mathbf{S}^{'e}, \mathbf{S}^{'e})) \\
\mathbf{H}'_i &= \mathbf{H}_i^{'\text{proj}} \mathbf{W}_2^{\text{proj}} + \mathbf{b}_2^{\text{proj}} + \mathbf{H}_i
\end{aligned} \tag{2.13}$$

In addition to the BERT pre-training tasks, Entity Linking loss is added to the model. The KAR module from KnowBERT also includes additional KBs inserted at different locations between layers.

Nevertheless, as mentioned above, these approaches encounter issues when it comes to generalizing well and producing accurate representations of *unseen entities during the training* phase. These issues extend not only to the entity level but also to handle relations, higher structures, and interactions between the nodes in the KB.

To address these problems, JAKET (Yu et al., 2022) proposes a dual module for knowledge and language to assist one another. JAKET works directly with entity text descriptors, which facilitates the adaptation to unseen knowledge graphs in the fine-tuning phase. This idea is similar to that proposed by KG-BERT (presented in Section 2.4.2.1).

Mainly, JAKET divides the PLM into the first six layers and the last six. The initial half processes the input text, and the hidden layer representation is obtained. Similarly, the entity representations are obtained from the knowledge module, which uses a graph neural network (GNN). Subsequently, the outputs of text and entity representations are combined. At each entity position in the text, the corresponding entity embedding representation is added and then input to the last six layers of the model for subsequent training. The knowledge and text representation spaces reinforce each other cyclically to learn improved representations.

Overall, the fusion modules approach has been found to enhance performance for various KG and NLP tasks compared to traditional PLMs. However, there are significant limitations associated with this approach. The use of additional modules in the architecture significantly increases the size of the final model in terms of parameters, which in turn increases the cost of pre-training new and existing modules (Roberts, Raffel, and Shazeer, 2020). Moreover, adopting new pre-training tasks carries the risk of *catastrophic forgetting* (Kirkpatrick et al., 2017), leading to the PLM losing previously acquired knowledge during its first pre-training phase. This phenomenon has been observed in different ranking (Lovón-Melgarejo et al., 2021) and generative applications (Vu et al., 2022).

While these limitations are still an open problem in the literature, recent approaches propose empirical solutions. These techniques suggest that the existing parameters of language models remain frozen during knowledge training while only specific parameters from complementary modules can be trained. K-adapters are one of the first methods that enable this approach to learn different module parameters for different tasks in parallel independently. In their original work, K-adapters fuse linguistic and factual knowledge via adapter

modules, which can limit the number of parameters needed to train by introducing trainable multi-layer projections in the middle of the transformer layer. Recent studies have demonstrated that utilizing adapter modules achieves superior performance and requires fewer resources to enhance existing PLMs (Houlsby et al., 2019; Pfeiffer et al., 2021; Hu et al., 2021; Pfeiffer et al., 2020).

2.4.2.3 *Retrieval-based approaches*

The presented model integrates external knowledge with textual representations within its parameters. However, real-world knowledge is dynamic, and these approaches have a limitation: they *do not facilitate updates to incorporated knowledge* without retraining the model. In contrast, retrieval-based approaches focus on leveraging knowledge as an external reference, primarily for Question Answering (QA) tasks. These approaches require the model to understand both textual semantic meanings and the latest real-world knowledge, emphasizing the retrieval, selection, and encoding of the most pertinent information from knowledge repositories.

One such model is the Retriever-Augmented Language Model, referred to as REALM, introduced by (Guu et al., 2020). REALM is a "*retrieve-then-predict*" model that retrieves and attends to documents from a large corpus during pre-training and subsequently predicts the correct answer when confronted with an open-domain question. During pre-training, the model learns a distribution $p(y|x)$, where x represents a sentence from a pre-training corpus with some tokens masked, and the model predicts the values of these missing tokens, y. In the fine-tuning stage, which is oriented towards Open-QA, x is a question, and y is the answer.

REALM comprises two modules: the *knowledge retriever* and the *knowledge-augmented encoder*. The *knowledge retriever* computes $p(z|x)$, where z denotes the relevant documents retrieved from a given corpus. The *knowledge-augmented encoder* calculates $p(y|z,x)$, predicting the masked tokens based on input x and the highly relevant retrieved documents z. In this paradigm, updates to the model are only required if the knowledge corpus itself requires updating due to changes in real-world knowledge.

Another research line has leveraged external KBs to enhance the *interpretability* and performance of PLMs. An instance of this research line is Entity as Experts (EaE) by Févry et al. (2020). EaE aims to refine entity representations directly from textual input and model parameters to partition the parameter space effectively based on entity identity. This enables the model to selectively access different mentions of an entity in textual input while maintaining consistent parameters.

For example, when dealing with the entity "Michael Jackson" EaE proposes to use the same set of parameters for references such as "The King of Pop" or "MJ".

To achieve this, EaE incorporates an Entity Memory Layer within the Transformer architecture that intricately maps entity mentions to specific parameters. The FaE model, introduced by Verga et al. (2021), takes a similar approach by storing entities and factual knowledge in an external memory component that encodes *triplets* sourced from a symbolic KB. The external memory in FaE, similar to EaE, forms a key-value memory structure composed of learned entity embeddings and plays a crucial role in retrieving entities based on their KB properties.

In conclusion, researchers have investigated various methods to incorporate external knowledge sources to enhance the representation of PLMs with factual and updated knowledge, which is not explicitly captured by the classical pre-training tasks. These injection methods are explored at different levels of the PLM architecture. Although these enhanced models have shown significant improvements in their related tasks compared to default PLMs, further exploration is needed to improve the methodology of encoding structural elements of these sources with a more cost-effective approach for pre-training. Table 2.1 lists all the presented models, including technical details omitted in the text, such as the backbone models and dataset used and main evaluation tasks.

2.5 EVALUATION APPROACHES

To determine the effectiveness of improving architecture or including external knowledge in PLMs or KEPLMs, researchers use two main evaluation approaches: *intrinsic* and *extrinsic* evaluations. From an *interpretable* standpoint, *intrinsic* evaluations are more appropriate when evaluating PLMs because they provide a more understandable evaluation approach. *Extrinsic* evaluations, on the other hand, depend on performance in related KG or NLP tasks. We will discuss these two evaluation types in Section 3.2.3 and provide a detailed comparison between them.

This section introduces *intrinsic* evaluation methods, which are organized into two groups: *knowledge capacity* and *task diagnostic*.

Model	Backbone Model	Source	Tasks	Method
AMS(2019)	BERT	ConceptNet, Wikipedia	QA, RC	Implicit
COMET(2019)	GPT	ConceptNet, OMCS	QA, RC	Implicit
ERNIE 1.0(2019)	BERT	Ch. Wikipedia, Baidu	SS, NER	Implicit
KG-BERT(2019)	BERT	Freebase, WordNet	LP, RC	Implicit
ERNIE(2019)	BERT	Wikipedia, Wikidata	ET, RC	Explicit
KnowBERT(2019)	BERT	Wikipedia, WordNet	ET, RC	Explicit
SentiLARE(2020)	RoBERTa	SentiWordNet	SC	Implicit
SenseBERT(2020)	BERT	WordNet Supersense	WSD	Implicit
LUKE(2020)	RoBERTa	Wikipedia	ET, RC	Implicit
E-BERT(2020)	BERT	Wikipedia	FR, EL	Implicit
K-BERT(2020)	BERT	Chinese Wikipedia	SC, QA	Explicit
CoLAKE(2020)	RoBERTa	Wikipedia	ET, RC	Explicit
BERT-MK(2020)	BERT	UMLS	ET, RC	Explicit
REALM(2020)	T5	Wikipedia, CC-news	IR	Explicit
KEPLER(2021)	RoBERTa	Wikidata5M, WordNet	ET, RC	Implicit
ERICA(2021)	BERT, RoBERTa	Wikipedia, Wikidata	ET, RC	Implicit
CokeBERT(2021)	BERT	Wikipedia, Wikidata	ET, RC	Explicit
SPOT(2022)	RoBERTa	Wikipedia, Wikidata	ET, RC	Implicit
JAKET(2022)	RoBERTa	Wikipedia, SLING	ET, RC	Explicit

Table 2.1: List of the KEPLMs presented in this chapter, including the backbone model used to train them, the knowledge source, the central evaluation task to which they were applied, and the integration method. We considered the tasks: question-answering (QA), reading comprehension (RC), semantic similarity (SS), named entity recognition (NER), entity typing (ET), relation classification (RC), link prediction (LP) stands for question-answering, RC for reading comprehension, sentiment classification (SC), word-sense disambiguation (WSD), fact recalling (FR), entity linking (EL), information retrieval (IR)

2.5.1 Measuring Knowledge Capacity

In the *knowledge capacity* method, the underlying assumption is that KEPLMs with more knowledge would perform better in downstream NLP tasks. Therefore, evaluations are conducted in terms of *fact-recalling* and *probing* approaches across various domains.

2.5.1.1 *Fact Recalling*

PLMs are evaluated based on their ability to recall facts without prior training (zero-shot setup). Various knowledge probes have emerged to measure the knowledge capacity of PLMs and KEPLMs. These probes use *cloze-style* prompts, where a (subject, relation, object) triplet is taken, and a phrase is created with the object masked. The model is then evaluated based on its ability to predict the masked object. For example, the model is given the phrase "Dante was born in [MASK]" and is expected to predict "Florence" as the masked token.

THE LAMA BENCHMARK.

The first benchmark in this domain is Language Model Analysis (LAMA) (Petroni et al., 2019), which evaluates the PLMs' performance. LAMA relies on completing statements derived from facts in knowledge bases, drawing from varied sources such as Google-RE[4], T-Rex (Elsahar et al., 2018), ConceptNet (Speer, Chin, and Havasi, 2017), and SQuAD (Rajpurkar et al., 2016).

The Google-RE corpus comprises five types of relational triplets, including "place of birth", "date of birth", and "place of death", selected by LAMA. These triplets are transformed into *cloze-style* sentences, called "prompts", using meticulously constructed templates. For instance, the "place of birth" triplet is articulated as "[S] was born in [O]", where S signifies the head entity and O is the tail entity. T-Rex, a Wikidata subset with 41 relations of varying cardinality types (1-1, 1-N, N-M), undergoes a similar manual transformation into *cloze-style* prompts. ConceptNet contributes 16 relations, and fill-in-the-blank sentences are manually constructed for the SQuAD dataset, comprising question-answer pairs.

4 https://code.google.com/archive/p/relation-extraction-corpus/

LAMA BENCHMARK VARIATIONS.

The LAMA dataset was found to have some *triplets* that were too "easy to guess" as the context provided hints on the searched token. This means that the model could predict that a person speaks Italian if the person has an Italian-sounding name, depending on the surface form of the entity name. A subset of LAMA called LAMA UnHelpfulNames (LAMA-UHN) (Poerner, Waltinger, and Schütze, 2020) was proposed to address this issue. LAMA-UHN removed the *triplets* with overly helpful entity names.

Although LAMA-UHN uses the original Google-RE and T-REx KB from LAMA, it presents a more challenging benchmark than LAMA. To filter LAMA, LAMA-UHN used two heuristic methods to filter out the "easy to guess" triplets:

1. *String match filter*: triplets where the correct answer (e.g., Apple) is a case-insensitive substring of the subject (e.g., Apple Watch).

2. *Person name filter*: it whitespace-tokenizes the subject name(e.g., Jean Marais into {Jean, Marais}), and using BERT, determine if a single token is available to give the correct answer (e.g., "Jean is a common [MASK] name", "Marais is a common [MASK] name", with [MASK]=french).

The accuracy of PLMs in answering questions based on KG *triplets* is limited by the quality of the prompts used to convert the triplets into understandable questions. The LAMA and LAMA-UHN benchmarks provide a lower bound estimate of the knowledge captured by PLMs due to their dependence on handcrafted templates for generating prompts (Jiang et al., 2020). For example, a model may have difficulty answering, "Obama is a by [MASK] profession", but can provide accurate results when asked, "Obama worked as a [MASK]".

To address this issue, researchers have proposed a new benchmark called the LM Prompt and Query Archive (LPAQA) (Jiang et al., 2020). LPAQA builds upon existing benchmarks, such as Google-RE and T-REx from LAMA, by using a mining methodology to identify phrases that encapsulate relations between entities. These phrases are generated by leveraging intermediary words between entities within a designated corpus. The paragraphing process generates multiple prompts that maintain the original phrase's semantic meaning. This process involves translating the original *triplet* phrase into another language and back.

The use of the LPAQA benchmark showed an improvement in the accuracy of factual knowledge retrieval. This improvement outper-

forms manually designed prompts by a large margin. Similarly, AutoPrompt (Shin et al., 2020) and OptiPrompt (Zhong, Friedman, and Chen, 2021) are two recently proposed techniques aimed at improving the recall performance of PLMs by automating prompt creation.

APPLICATIONS.

An application of fact recalling is proposed in the work by Jain and Anke (2022). The authors explored various prompts to extract hypernymy knowledge from PLMs in a zero-shot setup. This was done to reconstruct taxonomies from different depth levels. Based on the typical MLM task to predict the [MASK] token in the prompt, the authors propose to use the LMScorer approach from Salazar et al. (2020). The LMScorer approach scores a sentence based on its factuality, which is defined as:

$$\texttt{LMScorer}_{\mathcal{M}}(\mathbf{X}) = \exp\left(\sum_{i=1}^{|\mathbf{X}|} \log P_{\mathcal{M}}(x_i|\mathbf{X}_{\setminus i})\right) \tag{2.14}$$

where $\mathbf{X}$ is a given sentence, and $\mathcal{M}$ the PLM. The idea is to replace each token w_i in the sentence with `[MASK]` and then predict it using the past and future tokens. By masking each token iteratively and using the LM's prediction, we can compute a pseudo-likelihood score for each token x_i. This approach is efficient and provides competitive results when compared to other computationally expensive and heuristic techniques.

Considering a few-shot approach for factual recalling, the work from He, Cho, and Glass (2021) proposes a setup that involves fine-tuning a model using a small number of examples, typically 10 to 20. The authors used the T-REx dataset from LAMA and created a new dataset, T-REx-2p, that contains 2-hop relations in addition to the 1-hop triplets present in the original T-REx. The templates were hand-crafted, similar to LAMA. Interestingly, the authors observed that fine-tuning a model with just ten examples produced competitive results when compared to more complex approaches like OptiPrompt. This highlights the potential of few-shot probing setups as a simple yet effective way to improve the performance of language models in recalling factual knowledge. With the increasing availability of large-scale pre-trained models, this approach could be particularly useful in scenarios where a significant amount of fine-tuning data is unavailable.

2.5.1.2 *Probing*

Although fact-recalling is a useful evaluation metric for assessing the knowledge stored in a PLM, it may not always be sufficient to understand which features are utilized to enhance the performance of a particular task. This is because relying solely on a top MLM layer and reducing the problem to an input phrase does not provide a complete picture. Therefore, more comprehensive evaluations are necessary.

One approach to answering this question is through the use of *probes*, which are supervised models trained to predict a specific property, such as parts-of-speech, from a limited view of the representation (Hewitt and Liang, 2019). By training these *probes*, researchers can gain insights into the information encoded in a model's representations.

However, to assess the performance of a model on a probe can be challenging, as it is unclear whether the model's performance is due to the encoded information or the complexity of the *probe* itself (Maudslay et al., 2020). To overcome this difficulty, researchers have proposed the use of *diagnostic classifiers*. These are classifiers that are trained on a simplified version of the dataset to understand the task, and then they are tested on a more complex version of the dataset. This approach is also referred to as "poverty of the stimulus" (Lin, Tan, and Frank, 2019).

There are several types of probes available for assessing different language characteristics and tasks. A first study conducted by Lin, Tan, and Frank (2019), named Open Sesame, examined the organization of sentence representations produced by BERT-based models. They used diagnostic classifiers to investigate the presence of both hierarchical and linear representations of words. Similarly, Hewitt and Manning (2019) proposed structural *probes* to evaluate whether syntax trees are encoded in a model's word representation space, learning a linear projection.

Moreover, we can find numerous *probes* that have been developed for evaluating various linguistic aspects (Manning et al., 2020; Tenney, Das, and Pavlick, 2019; Voita and Titov, 2020), including those related to semantics. In the context of this dissertation, we are specifically interested in probes that examine the semantic-level information present in language models. In this regard, we now introduce the main probes that exist in literature for *reasoning-based* and *lexical semantic* tasks.

PROBES ON REASONING-BASED TASKS.

Various probes have been introduced at the semantic level to assess the reasoning capabilities of PLMs. One benchmark, oLMpics (Talmor et al., 2020a), has been designed to assess the symbolic reasoning skills of PLMs.

In this benchmark, the authors suggest that a model's ability to perform a given task should not require extensive training. If a model fails to perform well, it may be due to a difference between the language it was trained on and the language used in the task. A good model should be able to adapt effectively even with limited examples. The oLMpics probes cover diverse aspects of reasoning, including number comparison, frequency of events, negation, the conjunction of facts, and multi-hop reasoning.

Figure 2.5 shows some samples from the benchmark. The probes are structured into two setups based on multiChoice MLM and question-answering. Following the approach of the LAMA (Petroni et al., 2019) benchmark, the question is transformed into a *cloze-style*, with the [MASK] token containing the answer and five options are given to the model to choose. For example, an input like "[CLS] Cast [MASK] drinks coffee [SEP]" is processed through the model. The contextualized representation from [MASK] is fed through the MC-MLM layer to predict the correct output from the token options: *always*, *sometimes*, and *never* (with "never" being the gold answer for this example). The MC-QA setup adds an MC-QA layer atop the model after projecting the final representation of the [CLS] token. The model is presented with an input concatenating the question and answer candidate, such as "[CLS] question [SEP] answer[SEP]."

The oLMpics benchmark is designed to gauge the symbolic reasoning skills of PLMs without any prior assumptions about their competence in a particular task. In a recent study (Richardson and Sabharwal, 2020), the authors aimed to investigate whether specialized PLMs, trained on extensive collections for the QA task, can handle definition and taxonomic reasoning. To this end, they proposed a probe for the multi-hop question-answering task based on expert knowledge. Their probe is formulated as a question with multiple answer choices (5) sourced from WordNet and the GNU Collaborative International Dictionary of English (GCIDE).

The authors created four individual datasets to evaluate these probes to study specific semantic relations, including hypernymy, hyponymy, synonymy, and definitions, along with an additional probe oriented towards the word sense disambiguation task (Figure 2.6). The authors generated *four distractor* answers to sample the other

Probe name	Setup	Example	Human
ALWAYS-NEVER	MC-MLM	*A chicken [MASK] has horns.* **A.** never **B.** rarely **C.** sometimes **D.** often **E.** always	91%
AGE COMPARISON	MC-MLM	*A 21 year old person is [MASK] than me in age, If I am a 35 year old person.* **A.** younger **B.** older	100%
OBJECTS COMPARISON	MC-MLM	*The size of a airplane is [MASK] than the size of a house .* **A.** larger **B.** smaller	100%
ANTONYM NEGATION	MC-MLM	*It was [MASK] hot, it was really cold .* **A.** not **B.** really	90%
PROPERTY CONJUNCTION	MC-QA	*What is usually located at hand and used for writing?* **A.** pen **B.** spoon **C.** computer	92%
TAXONOMY CONJUNCTION	MC-MLM	*A ferry and a floatplane are both a type of [MASK].* **A.** vehicle **B.** airplane **C.** boat	85%
ENCYC. COMPOSITION	MC-QA	*When did the band where Junior Cony played first form?* **A.** 1978 **B.** 1977 **C.** 1980	85%
MULTI-HOP COMPOSITION	MC-MLM	*When comparing a 23, a 38 and a 31 year old, the [MASK] is oldest* **A.** second **B.** first **C.** third	100%

Figure 2.5: Samples of the oLMpics benchmark. Source: Talmor et al. (2020a).

answer options for a given probe, to simulate negative examples. To make the *distractors* challenging for the model, they were sampled based on edge distances from the golden nodes, taking advantage of the tree-based structure of the sources. Some of the models used an *inoculation* setup (Section 2.3), wherein models are trained on new challenge tasks (separately for each probe) using only a small number of examples to test model's adaptation to these tasks. The work by Richardson and Sabharwal (2020) has revealed that PLMs struggle to achieve human-level reasoning.

In the same research line, recent research aims to address this issue by determining whether PLMs, without fine-tuning on specific tasks, respect the transitivity constraint of the IS-A relation, which is crucial for logical consistency.

The IS-A relation establishes that for senses A, B, and C, if "A is a B" and "B is a C", then "A is a C". This study aims to quantify the extent to which BERT agrees with the transitive property of IS-A relations, where the senses are drawn from the WordNet dataset.

What did these evaluations show? The oLMpics study initially assessed the BERT and RoBERTa language models and found that their struggle in solving reasoning tasks is closely linked to specific values and language context, which limits their ability to generalize to arbitrary situations. The QA study evaluated the performance of BERT in predicting the is-a relationship and found that it achieves an accuracy of 72.6%, indicating a reasonable understanding of this lexical relationship. However, BERT's predictions are not always logically consistent. Specifically, when predicting A is a B and B is a C, it only predicts A is a C 82.4% of the time, implying the need for further research to improve the logical consistency of PLMs and BERT, thereby approaching human-level reasoning.

PROBES ON LEXICAL SEMANTIC TASKS.

The work from Vulić et al. (2020) introduced a probe to investigate lexical semantic knowledge in multilingual models. They conducted an empirical study with static type-level "BERT-based" word embed-

Example Questions and Answers (q, a)
q. In the sentence The baby nestled her head, the word nestled is best defined as: **a**. position comfortably
q. In The thief eluded the police, the word or concept eluded is best described as a type of **a**. escape event defined as to run away from..
q. Given the context they awaited her arrival, which of the following word or concept is a specific type of arrival? **a**. crash landing, defined as an emergency landing under circumstances where....'
q. Which set of words best corresponds to the definition a grammatical category in inflected languages governing agreement ....? **a**. gender,...

Figure 2.6: Samples of the probes from Richardson and Sabharwal (2020) for multi-hop question answering.

dings, which are similar to traditional static word embeddings. Their analysis included five lexical tasks across six languages: lexical semantic similarity, word analogy resolution, bilingual lexicon induction, cross-lingual information retrieval, and lexical relation prediction.

Overall, the study emphasized the need for distinct configurations to extract static representations for optimal performance across various languages and tasks. The research also revealed that lexical knowledge predominantly resides in lower Transformer layers. Therefore, excluding higher layers during the extraction process leads to superior performance.

Interestingly, while static word representations from monolingual language models may compete with or surpass traditional FastText embeddings (Bojanowski et al., 2017) in specific tasks, representations from massive multilingual models like mBERT demonstrate substantially lower performance in lexical tasks. This highlights the subtle

challenges associated with leveraging massively multilingual models for lexical semantics tasks.

2.5.1.3 *Improving Linear Probes.*

The existing probes have primarily been classified as *linear probes*, as they commonly employ a simple (typically linear) classifier to predict linguistic properties of interest, utilizing *probe* accuracy for its interpretation (Manning et al., 2020). Recent advancements have proposed refined probes to capture the knowledge embedded in these language models effectively.

In the work by Chen et al. (2020a), it was observed that many existing *probes* implicitly assume that the embedding representations from PLMs lie in specific metric spaces, typically the Euclidean space. The authors explored alternative subspace representations, precisely the hyperbolic space, which is a distinctive Riemannian space characterized by constant negative curvature. This space proves valuable for representing tree-like distributions more effectively (Nickel and Kiela, 2017). The authors adopted a generalized *Poincaré Ball*, a specialized model of hyperbolic spaces, to formulate a *Poincaré probe* for contextualized embeddings. They projected the embedding representations into this hyperbolic space to conduct *probes* for tasks related to dependency trees and sentiment analysis.

The rationale behind this choice stems from the intrinsic nature of the *Poincaré ball*, where the inner volume grows exponentially with its radius. The authors argued that this exponential growth provides a more accommodating space for embedding knowledge than Euclidean-based projections. In contrast, the Euclidean space grows polynomially, offering less capacity for embedding knowledge. A more in-depth exploration of the hyperbolic space and these probes is provided in Section 3.2.2.

Particularly, comparing KEPLMs introduces several challenges. In a recent investigation by Hou, Fu, and Sachan (2022), linear classifiers tend to exhibit high variance, especially in extensive knowledge graphs. This high variance complicates the reliability of direct comparisons between two linear classifiers. Additionally, the same study highlights the drawbacks of prompt-based evaluations, emphasizing the labor-intensive and time-consuming nature of manually constructing the mentioned prompt-based templates for assessment.

To address these challenges, Hou, Fu, and Sachan (2022) presents an approach based on *information theory* to measure the difference in factual knowledge encoded between a language model and an enhanced language model incorporating a knowledge graph. They suggest evaluating the encoded factual knowledge in terms of Mutual

Information (MI) between the representations of the model and the knowledge graph. This proposed method introduces a more dependable and efficient way of comparing diverse KEPLMs.

The MI proposed in this study is a metric that quantifies the information gained about a random variable, x, from a knowledge graph by observing its representation in a PLM, denoted as $MI(x; g)$. In order to assess different encoded knowledge about an entity 'g' between its representation in a *base PLM* (x) and an *enhanced PLM* (h), we measure the difference between their respective MI values ($MI(x; g)$ and $MI(h; g)$). The authors suggest using a Graph Convolution Simulator (GCS) to approximate MI values, which enables a more efficient and reliable way to compare different KEPLMs and compute knowledge changes between them. Under this approach, one can compare two different models (x and h in this case) for the following scenarios:

1. MI(h;x) $\simeq$ MI(x;x), indicates that the *enhanced representation* of x is similar to the *base PLM* representation of x, meaning that the *enhanced model* does not forget the original representation.

2. MI(h;g) $\simeq$ MI(g;g), suggests that the *enhanced representation* of the graph is fully captured, since its knowledge representation is similar to the graph knowledge.

3. MI(h;g) $\simeq$ MI(x;g), signals that the *enhanced representation*, considering the graph, is similar to the prior knowledge of the model about the graph. This shows that the previous knowledge from the *base PLM* hinders the integration of new knowledge, which is known as *catastrophic remembering* (Kaushik et al., 2021).

4. MI(h;x) $\simeq$ MI(x;g), indicates that the *enhanced representation* of x is similar to the prior knowledge of the *base PLM* about the graph, suggesting that the old knowledge of the *enhanced model* is forgotten, which is known as *catastrophic forgetting* (Kirkpatrick et al., 2017).

It is important to note that this approach has a limitation since it only relies on entities and does not consider relation information.

2.5.2 Performing Task Diagnostic

Another line of research is dedicated to developing interpretable approaches that shed light on how the encoding elements of PLMs contribute to specific tasks. These approaches adopt a diagnostic methodology, where a set of axiomatic characteristics is defined for a given

task, and then the quality of model representations is evaluated based on these axioms.

An early example of this paradigm is the OpenSesame study by Lin, Tan, and Frank (2019). The OpenSesame framework introduces diagnostic classifiers to comprehend the linguistic knowledge encoded in BERT. The linguistic properties are evaluated by examining the embeddings of individual words, and the evaluation of linguistic knowledge is divided into four axiomatic tasks. These tasks include identifying the main auxiliary in a sentence, identifying the subject noun in a sentence, diagnosing for subject-verb agreement, and identifying anaphor-antecedent dependencies.

The findings from the OpenSesame study is that linguistic structures are encoded in self-attention layers in BERT. Additionally, higher layers of the model capture more complex knowledge. This approach provides a more comprehensive understanding of the model's representations, allowing for better interpretability and transparency.

PLMs have demonstrated impressive results in Information Retrieval (IR) tasks (Lin, Nogueira, and Yates, 2021). In an attempt to better understand how BERT comprehends the relevance between queries and documents, a diagnostic exploration was conducted by Câmara and Hauff (2020). They examined BERT's performance on retrieval heuristics using a set of nine diagnostic datasets. These datasets were designed to fulfill specific retrieval axioms, which represent the characteristics of good retrieval functions, and were grouped into four categories. As an example of these axioms, we show them regrouped at the top of Table 2.2.

This particular analysis uses the corpus and queries from the TREC 2019 Deep Learning track's Document Ranking Task[5]. Every axiom is represented by a query q and two or three documents d_1, d_2 where $q \neq d_1 \neq d_2$. After constructing the diagnostic datasets, a DistilBERT model (Sanh et al., 2019) is fine-tuned for ranking and later evaluated for accuracy. The accuracy metric is used to indicate the fraction of instances where BERT satisfies each diagnostic dataset.

The findings reveal that even though BERT performs better than traditional models in ad-hoc retrieval, it falls short of meeting most retrieval heuristics created by IR experts. This result highlights the need for more customized heuristics to analyze and comprehend the complexities of BERT's performance in IR tasks (refer to Table 2.2).

Similarly, in a recent work, Amigó et al. (2022) developed a formal framework for embeddings, composition, and similarity functions in the context of Compositional Distributional Semantics. The frame-

5 https://microsoft.github.io/TREC-2019-Deep-Learning/.

Name	Definition
Task: Information Retrieval	
Term frequency	Including 3 axioms: i) "The more occurrences of a query term a document has, the higher its retrieval score"; ii) "The increase in retrieval score of a document gets smaller as the absolute query term frequency increases"; iii) "The more discriminating query terms (i.e., those with high IDF value) a document contains, the higher its retrieval score".
Length normalization	Considering that i) "The retrieval score of a document decreases as terms not appearing in the query are added"; and ii) "A document that is duplicated does not have a lower retrieval score than the original document".
Semantic term matching	Arguing that i) "A document's retrieval score increases as it contains terms that are more semantically related to the query terms"; ii) "The document terms that are a syntactic match to the query terms contribute at least as much to the document's retrieval score as the semantically related terms"; and iii) "A document's retrieval score increases as it contains more terms that are semantically related to different query terms".
Term proximity	Defines that: "A document's retrieval score increases as the query terms appearing in it appear in closer proximity".
Task: Compositional Distributional Semantics	
Embeddings functions	Considering the properties of measurability and angular isometry.
Composition functions	Including neutral element, norm lower bound, norm monotonicity, and sensitivity to structure.
Similarity functions	With the properties of angular distance, orthogonal embedding, and equidistant embedding.

Table 2.2: Examples of diagnostics defined for Information Retrieval and Compositional Distributional Semantics tasks. These diagnostics are defined as axioms and constraints following their original work.

work is based on Shannon's Information Theory, which considers co-occurrence distributions and text structure. The authors proposed nine formal constraints based on Information Content theory to analyze these three functions. Similar to the previous task, we showed these constraints at the bottom of Table 2.2.

The study's findings indicate that adding contextual embeddings from internal layers of PLMs does not outperform static embeddings in textual semantic composition. However, PLMs outperform the defined benchmarks for standard composition and similarity functions. Additionally, they found that the cosine distance is a robust similarity estimate, and considering structures such as word order is more effective than representing texts as a bag of words, which is consistent with previous research.

2.6 KNOWLEDGE-ENHANCED LARGE LANGUAGE MODELS

Researchers have found that scaling PLMs through model and data size adjustments often leads to improved model capacity for downstream tasks. Several studies have demonstrated this by training PLMs with an increasing number of parameters, such as the 175-billion-parameter GPT-3 and the 540-billion-parameter PaLM (Chowdhery et al., 2023). These large PLMs exhibit unique behaviors and demonstrate unexpected capabilities, known as *emergent abilities* (Wei et al., 2022a). It is important to note that these larger models differ from "smaller" ones like the 330-million-parameter BERT and 1.5-billion-parameter GPT-2.

For example, GPT-3 can solve few-shot tasks through in-context learning, while GPT-2 struggles with it. As a result, the research community has coined the term "Large Language Models (LLM)" for these larger PLMs, generating growing interest in the field (Zhao et al., 2023). Notably, this work proposes three key differences to differentiate between LLMs and conventional PLMs:

- LLMs manifest surprising *emergent abilities* not seen in smaller PLMs, contributing to enhancing language model performance on complex tasks.
- The primary means of interacting with LLMs is through the *prompting interface*, such as the GPT-4 API.
- The development of LLMs blurs the line between research and engineering, requiring extensive practical experience in large-scale data processing and distributed parallel training.

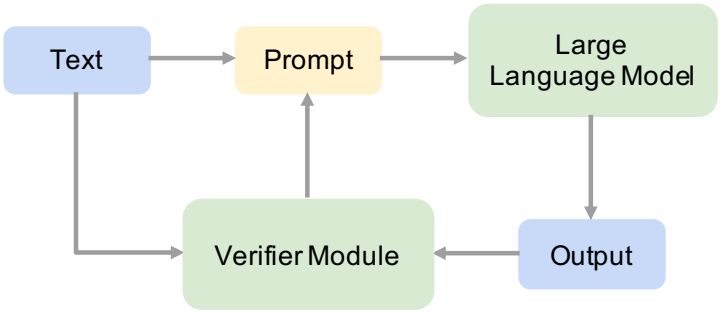


Figure 2.7: Framework for the PiVE model. Source: (Han et al., 2023).

LLMs have proven to perform better on complex tasks. However, incorporating structured data into these architectures remains a challenge. Unlike PLMs, integrating knowledge into LLMs poses challenges both theoretically and technically and is a relatively new and emerging field with limited existing literature.

One of the main challenges in integrating knowledge into LLMs is the high cost of pre-training these models and the large amount of data required. As highlighted in the survey presented in Section 2.4, most of the implicit and explicit methods required pre-training the model. Some of the surveyed work using LLMs are similar to those presented in the retrieval-based methodologies, presented in Section 2.4.2.3. These approaches use the knowledge graph to guide the reasoning behind a complex query or to verify the factual accuracy of the generated sentence by the LLM.

Among these works, we find the PiVE framework (Han et al., 2023). In this work, the authors address the limitations of LLMs such as ChatGPT, GPT-4, BARD2, and LLAMA in generating structured data. Despite excelling in different generative and reasoning tasks, LLMs struggle with tasks like text-to-graph (T2G) generation, especially when dealing with semantic graphs. PiVE proposes incorporating a verifier module, a smaller language model like T5, to refine LLM prompts iteratively through corrective instructions, as illustrated in Figure 2.7. This external verifier module enhances the quality of generated semantic graphs, employing a process known as Iterative Prompting during T2G generation. Results from PiVE demonstrate a significant improvement in LLM output quality by an average of 26%.

AutoKG, proposed by Chen and Bertozzi (2023), was built on the reasoning capabilities exhibited by LLMs when provided with sufficient information. AutoKG simplifies automatically generating knowledge graphs and their integration with LLMs. The proposed approach involves using fundamental strength indicators of associa-

tion instead of intricate relational patterns found in traditional KGs. AutoKG designs prompts for different KG construction tasks, such as entity typing, entity linking, and relation extraction. The prompts are adapted for KG construction using LLMs.

Similarly, LARK (Choudhary and Reddy, 2023) leverages logical queries to search for relevant subgraph contexts over KGs. They used logically-decomposed LLM prompts, recognizing that LLMs are less effective on complex prompts. This logical decomposition enhances the reasoning capabilities of LLMs, particularly in dealing with intricate queries involving chain reasoning.

In summary, these recent works highlight innovative approaches in logical and graph-based tasks, showcasing the potential of leveraging external modules and rethinking traditional methods to enhance the capabilities of LLMs.

2.7 CONCLUSION

In this chapter, we thoroughly explored the complex landscape of integrating knowledge into PLMs. We started by discussing the various literature resources available for this integration and the standard knowledge components used in these processes. We then provided an in-depth overview of KEPLMs and highlighted key models in the literature. We classified the main KEPLMs based on integration methodologies, enabling a fine-grained comparative analysis focusing on architectural differences and loss functions.

Evaluating KEPLMs is a challenging task, and a simple fine-tuning approach is inadequate for gaining explicit insights. Therefore, we discussed primary evaluation techniques prevalent in the community, including frameworks and benchmarks, and their respective strengths and limitations. These evaluations include factual recall datasets and semantic probing as integral components of this evaluative process. Additionally, we introduced task diagnostic evaluations based on axiomatic properties or constraints specific to individual tasks to improve interpretability regarding the augmented elements within the models.

Among contemporary research on LLMs and their emergent capabilities, we reviewed ongoing work focusing on KG integration. Our analysis revealed a paradigm shift, indicating increased utilization of KGs for verifying and correcting LLMs at the inference level. This marks a departure from the conventional practice of jointly pretraining both sources, as commonly observed with PLMs.

To summarize, our exploration has shed light on the multifaceted landscape of knowledge integration in PLMs, providing a foundation

for understanding the methodologies, models, and evaluation techniques employed in this field.

3

HIERARCHY-ENHANCED LANGUAGE MODELS

3.1 INTRODUCTION

In the previous chapter, we learned how different methodologies leverage the content information provided by Knowledge Bases (KBs) in the form of factual information or enriched entity and relation embedding representation. However, it is essential to note that beyond the content information provided by the KBs, the structural knowledge organization inherent to these structures, represented through the relationships, is often overlooked.

The relationships between different nodes in KBs can be classified into two categories, i.e., hierarchical and non-hierarchical relations (Krackhardt, 2014). In the realm of structural integration, more efforts have been dedicated to developing methods to integrate hierarchical relations, which typically include *hypernym* and *hyponym* relationships. The injection of these hierarchical relations can significantly improve the performance of various Natural Language Processing (NLP) tasks (Nickel and Kiela, 2017; Du et al., 2022). As a result, different approaches have been proposed to capture the hierarchical structure of KBs including hierarchical relationships, commonly called *taxonomies*. It is worth mentioning that in practice, both terms KB and taxonomies are used interchangeably.

In this chapter, we will focus on the approaches that integrate the hierarchical structure of the KBs into PLMs. We will begin by reviewing some of the main work developed with traditional static word embeddings (Section 3.2), which have made most of the progress in this domain.

In these methods, the encoding of hierarchical relations has been explored in different ways. A group of methods integrates explicit hierarchical knowledge using different elements from a taxonomy, such as their relations. However, the availability of taxonomic resources is expensive due to the high level of expert knowledge and agreement required to build them.

Another group of methods aims to integrate hierarchical representations by first learning to extract taxonomic relations from plain text and then constructing taxonomies from these extracted resources.

In our work, we primarily focus on methods using patterns or frequency co-occurrence representations of words, which align with the

PLMs conception and, therefore, with this dissertation. However, we point the reader to methods based on graph-based approaches such as encoders or KB embeddings (Bansal et al., 2014), which focus on the structure but without particular attention to the hierarchy.

Then, in Section 3.3, we review some recent work exploring this injection approach for Pre-trained Language Models (PLMs). These studies explore different methods to incorporate hierarchical information into the PLMs, such as fine-tuning lexical relationships or modeling the learning function to capture structural features. However, it is important to note that research in this area is ongoing, and the existing approaches have some limitations.

3.2 HIERARCHY-AWARE STATIC WORD EMBEDDINGS

Similar to the exploration presented in Chapter 2, the enhancement of word representations using *static word embeddings* with external knowledge constitutes a well-explored domain. The knowledge injection methods are typically divided into two main categories: *joint specialization* and *retrofitting* (Camacho-Collados and Pilehvar, 2018; Apidianaki, 2023; Chandrasekaran and Mago, 2021). *Joint specialization* models involve modifying the optimization objective of distributional-based models by integrating constraints directly into the objective function. In contrast, *retrofitting* models update word vectors in distributional models through a post-processing training step, which incorporates data generated from the specified constraints.

In the following section, we review some of the main approaches from these works that leverage hierarchical representations from the sources. Our criteria to filter out which methods fall into this category relies on two questions: *is the method using taxonomy as a resource for learning?* and *does this method learn constraints related to lexical taxonomic relations?*.

We will briefly summarize different techniques to learn hierarchical relationships, grouping them into two categories: learning lexical taxonomic knowledge and modeling hierarchical-aware vector spaces. Additionally, we discuss the different evaluation methods for these approaches, including assessments performed directly on the learned representations and evaluations based on their performance in relevant downstream tasks (refer to Section 3.2.3).

3.2.1 Leveraging Lexical Taxonomic Knowledge

The methods generally used in this context are based on supervised learning. These methods are designed to learn the lexical relationships that are inherent in a taxonomy, such as the relationships between ancestors and siblings. The process of learning these relationships involves extracting various common features and heuristics that can accurately model them. We present some of these works under two categories: learning edge-level and taxonomic-level representations.

3.2.1.1 *Edge-level representations: Hypernymy*

Hierarchical relationships play a critical role in many NLP applications. When these relationships are captured explicitly, it can significantly enhance the performance of applications such as word sense disambiguation and taxonomy reconstruction. To this end, several studies have primarily focused on explicitly identifying hypernym relationships to encode hierarchical representations (Camacho-Collados and Pilehvar, 2018; Chandrasekaran and Mago, 2021).

Two primary methods have gained attention in this exploration: *pattern-based* and *distributional* approaches.

Pattern-based methods are a popular way to extract hypernymy pairs from raw corpora. These methods involve identifying prevalent and explicit "templates" within the textual data that encode hypernym relationships. Examples of templates include *"is an instance of"* and *"is a"* that encapsulate the hypernym-hyponym relationship within word pairs. Among these, the Hearst patterns, introduced in the work from Hearst (1992), stands out as widely applied patterns. In this work, the authors identified seven distinct patterns crucial for discerning these relations.

Later research has attempted to automate the process of identifying patterns in text corpora (Snow, Jurafsky, and Ng, 2004; Shwartz, Goldberg, and Dagan, 2016). These methods mainly propose different neural network classifiers that learn lexicon-syntactic patterns or hypernymy path features under a supervised approach. Despite being a simple approach, these models are still widely used today because they can efficiently capture structured knowledge, resulting in improved performance in detecting hypernymy pairs, especially in benchmark evaluations that concentrate on hypernymy pair detection, when compared to other method (Roller, Kiela, and Nickel, 2018).

However, pattern-based methods have their limitations. They are often too specific and may not cover diverse linguistic circumstances. This makes them prone to errors resulting from idiomatic expressions, parsing discrepancies, and inherent ambiguities (Kozareva, Riloff, and Hovy, 2008).

Distributional methods have been used to overcome these limitations. These can be approached through either unsupervised or supervised strategies, where words are represented as vectors based on their distribution within extensive text corpora. Learning these representations involves specialized similarity measures designed to capture diverse lexical relationships.

Most of these works relied on the Distribution Inclusion Hypothesis (DIH) (Geffet and Dagan, 2005), which suggests that the contexts for more specific terms are a subset of those in which more general terms may appear. This principle implies that subordinate words can be replaced with more general terms while preserving the validity of the utterance.

Let us illustrate the DIH with an example. "Cat" entails "animal", which means that "animal" is a more general term than "cat". Given the phrase "A cat is sleeping." we can replace "cat" and obtain "An animal is sleeping." keeping the phrase's validity. On the other hand, "dish" does not entail "cat"; therefore, replacing the phrase "a dish is sleeping" does not make sense.

Various methods have been proposed to leverage the DIH, such as capturing hypernym word pairs (Rei and Briscoe, 2014; Roller, Erk, and Boleda, 2014) or finding hypernym paths (Shwartz, Goldberg, and Dagan, 2016). Now, we introduce some of the main works, which are classified into three groups: methods based on a projection of existing representations, methods that integrate hierarchy through loss constraints, and methods leveraging hierarchical relations in a taxonomy.

PROJECTION-BASED APPROACHES.

The potential of static word embeddings' inherent semantic knowledge has been explored to transform them into structural-based representations. One approach involves obtaining a subspace representation from these encoded vectors to specialize them in capturing the nuances of hypernym relationships.

One of the first works proposing this approach is done by Fu et al. (2014), where the authors propose a method for constructing semantic hierarchies based on word embeddings. The study observed the offset behavior in word embeddings, as done by the classical anal-

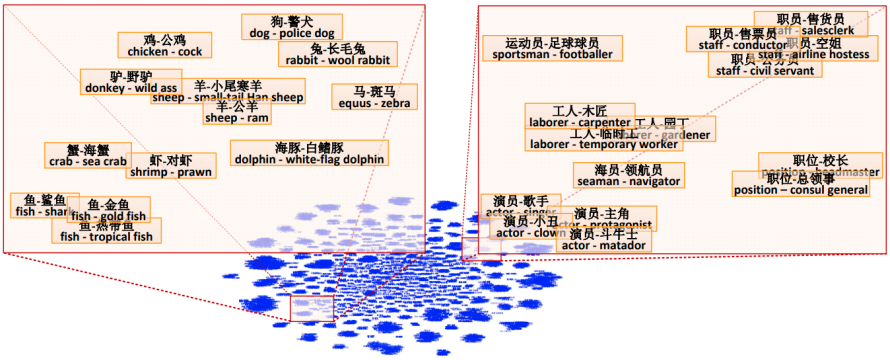


Figure 3.1: The figure shows that the vector offsets distributed clusters. The left cluster represents animal hypernym–hyponym relations, while the right represents people's occupations. Source (Fu et al., 2014).

ogy "v(king)-v(queen) ≃ v(man)-v(woman)" from Mikolov, Yih, and Zweig (2013), where v() indicates the vector embedding representation of a given word. The authors suggested that this offset behavior preserves not only typical semantic relationships but also encapsulates certain hypernym and hyponym relationships.

However, the variability in the offset values when measuring the distance between distinct hypernym pairs led to the insight that hypernym and hyponym relations are more intricate than a simple offset can represent. Figure 3.1 shows the distribution observed from these offsets. To address this, the authors proposed to learn a linear projection of word embeddings into a semantic space to identify relevant clusters for hypernyms and hyponyms.

Formally, they learn an approximative projection function $A*$, which minimizes the mean-squared error:

$$A* = \operatorname{argmin} \frac{1}{N} \sum_{(x,y)} ||Ax - y||^2 \tag{3.1}$$

with N as the number of hyponym-hypernym pairs with the form (x,y), x is the hyponymy and y the hypernym, and A the projection matrix.

In contrast to word embeddings, the TaxoEmbed algorithm proposed by Espinosa-Anke et al. (2016) employs word sense embeddings to identify hypernyms. Inspired by the work of Fu et al. (2014), they employed a function sensitive to a predefined knowledge domain, operating under the assumption that vectors clustered within the same domain would likely exhibit similar semantic properties.

Similarly, they proposed using a domain-sensitive function, learning linear projections for each domain.

Similarly to Equation 3.1, this work proposes learning dedicated projections A^d minimizing:

$$\min \sum_{i=1}^{|d|} \|A^d x_i^d - y_i^d\|^2 \tag{3.2}$$

where $|d|$ are the available domains, A^d the linear projection to learn, and the pairs (x_i^d, y_i^d) the term-hypernym sense pair corresponding to the domain d.

HIERARCHY THROUGH LOSS CONSTRAINTS.

Previous approaches to learning entity embeddings focused on hypernymy-hyponymy projections, to create a subspace that encodes this information. Another line of work proposes to transform the vector space by specializing it for a particular relationship between words.

Based on the skip-gram architecture, the work from Hu et al. (2015) incorporates *explicit structural* information from knowledge bases into distributional-based representations.

This embedding model construction is used to refine entity representations by taking into account other entities present in a context, such as within a Wikipedia paragraph. This contextual information is utilized to enhance the hierarchical structure of the entity embeddings. The methodology uses semantic distance to ensure that closely located entities share common semantic features, while entities farther apart in the knowledge base exhibit semantic distinctness. To further reinforce the hierarchical structure, the entity representation is incorporated through distance metric learning and aggregation heuristics that consider the path between entities in a taxonomy.

Another noteworthy model in this domain is the Lexical Entailment Attract Repel (LEAR) model, introduced by Vulić and Mrkšić (2018). It is a post-processing model that can incorporate external constraints in the form of hypernym relations extracted from WordNet into any word vectors. The LEAR model leverages the Attract-Repel framework (Mrkšić et al., 2017) similar to the learning constraint from Hu et al. (2015).

First, it strategically pulls desirable examples that satisfy the hypernymy constraints closer together (ATTRACT) while simultaneously pushing undesirable word pairs (REPEL) away from each other. Second, LEAR also has the ability to rearrange vector norms in the

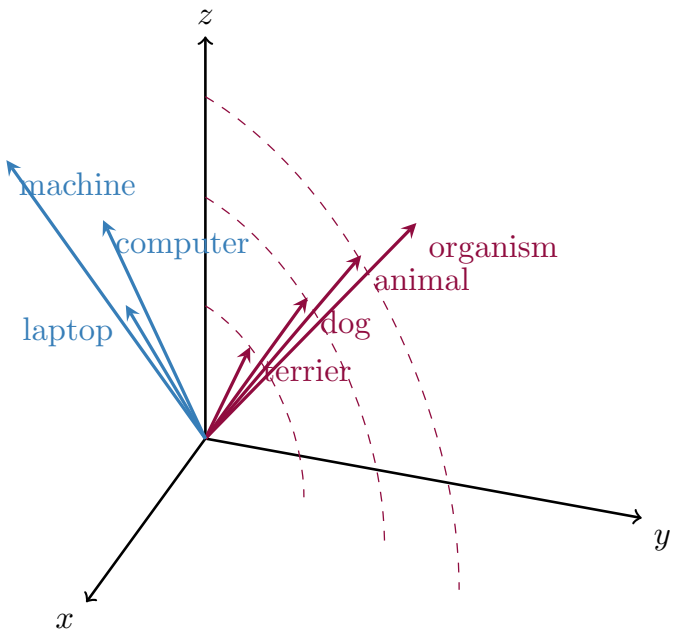


Figure 3.2: Illustration of LEAR representations. LEAR controls the arrangement of vectors in space, emphasizing symmetric similarity and imposing an LE ordering based on vector norms. Norms are adjusted, so higher-level concepts have larger norms (e.g., animal > dog > terrier). Source (Vulić and Mrkšić, 2018).

Euclidean space to reflect the hierarchical organization of concepts based on the provided lexical entailment constraints (refer to Figure 3.2).

This dual functionality of LEAR enables it to capture both the hierarchy of concepts, which is manifested through vector norms, as well as their similarity, which is expressed through cosine distances. Moreover, LEAR is explicitly designed for the lexical entailment task. It uses an asymmetric distance function based on the norm of the representations and the offset between them, providing a nuanced perspective on hierarchical relationships within the semantic space.

LEVERAGING HIERARCHICAL RELATIONS.

The use of taxonomies as an external resource in NLP has been gaining attention in recent years. In this regard, Alsuhaibani, Maehara, and Bollegala (2018) proposed a novel method for encoding hierarchical word embeddings by leveraging both a plain text corpus and a taxonomy. The authors extracted *hierarchical hypernymy paths* for a given word to assimilate structural information from the tax-

onomy. They learned a vector based on contextual information from hyponym and hypernym words. The representation of a word was crafted by considering the hierarchical information inferred from its parent nodes along the *hierarchy path*. In this methodology, the authors aggregated all parents in the *hierarchy path* by summing their embeddings, incorporating a discounting term tied to the distance between the parent and the target word. Formally, they defined an objective function $\mathcal{J}$, for all the vocabulary $\mathcal{V}$, considering the hypernym path $\mathcal{P}(w)$, and a discount term λ to assign a weight for each hypernym word in the path:

$$\mathcal{J} = \frac{1}{2} \sum_{w \in \mathcal{V}} \|w - \sum_{p \in \mathcal{P}(w)} \lambda(p) p\|_2^2 \tag{3.3}$$

For example, if we consider the hierarchy path "bird → vertebrate → chordate → animal" and the word "bird", the embedding for "bird" is influenced more by the direct hypernym "vertebrate" than the indirect hypernym "animal".

Liu et al. (2021) proposed a new method to capture implicit information in taxonomy resources that accounts for multiple meanings of a word. Their approach uses word embeddings to represent the taxonomy's hypernym, hyponym, and sibling relationships. To capture more nuanced semantic relationships, the method uses an attention mechanism at the term level and a random walk-based metric (refer to Section 3.3.2 for further details about random-walk approaches).

In NLP, some tasks involve using hierarchical concepts explicitly. This requires the use of hierarchical knowledge injection when developing approaches to these specific tasks. For instance, some examples of such tasks include Named Entity Typification using hierarchy loss and word embeddings (Xu and Barbosa, 2018) or hierarchical text classification with end-to-end reinforcement learning (Mao et al., 2019). Although these tasks extend beyond the scope of this dissertation, we will briefly discuss them in Section 3.3.2 to gain a better understanding of the structural features used.

3.2.1.2 *Taxonomy-level representations: Taxonomy Reconstruction*

Considerable research has been conducted on the recognition of lexical semantic relationships, but relying solely on word pairs may not consistently capture the various hierarchical characteristics inherent in these sources (Alsuhaibani, Maehara, and Bollegala, 2018; Camacho-Collados, 2017). To address this, an additional avenue for learning structured representations involves the taxonomy recon-

struction task. In this task, a model is expected to rearrange a list of words extracted from texts into a taxonomy-shaped structure.

The potential benefits of using multiple edges to refine representations become particularly apparent when employing graph neural networks (GNNs). These networks enable the encoding of subgraph structures, which can subsequently be jointly trained using Hearst patterns to capture hierarchical relationships such as parent and sibling features (Bansal et al., 2014). Alternatively, they can be combined with encoder-decoder architectures to leverage their strengths (Shang et al., 2020).

Apart from GNNs, Gupta et al. (2016) employs heuristics to retrieve hypernym paths in text corpora. This technique diverges from conventional methods that focus solely on hypernym pairs. By seeking better generalization paths that align more closely with correct taxonomy representations, this approach has shown promise in improving the accuracy of taxonomies.

However, there is a lack of exploration regarding *distributional* techniques to capture higher levels of hierarchy in the existing literature. Typically, reconstructing a taxonomy involves two steps: extracting hypernym pairs and organizing terms into a structured taxonomy. To bypass this potentially bottlenecked process, Gupta et al. (2017) adopted a probabilistic framework to extract longer hypernym subsequences. Using a minimum-cost flow-based optimization approach, their framework focuses on inducing a tree-like taxonomy from a noisy hypernym graph.

In order to address the issue of over-reliance on detecting hypernym relationships alone, Mao et al. (2018) proposed a unique reinforcement learning approach that tackles both the hypernym relationship identification and taxonomy reconstruction tasks simultaneously. The authors utilized various information sources to model hypernym relationships, including dependency path-based contextual embeddings derived from the dependency path of co-occurring words and distributional term embeddings, as previously explored by Shwartz, Goldberg, and Dagan (2016). Furthermore, they incorporated frequency, generality, and surface string features to identify hypernym relations based on capitalization, common substrings, and other criteria.

On a different note, Shen et al. (2018) introduced hierarchical characteristics into their model by considering three types of features: skip-patterns based on the patterns approach, term embeddings to encompass the semantics of each term, and entity types retrieved for a given word. These features were learned to predict hypernym relationships and extend existing taxonomies.

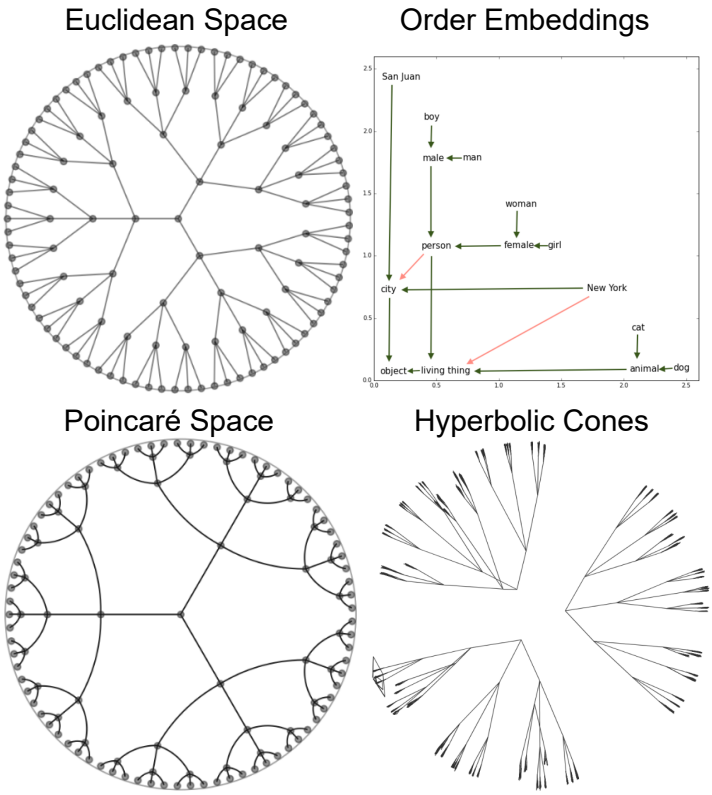


Figure 3.3: Distribution examples for different vector space representations.

3.2.2 Defining Hierarchical Vector Spaces

The preceding methodologies address, to some extent, the hierarchical structure within knowledge bases by modeling subordinate relationships, primarily focusing on hypernymy. However, these methods typically attempt to capture intricate patterns based on distributional approaches, leading to certain limitations. Firstly, Euclidean space is used to map nearby objects, employing *symmetric* distances such as Euclidean or cosine distance to represent a semantic hierarchy, which is an inherently *antisymmetric* relation. This leads to systematic model errors. Secondly, the representations are restricted by the dimensionality of the embedding space. Notably, the research by Nickel, Jiang, and Tresp (2014) demonstrated that linear embeddings of graphs may necessitate prohibitively large dimensionality to model certain types of complex relations.

We now introduce different families of non-Euclidean vector spaces used to integrate and model hierarchical structures. Figure 3.3 shows how elements distribute differently according to the defined vector space.

3.2.2.1 *Order Embeddings*

To address the first limitation, the approach proposed by Vendrov et al. (2015) introduces *order embeddings*, which circumvent the use of a distance-preserving approach in favor of an order-preserving one. They map a *partial order* over the embedding space. The authors proposed a task called partial order completion to learn these *order embeddings*. This task involves predicting whether a given pair of words is ordered or unordered. The training process involves positive, ordered pairs and negative sampling with unordered pairs. While this training task is applicable to hypernym detection pairs, the authors also applied it to caption-image retrieval tasks.

3.2.2.2 *Hyperbolic Embeddings*

Despite improvements in the inner nature of distributional-based approaches, the default representation space, typically Euclidean, remains limited. Addressing the second limitation mentioned earlier, Nickel and Kiela (2017) advocates for the use of *hyperbolic spaces* to encode the inner hierarchical structure of knowledge bases. They model word embeddings based on *hyperbolic space* geometry, considering the underlying tree-like structure they possess. Hyperbolic spaces exhibit non-Euclidean geometry with a constant negative curvature, similar to continuous versions of trees, making them ideal for representing hierarchies.

In their work, Nickel and Kiela (2017) specifically used the *Poincaré ball* model, a well-suited hyperbolic space model for gradient-based optimization. The efficiency of this space is demonstrated through the successful completion of link prediction and lexical entailment tasks, which confirm that this embedding space corresponds well to the underlying semantics of the data.

3.2.2.3 *Convex Entailment Cones*

However, the use of Poincaré embeddings, based on the hyperbolic distance, which is *symmetric*, does not encode the *asymmetric* characteristic of hypernym relations. To address this issue, Ganea, Bécigneul, and Hofmann (2018a) built upon the *partial order* embedding proposed by Vendrov et al. (2015) and uses geodesically *convex entailment cones* to induce a partial ordering relation in the embedding space. The authors' proposed method of using *entailment cones* in the Poincaré ball encoding space provides several advantages, including:

1. Encoding *asymmetry*,

2. Representing the *degree of hypernym* relation using continuous cone aperture functions, and

3. Capturing the *transitivity* between hypernym pairs using transitivity of nested angular cones.

Aly et al. (2019) further demonstrated the effectiveness of Poincaré embeddings in refining taxonomies by correcting wrong hypernym predictions.

It's important to note that even though hierarchical embeddings can enhance performance on tasks that rely on hierarchy, like lexical entailment, they may have a negative impact on word similarity tasks. In fact, Dhingra et al. (2018) demonstrated that unsupervised encoding of text objects beyond explicit hierarchical structures could lead to a decrease in performance.

Although hyperbolic space representations offer several advantages, their adoption has been limited due to a lack of available tools that can generalize basic operations such as vector addition, matrix-vector multiplication, vector translation, and vector inner product, as well as closed-form expressions like distance and geodesics in complex geometries. To address this issue, researchers have adapted default frameworks for Euclidean spaces to work seamlessly in hyperbolic spaces for neural networks, such as multinomial logistic regression, feed-forward, simple and gated recurrent networks, and the attention mechanism. Notable works in this area include Ganea, Bécigneul, and Hofmann (2018b) and Gulcehre et al. (2018). Some studies have proposed applications of these adaptations, such as HyperText, which learns FastText embeddings with hyperbolic geometry (Zhu et al., 2020).

The HypeHan model, introduced by Zhang and Gao (2021), is based on an architecture that focuses on three levels of embeddings: word, sentence, and document. It combines different strategies explored in previous work. HypeHan models the hierarchy at three different levels using two layers of hyperbolic attention mechanisms. The architecture begins with word representation, utilizing typical static word embeddings, and applies hyperbolic attention through these words to obtain sentence embeddings, which are then projected once more into a hyperbolic space to learn document embedding representations. In addition to this method, another approach has been proposed by Sato et al. (2022) to deal with unseen words during training.

The adoption of hyperbolic geometry is not limited to distributional-based approaches but extends to its applications in developing

knowledge graph embeddings (Balazevic, Allen, and Hospedales, 2019) and graph convolutional networks (Shen et al., 2021).

3.2.2.4 *Other Non-Euclidean Vector Spaces*

Many studies have adopted hyperbolic representations that use various geometric tools like manifold space projections. This has made them easily adaptable to existing frameworks. However, it is worth noting that there are other approaches based on the spatial characteristics of hypernymy and hierarchical structures.

An alternative avenue of exploration is through *spherical text embeddings*, as proposed by Meng et al. (2019). These embeddings capture directional similarities within a spherical space with a positive constant curvature, which differs from hyperbolic spaces.

The predominant methodologies employed to capture hierarchical relationships are founded on defined connections between elements in a taxonomy. However, an alternative strategy adopts the use of *Markov random walks* to assimilate diverse interactions within the hierarchical structure (Mao et al., 2019). This method has been proven to be effective in capturing complex hierarchical relationships.

Lastly, rooted in partial order theory, the concept of Strict Partial Order Networks (SPON) (Dash et al., 2020) is introduced to enhance representations. In this framework, the authors propose a neural network architecture that enforces *strict partial order* as a soft constraint, encompassing symmetry and transitivity. By incorporating these principles, the SPON model can effectively capture hierarchical relationships' nuances and improve the representations' overall quality.

3.2.3 Evaluation Approaches

The evaluation of knowledge encapsulated by enhanced word embeddings follows a methodology similar to that outlined in Section 2.4. In the literature, these evaluation methods can be broadly classified into *intrinsic* and *extrinsic* evaluations (Schnabel et al., 2015).

Intrinsic evaluations directly examine semantic relationships between words. In the context of structure-based injection methodologies, these assessments primarily focus on taxonomic relationships, such as hypernym or siblings. However, intrinsic evaluations have certain limitations, as the observed performance in these tests does not consistently correlate with the effectiveness of the embeddings in downstream tasks (Faruqui et al., 2016).

On the other hand, *extrinsic evaluations* incorporate word embeddings as input features within a more extensive network, which is

deployed for a specific downstream task. The evaluation involves comparing the model's performance on the given task. Despite their task-specific nature, *extrinsic evaluations* may not provide comprehensive insights into the inherent quality of knowledge representation (Faruqui et al., 2016).

INTRINSIC EVALUATION.

Static word embeddings have been extensively investigated for encoding hypernym relationships. Such relationships are often referred to as lexical entailment, and evaluating them has led to the creation of numerous datasets, such as BLESS (Baroni and Lenci, 2011) and BiBLESS (Kiela et al., 2015). These datasets use a binary decision framework to determine whether a given word pair exhibits a hypernym relationship or not.

However, the focus of these evaluations is often limited to the hypernym relationship itself, without fully capturing the broader hierarchical structure they aim to encode. Many distributional-based approaches run the risk of relying on single-word features and may fall short of learning lexical inference relations, commonly referred to as *"prototypical hypernyms"* (Levy et al., 2015). These models tend to memorize frequently encountered hypernyms without fully grasping the nuances of hierarchical features.

Therefore, there is a crucial need for more nuanced evaluations and learning methods to capture higher granularity levels beyond binary detection (Camacho-Collados, 2017).

Recognizing the limitations of the rigid binary evaluation approach, Vulić et al. (2017) introduced a *graded lexical entailment* evaluation dataset known as HyperLEX. Departing from the binary decision framework, HyperLEX adopts a more human-centric approach to assess lexical entailment. Instead of asking whether a word pair follows a hypernym relationship or not, HyperLEX annotates human expert judgments on the question *"To what degree X is a type of Y"*. This graded approach aligns more closely with human cognition, incorporating two cognitive science phenomena: typicality, where certain categories and instances exhibit stronger relationships due to their frequent usage, and graded membership, acknowledging the lack of a clear definition for the class into which an object fits within a given category.

Complementary, the work by Gupta et al. (2016) and Gupta et al. (2017) introduces a new approach to evaluate taxonomies by considering *hypernym paths* instead of just hypernym pairs. This method evaluates individual edges and path-level evaluations, which assess

hypernym paths. The criteria for this evaluation include assessing the specificity or generality of hypernym predictions and evaluating correct prefix paths rather than individual pairs.

EXTRINSIC EVALUATION.

Extrinsic evaluations leverage hierarchical knowledge in enhanced static word embedding models for downstream tasks like named entity recognition, information retrieval, and taxonomy reconstruction.

The TexEval datasets, TexEval I (Bordea et al., 2015) and TexEval II (Bordea, Lefever, and Buitelaar, 2016), are particularly useful for the taxonomy reconstruction task. These datasets provide diverse domain-specific taxonomies for evaluation, grounded in WordNet and spanning various languages. The goal of this task is to deduce the appropriate taxonomy structure by discerning hypernym and hyponym relationships among given terms. The proposed metrics within these evaluation sets include accuracy at the edge level, structural evaluations involving criteria like the number of cycles in the predicted taxonomy, and cluster-based metrics, which offer a higher-level assessment of term distribution across different depth levels of the taxonomy.

Instead of relying on binary decisions regarding whether two nodes share a relationship, another evaluation type seeks to capture various potential hypernyms by adopting a ranking approach. For instance, Hypernym Discovery (Camacho-Collados et al., 2018) adopts this approach. Using an information retrieval approach, this task seeks to evaluate hypernym predictions for a given word within a more nuanced framework.

3.3 HIERARCHY-AWARE PRE-TRAINED LANGUAGE MODELS

The infusion of hierarchy into PLMs has been a natural progression in this research domain after static word embeddings. However, their exploration is still relatively new and constantly evolving.

In this context, we review some recent models and categorize them into two primary classes. Firstly, we discuss the methods based on *lexical relations* to improve hierarchy at the word level. Secondly, we delve into various approaches that harness *structured signals* within the corpus or knowledge resources at distinct and more abstract levels, which are customized for diverse applications.

3.3.1 Leveraging Lexical Relations

The utilization of PLMs for encoding extensive text corpora has been widely investigated as a means to learn and extract knowledge about lexical semantic relationships. Similar to the preceding Section, exploring hierarchical signals in text involves acquiring taxonomic relationships. A simple yet effective method for learning these relationships involves fine-tuning.

The LEXFIT framework (Vulić et al., 2021) adopts this approach, aiming to enhance lexical knowledge from PLMs in a decontextualized setting by employing a *bi-encoder* architecture. This means LEXFIT learns lexical relations between pairs of words without considering additional context, similar to traditional *static word embeddings*. This strategy capitalizes on the inherent knowledge encoded in PLMs, adapting them to specific lexical relationships. Specifically, LEXFIT uses a Sentence BERT encoder architecture (Reimers and Gurevych, 2019) with various contrastive loss functions and demonstrates improved performance when using larger batches during the training phase of fine-tuned PLMs. The reader can refer to Appendix A for further information about bi-encoder architectures.

Another endeavor, Constructing Taxonomies with Pre-trained models (CTP) by Chen, Lin, and Klein (2021), follows a comparable approach by employing a cross-encoder. CTP leverages the contextualized embedding representations of PLMs to identify robust taxonomic relations. Unlike LEXFIT, CTP uses a classification layer atop PLMs to discern hypernymy between word pairs. In this study, words extracted from WordNet are enriched with contextual information, referred to as glosses, to provide the model with additional information, particularly in polysemous word cases. In the context of taxonomy reconstruction, CTP successfully predicts taxonomies with small and medium depths. However, it struggles when dealing with deeper taxonomies, tending to predict flat taxonomies, and facing challenges defining abstract classes at higher levels.

Similarly, another approach to identifying hypernym relationships and reconstructing taxonomies is explored by Jain and Anke (2022) under a zero-shot setup. The authors leverage prompt inputs and apply the LMSCorer approach (as discussed in Section 2.5.1.1). While this approach proves to be cost-effective compared to fine-tuning, its results are still limited in performance.

In a subsequent work, Takeoka, Akimoto, and Oyamada (2021) additionally incorporates Hearst patterns to enhance the performance of PLMs in predicting hypernym relationships. The authors focused on extending existing low-resource taxonomies, framing it as a tax-

onomy expansion task consisting of a Hearst pattern generator and a PLM classifier.

PLMs are still widely used to extract and learn lexical relationships. A recent study by Liu, Cohn, and Frermann (2023) examined various hypernym extraction prompts using a BERT model, taking into consideration different patterns and the Masked Language Model (MLM) task under a zero-shot setup. The authors discovered that incorporating *rare hypernyms* and hyponyms in *prompts* enhances the performance of PLMs. Furthermore, including an anchor element, a third element in the context, improved hypernym detection performance compared to using patterns with only two words.

3.3.2 Leveraging Structured Signals

We introduce some of the approaches relying on random-walk strategies, hyperbolic vector spaces, and high-order hierarchical components.

3.3.2.1 *Random Walk Strategies*

As discussed earlier, while conventional hypernym-based approaches are helpful, they often fail to capture the intricate information inherent in hierarchical relationships. Novel local heuristics have been developed to overcome this limitation and unlock the full potential of existing taxonomies. One such approach is the utilization of *random walk* methodologies, which has shown effectiveness in extracting comprehensive hierarchical information from graphs (Chen et al., 2020b).

Considering the elements of a KB, a *random walk* is a process where an entity moves through a graph by taking random steps from one node to another. During each step, the walker chooses a neighboring node randomly, and the probability of transitioning to a particular neighbor is either proportional to the edge weights or uniformly distributed across all neighbors. This method is a more generalized approach than the classical approach of hypernym pairs, where a node is associated only with its direct ancestor.

One contribution in this domain is TaxoPrompt, introduced by Xu et al. (2022). TaxoPrompt blends the principles of *random walk* with a PLM to devise a taxonomy expansion framework. By employing a *prompt-based* approach, the authors advocate using specialized templates for hypernym generation, facilitating the acquisition of lexical-syntactic features within a taxonomic context. The overarching objective is to generate hypernyms that exhibit structural consistency, informed by the acquired knowledge of an existing taxonomy. Taxo-

Prompt leverages *random walk* techniques to sample relations such as "parent-of", "child-of", "sibling-of", "nephew-of", and "posterity-of" from a given node to achieve this. This diverse set of traversal options for the KB enables the model to capture various forms of implicit hierarchical knowledge.

Similarly, Liu et al. (2021) have explored the integration of *random walk* techniques within a KB, focusing specifically on the relationships between hypernym, hyponym, and siblings. Their research aims to cultivate lexical taxonomical embeddings by adopting a curriculum learning approach. The application of random walk in this context proves to be a critical tool for navigating the intricate web of relationships within the KB, contributing to a more nuanced understanding of hierarchical structures.

3.3.2.2 *Hyperbolic Vector Spaces*

As described in Section 3.2.2, the utilization of hyperbolic space for representing hierarchical structures has been identified as advantageous due to its exceptional capacity and continuous tree-like properties. Consequently, some researchers in the field of PLMs have recently explored the use of hyperbolic geometry to encode hierarchical information more effectively. However, this exploration remains limited to date.

Many approaches rely on the following working mechanism. These techniques use contextualized representations that are projected into a hyperbolic space. The Poincaré ball model is usually preferred as it is easy to optimize. Once the projections are made, various distance-based metrics are utilized to capture *semantic similarity* distances between different elements. This process enables the characterization of the hyperbolic nature of these elements.

One noteworthy exploration in this direction is presented by Song, Feng, and Jing (2022a), which focuses on the task of Multi-Document Summarization. By employing PLMs and hyperbolic spaces, the authors leverage a *semantic similarity* approach to identify accurate summary candidates. By projecting these candidates into a hyperbolic space, specifically the Poincaré ball, the latent hierarchical structure inherent in the text information is better captured. This involves contextualized representation projection and learning using the Poincaré distance within a contrastive learning setup.

Building upon the foundation of Song, Feng, and Jing (2022a), HISum (Song, Feng, and Jing, 2023) extends the approach by considering global and local interactions within the same hyperbolic space. While maintaining a global view akin to HyperMatch, HISum incorporates hyperbolic Convolutional Neural Network

(CNN) encoders for local interactions. These encoders derive n-gram phrase representations, enabling the model to generate phrase-aware summaries and document representations via the Poincaré distance.

Similarly, HyperMatch (Song, Feng, and Jing, 2022b) employs a hyperbolic projection approach for keyphrase extraction in documents. This technique leverages the hierarchical structure of keyphrases, with longer keyphrases inherently possessing more complex structures. Employing the Poincaré distance in hyperbolic space, the relevance between a document and its keyphrase candidates is determined after obtaining and projecting PLM representations for both elements.

3.3.2.3 *High-order hierarchical components*

Many current works utilize elements from taxonomy, including entities and relationships. However, other approaches consider high-order hierarchical components, such as the hypernym path of a taxonomy, or the hierarchical structure of a corpus of documents, where the nodes are represented by the documents and the edges by the links between them.

HYPERNYM PATH

An alternative avenue for injecting hierarchy into models focuses particularly on the relation between an element and its hypernym ancestors. A notable NLP application where this approach yields direct benefits is Fine-grained Entity Type Classification (FET), a task focused on labeling entity mentions in context with one or more specific types organized within a hierarchy (e.g., identifying "actor" in the hypernym path "actor $\rightarrow$ artist $\rightarrow$ person). Integrating the hypernym path into these models improves the performance on this type of task and related applications, which also benefit from learning taxonomic distributions, such as relation extraction, question answering, and entity linking (Xu and Barbosa, 2018).

The use of box embeddings (Onoe et al., 2021) has been explored to represent entity-type embeddings within a high-dimensional space. In this paradigm, each entity mentions and entity type is embedded into the same box space, enabling the model to discern relationships between them. The geometric properties of these boxes, including nesting, overlap, or complete disjointness, facilitate the capture of diverse relationship characteristics, notably hierarchical nuances. In this context, PLMs serve to transform entity mentions and contexts into the box space.

Similarly, the methodology employed by PICOT (Zuo et al., 2022) employs a contrastive learning strategy, utilizing *prompts* fed into a PLM model to capture the hierarchy of fine-grained types. The *prompts* dynamically adapt to the granularity of the entity type, incorporating both coarse-grained and fine-grained distinctions. This adaptive approach enhances the model's ability to distinguish between multi-grained similar types.

Furthermore, in the work from Chen et al. (2022), sibling relationships are considered to bolster hierarchical representations. This observation is consistent with hypernym encoding approaches that recognize the significance of sibling relationships in capturing hierarchical structures. By incorporating sibling relationships into the model, it is possible to improve the hierarchical representations of the data further.

DOCUMENT-LEVEL STRUCTURE.

A complementary line of work exploits the document-level structure for improving entity representations. One of these works is ERICA (Qin et al., 2021), presented in Section 2.4.1, which goes beyond the within-sentence relations and considers interactions among multiple entities at the document level. ERICA captures the interaction between entities *across different paragraphs* from different documents and *within paragraphs* from the same document, using two pre-training tasks called Entity Discrimination and Relation Discrimination

Another recent work by Raman, Shah, and Veloso (2022) proposes an extensive approach that integrates various hierarchical structures inherent in documents. The authors model document representation by considering both semantic similarity and structural relationships embedded within the underlying link structure that connects documents. This enables them to capture the complex interplay between different document components, leading to a deeper understanding of the document's meaning and context.

Table 3.1 lists all the presented models, summarizing the leveraged hierarchical feature and the primary approach to inject (or extract) the hierarchy knowledge.

3.4 CONCLUSION

The field of augmenting language models with external knowledge has become a crucial area of focus in recent research. While current approaches primarily focus on utilizing explicit information from external sources, the implicit hierarchical knowledge they inher-

Model	Hierarchical feature			Method					
	Hyper-nym	Random Walk	Distance	Fine-tuning	Zero-shot	Pattern	MLM	Poin-caré	Graph based
TaxoPrompt (Xu et al., 2022)		x				x	x		
LEXFIT (Vulić et al., 2021)	x			x					
CTP (Chen, Lin, and Klein, 2021)	x			x					
Takeoka, Akimoto, and Oyamada (2021)	x			x		x			
Liu et al. (2021)		x							x
Onoe et al. (2021)			x	x			x		
ERICA (Qin et al., 2021)			x	x					
Jain and Anke (2022)	x				x	x			
Song, Feng, and Jing (2022a)			x					x	
HyperMatch (Song, Feng, and Jing, 2022b)			x					x	
PICOT (Zuo et al., 2022)	x			x		x			
Chen et al. (2022)	x	x							x
Raman, Shah, and Veloso (2022)			x	x					
Liu, Cohn, and Frermann (2023)	x				x	x			
HISum (Song, Feng, and Jing, 2023)			x					x	

Table 3.1: List of the models presented in this section, including the leveraged hierarchical feature and the main method to inject (or extract) the hierarchy knowledge.

ently possess has often been overlooked. Considering two distinct types of relationships within the realm of KBs - hierarchical and non-hierarchical - this chapter focused on the hierarchical injection of knowledge and evaluated their integration into both static word embeddings and more contemporary contextual-based PLMs.

Initially, we explored various strategies that employ *static word embeddings* and categorized them based on the type of hierarchical representation they aim to learn. Notably, we examined approaches focused on learning word-pair relations, such as hypernymy, and those delving into higher levels including hierarchical paths and subgraphs from KBs. Among the strategies for learning hypernym relations, we identified *projection-based* approaches that seek subspace representation, methods that employ *loss functions* translating hierarchical distance into semantic distance, and other approaches that extend *beyond hypernym constraints*.

Later, we delved into *static word embedding* techniques that tailor specific geometries to capture the inherent exponential expansion of hierarchical relationships accurately. This involves introducing *order embeddings* that define a geometric space aligning with *hypernym characteristics*, such as directionality and transitivity. Additionally, we presented Poincaré word embeddings, which are situated in the *Poincaré ball space* - a hyperbolic space - and can facilitate the encoding of hierarchical knowledge. The section concluded by outlining *intrinsic* and *extrinsic* evaluation methodologies that can be used to assess the effectiveness of the encoded information in these approaches.

Finally, moving on to similar endeavors within PLMs, we introduced the emerging field that explores encoding hierarchical knowledge into these models. While this domain is still evolving, recent work has primarily involved fine-tuning approaches that aim to imbue PLMs with lexical taxonomical relationships, aligning with hierarchical constraints. We also highlighted complementary work that expands the scope of hierarchy within PLMs, incorporating hyperbolic spaces and random walk strategies to capture a broader spectrum of hierarchical relations. Furthermore, we introduced models that directly model the structure of abstract elements such as documents, reflecting a holistic approach to hierarchical knowledge integration.

Part II

CONTRIBUTIONS

4

IMPROVING IMPLICIT KNOWLEDGE INTEGRATION INTO PLMS WITH INCREMENTAL REASONING

4.1 INTRODUCTION

A practical and affordable method to enhance PLMs is implicitly integrating information from the knowledge graph (KG) with text using text templates or descriptors, as presented in Chapter 2. Particularly in Pre-trained Language Models (PLMs), this approach is beneficial in applications such as question-answering and reading comprehension (Talmor et al., 2019) thanks to their adaptability to perform reasoning-based tasks, such as fact recalling or reasoning inference (Petroni et al., 2019; Kassner and Schütze, 2020). Consequently, multiple state-of-the-art (SOTA) models aim to enhance PLMs' reasoning capabilities by integrating hypernym path structures from a KG (Clark, Tafjord, and Richardson, 2021).

These reasoning-based tasks have varying difficulty levels, depending on the complexity of the required inference to find the answer. For instance, in the reading comprehension task, one is given a question and a text and then expected to find the answer to the question within the text. The complexity of the reading comprehension task can range from simple fact retrieval to performing inferences across multiple paragraphs from the text source. Therefore, to cover the maximum types of difficulties possible, SOTA models simultaneously learn different reasoning complexities by adopting a multi-task setup (Richardson et al., 2020; Talmor et al., 2020b), presented in Section 4.3.

Moreover, the incorporation of PLMs brings additional difficulties. Recent research has shown that PLMs not only rely on information learned from KG but also use information acquired during pre-training (Kassner, Krojer, and Schütze, 2020; Talmor et al., 2020b) when performing reasoning tasks. This behavior of PLMs makes it challenging to distinguish what the model is genuinely learning and determine which elements in the learning process are necessary to perform these tasks. In response to this problem, different probes have emerged to elucidate the capabilities and limitations of PLMS in this domain (Richardson et al., 2020; Talmor et al., 2020a).

In this chapter, we introduce our first contribution, which addresses the two challenges previously mentioned. We focus on understanding the interactions between different difficulty levels to better cope with the challenges of the task. Also, we investigate the complexities of incorporating PLM by exploring various strategies based on contextual information. Our primary objective is to gain insights into the behavior of models trained on a specific difficulty level, which we call single-hop reasoning models. By doing so, we expect to develop more predictable models.

Our method focuses on multi-hop reasoning (introduced in Section 4.3). We aim to mimic human-style reasoning, breaking down the task into a sequence of explicit single-hop reasoning steps in an incremental learning approach. To equip PLMs with incremental reasoning skills, we propose a set of inference strategies on relevant facts and distractors (Section 4.3). These strategies allow us to build automatically generated training datasets. Finally, we assess the transferability of these incremental learning reasoning skills to related question-answering tasks.

The remainder of this chapter is structured as follows. We describe the context of the problem and main contributions in Section 4.2. Then, we describe our experimental setup, including the datasets and models for our experiments, in Section 4.4. Later, we study the impact of applying these different incremental reasoning strategies on the reasoning task and downstream tasks such as question-answering and reading comprehension in Section 4.5. Finally, Section 4.6 lists our conclusions.

4.2 OVERVIEW AND CONTRIBUTIONS

Multi-hop reasoning problems involve performing logical inference that requires traversing multiple intermediate steps or connecting pieces of information to arrive at a conclusion. State-of-the-art PLMs, such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019), have proven successful in solving multi-hop reasoning problems, including multi-hop question-answering (QA) tasks (Weber et al., 2019; Richardson and Sabharwal, 2020; Saxena, Tripathi, and Talukdar, 2020; Saha, Yadav, and Bansal, 2021) and multi-hop reading comprehension (RC) (Min et al., 2019; Ding et al., 2019).

Training multi-hop reasoning tasks specifically implies a two-step process, illustrate in Figure 4.1:

1. Distinguish -within a context- the relevant facts from the distractors to be used for reasoning; both relevant facts and distractors

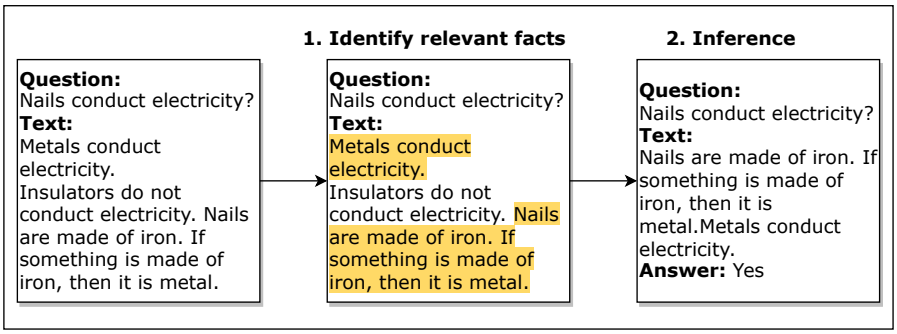


Figure 4.1: The multi-hop reasoning process involves 2 steps: 1) Identify relevant facts from distractors in a context. 2) Inference.

are generally expressed as statements in natural language using linguistic patterns (Clark, Tafjord, and Richardson, 2021);

2. Reasoning over a sequence of relevant facts leading to chains of reasoning (Das et al., 2019).

A common approach to teaching PLMs to solve a multi-hop reasoning task is to convert the structural reasoning task into subtasks that model the sequence of reasoning tasks. For instance, Richardson and Sabharwal (2020) and Clark, Tafjord, and Richardson (2021) rely on a multitasking training strategy (Caruana, 1997) that uses training instances mixing different depths of reasoning steps (hops). In the same line, (Min et al., 2019) and (Ding et al., 2019) carry out several steps of single-hop reading comprehension to simulate multi-hop reasoning.

However, while yielding impressive results, it is still unclear if PLMs endowed with multi-hop reasoning skills really leverage the learned skills at each *single-hop* depth level along the reasoning chain. More specifically, our contribution is motivated by observing that PLMs yield unpredictable results while performing multi-hop reasoning. For instance, previous studies show that the performance of PLMs degrades substantially even with a slight increase in the number of hops in the underlying reasoning tasks (Richardson and Sabharwal, 2020). This result indicates that multi-hop models at lower depths struggle to transfer information to deeper-hop models, giving rise to the *compositionality generalization* (Chaabouni et al., 2020) issue from simpler to complex tasks.

We advocate that a better understanding of the inherent relationships between the different single-hop reasoning models allows the design of more predictable models. Our contributions seek to answer three main questions:

- *RQ1*: grounded in previous findings in the literature (Richardson and Sabharwal, 2020) showing the compositionality generalization issue, *do single-hop reasoning models incrementally learn?*
- *RQ2*: inspired by the human reasoning style to solve complex problems based on simpler ones (Anderson, 1980), *can PLMs be guided toward incremental reasoning?*
- *RQ3*: grounded on previous findings revealing that PLMs trained on one specific reasoning task improve their performance on different and unrelated reasoning tasks (Talmor et al., 2020b), *do QA tasks leverage incrementally trained reasoning models?*

4.3 METHODOLOGY

In this section, we formally present the multi-hop reasoning task, and introduce the proposed data probe generation methodology.

4.3.1 Multi-hop Reasoning

We focus on the multi-hop symbolic reasoning task over explicit knowledge. Following previous work, our setting includes the following:

1) a **knowledge graph** (KG) $G = (\mathcal{E}, \mathcal{R})$ as a graph containing a set of entities (represented as nodes) $(e \in \mathcal{E})$, and a set of inference relationships, represented as edges $(r \in \mathcal{R})$. Additionally, it contains a set of real relation *facts* f_{ij} as positive triples (e_i, r, e_j) denoted F^+ among all the possible ones in $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$;

2) a **hypothesis** $\mathcal{H}_{ij}^k$ about the inferred relationship $r*$ between two target entities $(e_i, r*, e_j)$ in the KG that are separated by k hops in the graph G;

3) a **hypernym inference path** of depth k on G, referred to as $\mathcal{I}_{ij}^k$, allowing to build a new relation fact $f_{ij}^* \notin F^+$ by combining $k+1$ relation facts along the reasoning chain $< (e_i, r_0, e_1)(e_1, r_1, e_2) \dots (e_k, r_k, e_j) >$, such as $\forall 0 \leqslant n \leqslant k$, $(e_n, r_n, e_{n+1}) \in F^+$, $(e_i, r_0, e_1), (e_k, r_k, e_j) \in F^+$;

4) a **context**, composed of *relevant facts* $\mathcal{F}_{ij}^* \subset F^+$, defined by the facts that form the hypernym inference chain $\mathcal{I}_{ij}^k$ and distractors $\mathcal{D}_{ij}$, which are formed by triplets that do not form the hypernym inference in $\mathcal{I}_{ij}^k$.

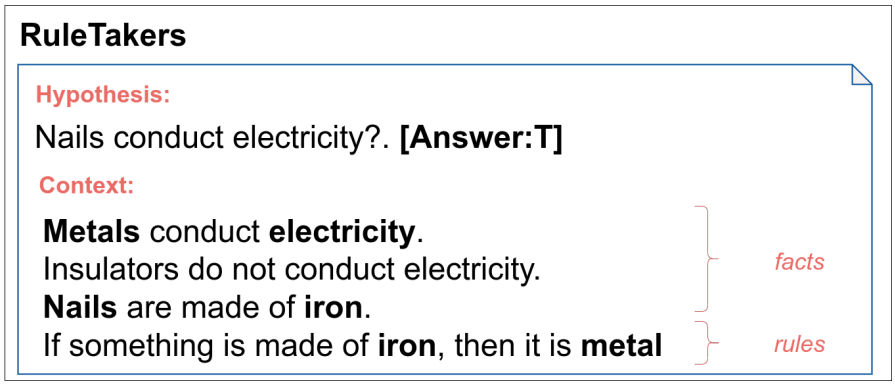


Figure 4.2: Sample from the RuleTakers dataset.

Given a *hypothesis* in *context* $< \mathcal{H}^k_{ij}, (\mathcal{F}^*_{ij}, \mathcal{D}_{ij}) >$, the task consists in inferring its truth value. The *hypothesis*, denoted by $\mathcal{H}^k_{ij}$, is considered to be *true* if it can be deducted following a *hypernym inference* $\mathcal{I}^k_{ij}$ from the *context* $(\mathcal{F}^*_{ij}, \mathcal{D}_{ij})$. Otherwise, the *hypothesis* is considered to be *false* under the close-world assumption.

In this task, we use the term "k-hop" to refer to a hypothesis where the number of elements in the hypernym inference path is equal to k.

To provide an example, let us consider the input from Figure 4.2 taken from the RuleTakers datasets. Here, the *hypothesis* is represented by the *triplet* (nails, conductivity, electricity) and is stated as "Nails conduct electricity". The task is to infer the factuality of this *hypothesis* considering the given *context*. In this case, the prediction is *true* because it can be deducted using the textual representations of the *triplets* (nails, made of, iron), (iron, type-of, metal), (metal, conduct, electricity), which are part of the given context. The textual representations of these *triplets* in Figure 4.2 are "Nails are made of iron", "If something is made of iron, then is metal", and "Metals conduct electricity", respectively. Therefore, this example is considered a 3-hop inference.

SINGLE-HOP REASONING MODELS As mentioned earlier, the approach to training models on the multi-hop question-answering task relies on teaching the models to solve k-hop reasoning, where the training set includes examples of different values of k-hop inferences. In this work, we refer to a *single-hop* model as a model that has only seen k-hop inference examples with a single value of k. In contrast, a *multi-hop* model is a model that saw i-hop inference examples with $1 \leqslant i \leqslant k$. SOTA models are typically trained on *multi-hop* datasets (Saxena, Tripathi, and Talukdar, 2020; Saha, Yadav, and Bansal, 2021).

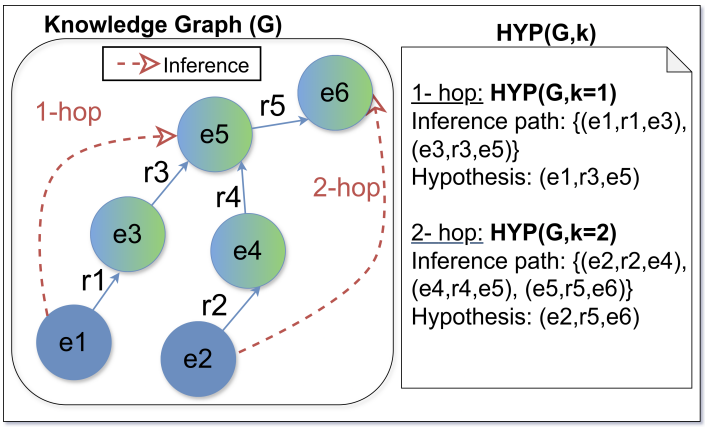


Figure 4.3: Hypothesis and inference path generation from a knowledge graph for 1-hop and 2-hop reasoning depths.

4.3.2 Probing Generation Algorithm

Given a knowledge graph G, we propose a generic dataset generation methodology to probe multi-hop reasoning PLMs in a *single-hop* setup. We define two generation functions to construct the input $< \mathcal{H}^k_{ij}, (\mathcal{F}^*_{ij}, \mathcal{D}_{ij}) >$:

i) $\textsc{Hyp}(G, k)$, to generate both the hypothesis $\mathcal{H}^k_{ij}$ and the related inference path $\mathcal{I}^k_{ij}$; and

ii) $\textsc{Distr}(G, \mathcal{I}^k_{ij}, \mathcal{H}^k_{ij})$, to generate a set of distractors $\mathcal{D}_{ij}$ with respect to the inference path $\mathcal{I}^k_{ij}$.

HYPOTHESIS GENERATION - $\textsc{Hyp}(G, k)$.

First, we apply the Depth First Search (DFS) algorithm to visit all entities of the knowledge graph G, generating a set of paths of length $k+1$, excluding the root, used as inference paths. Then, each inference path $\mathcal{I}^k_{ij}$ has the form of $< (e_i, r_0, e1), \ldots, (e_k, r_k, ej) >$.

For the true hypothesis, we create $\mathcal{H}^k_{ij}$ with the form (e_i, r_k, e_j) using the first and last facts from the inference path (see Figure 4.3, which illustrates examples of 1-hop and 2-hop hypothesis). Unlikely, for the false hypothesis, we simply generate a hypothesis $\mathcal{H}_{iz}$ replacing the last real fact of the inference path by $(e_k, r_k, e_z) \notin F^+$.

DISTRACTOR GENERATION - $\textsc{Distr}(G, \mathcal{I}^k_{ij}, \mathcal{H}^k_{ij})$.

The use of distractors has been given little or none attention in the literature. Mainly, because many of the works consider that the model must select the pertinent phrases from a context to construct

the inference. These approaches do not consider the influence of these distractors phrases in the context representation, and consequently, in the models performance on training and evaluation.

To our study, we propose three types of distractors that represent different levels of difficulty from a reasoning perspective, by creating additional information from different elements of the inference reasoning chain. These distractors are referred to as *object*, *relationship*, and *inference* distractors. We generate these distractors for each hypothesis $\mathcal{H}_{ij}^{k}$ (as shown in Figure 4.4).

- **Object distractor**. An *object distractor* is generated by sampling a fact with the form $(.,.,e_j)$, where e_j references the last entity in the hypothesis (or the last entity/object in the inference path).
- **Relationship distractor**. A *relationship distractor* is a sampled fact with the form $(.,r_k,.)$. Similarly, r_k references to the relationship used in the hypothesis.
- **Inference distractors**. Finally, we generate *inference distractors* in such a way that they exploit evidence from the structure of the inference path $\mathcal{I}_{ij}^{k}$ by linking two of its facts with a pivot element.

Having in mind the goal of guiding a k-hop model to perform incremental inference over single-hops, we propose distractor strategies that either improve the entity representations or bridge between entities by transferring information along with *intermediate hops* necessary to complete the reasoning. More precisely, based on a previous finding (Kassner and Schütze, 2020) showing that fine-tuned PLMs are good for recognizing false facts, we assume that distractors have a hidden impact on the reasoning task.

While most common approaches attempt to improve PLMs entity representations by enriching the context-based relevant facts, we believe distractors can significantly leverage PLMs entity representations and thus the reasoning performance. Thus, we investigate the rationale behind this assumption by designing the following strategies for generating inference distractors:

1. **Individual (i-inf)**, that uses different distractor entities ($x \neq y$) of two facts from the inference path. These distractors include the challenge of filtering irrelevant facts from the context, and to lead with negation forms. The implementation is shown in Algorithm 1, with the third parameter shared= `False`.
2. **Shared (s-inf)**, which uses the same distractor entity ($x = y$) of two facts from the inference path. This could be interpreted as a

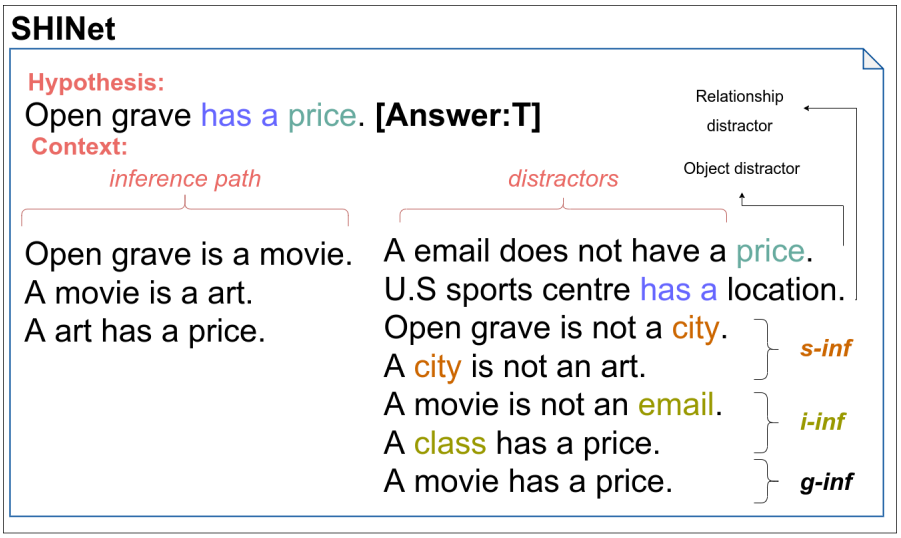


Figure 4.4: Sample from the SHINet 2-hop dataset.

harder type of distractor than (i-inf) where besides the negation form and relevance criteria, the distractors adopt a new branch of reasoning, which do not participate directly in the inference. The implementation is shown in Algorithm 2, with the third parameter shared= True.

3. **Guided (g-inf)**, Additionally, we explicitly guide a k-hop model to perform incremental inference using a *guided distractors* that connects the two consecutive facts in the inference path. The implementation is shown in Algorithm 3. The key underlying idea is to drive the PLM to *incrementally reason* over the inference path by transferring information between entities along *intermediate hops* of a multi-hop reasoning path.

Figure 4.4 illustrates the different distractors generated for a specific example.

Please note that other approaches, such as Chain-of-Thought, also explore the idea of guiding the reasoning by prompting (Wei et al., 2022b). These approaches aim to guide the reasoning of the PLM by extending the answer by adding textual justifications. However, unlike those approaches that provide a clean context input, we leave this filtering task to the models, simulating a more challenging setup. In real-life scenarios, understanding the reasoning behind a given text may not be straightforward and could have multiple paths.

Algorithm 1 Pseudocode for the distractors generation of individual (i-inf) distractors. The P-INF method is defined in Algorithm 4

I-INF ($\mathcal{I}_{ij}^{k}$,k)
D = P-INF($\mathcal{I}_{ij}^{k}$,k, False)
return D

Algorithm 2 Pseudocode for the distractors generation of shared (s-inf) distractors. The P-INF method is defined in Algorithm 4

S-INF ($\mathcal{I}_{ij}^{k}$,k)
D = P-INF($\mathcal{I}_{ij}^{k}$,k, True)
return D

Algorithm 3 Pseudocode for the distractors generation of guided (g-inf) distractors. The P-INF method is defined in Algorithm 4

G-INF ($\mathcal{I}_{ij}^{k}$,k)
D = P-INF($\mathcal{I}_{ij}^{k}$,k, True)
for $y \in 1 \dots k+1$ **do**
    $(e_x, \_, \_) = \mathcal{I}_{ij}^{k}[y]$
    $(\_, r, e_{x+2}) = \mathcal{I}_{ij}^{k}[y+1]$
    $D = D \cup \{(e_x, r, e_{x+2})\}$
end for
return D

Algorithm 4 Generic algorithm to create distractors for each two consecutive facts from the inference path. D, L1, and L2 are lists used to stock the generated distractors. AD stands for "available distractors", considering all the possible triples in the KG not used in the inference. AD(*e*) represents a filtered AD where *e* is present.

P-INF ($\mathcal{I}_{ij}^{k}$,k, shared)
$AD = \{\mathcal{E} \times \mathcal{R} \times \mathcal{E}\} \setminus F^{+}$
$D = \emptyset$
for $y \in 1 \ldots k+1$ **do**
  $(e_x, \_, \_) = \mathcal{I}_{ij}^{k}[y]$
  $(\_, \_, e_{x+2}) = \mathcal{I}_{ij}^{k}[y+1]$
  $L1 = \emptyset, L2 = \emptyset$
  for $(e_x, r_a, p) \sim AD(e_x)$ **do**
    $L1 = L1 \cup \{(e_x, r_y, b_j)\}$
  end for
  for $(q, r_b, e_{x+2}) \sim AD(e_{x+2})$ **do**
    $L2 = L2 \cup \{(e_y, r_j, e_{x+2})\}$
  end for
  for $(d_1, d_2) \sim L1 \times L2$ **do**
    $(e_x, r_a, x) = d_1$
    $(y, r_b, e_{x+2}) = d_2$
    if $x = y$ and shared **then**
      $D = D \cup \{d_1, d_2\}$
      break
    else if $x \neq y$ and not shared **then**
      $D = D \cup \{d_1, d_2\}$
      break
    end if
  end for
end for
return D

4.4 EXPERIMENTAL SETUP

In this section, we introduce the generated dataset probes and describe a bias reduction algorithm that is applied to them. We also provide details on our training strategy and implementation.

4.4.1 Dataset Probes Generation

We present here the dataset probes, namely Single-RuleTakers (S-RT) and SHINet, we automatically constructed using the previously presented generation functions (see Section 4.3.2).

These datasets are based on three different publicly available resources: the RuleTakers dataset (Clark, Tafjord, and Richardson, 2021), SHINRA (Sekine, Kobayashi, and Nakayama, 2018), a knowledge graph manually built upon a structured taxonomy, and ConceptNet (CN) (Speer, Chin, and Havasi, 2017), another KG widely used in NLP tasks (Talmor et al., 2020b; Ma et al., 2021). Please refer to the Figure 2.1 (Chapter 2) for samples on the SHINRA and ConceptNet dataset.

THE SINGLE RULETAKERS DATASET (S-RT)

In the original RuleTakers dataset, each entry is composed of a small theory, which represents the context ($\mathcal{F}^*_{ij}$), and a True/False question representing the hypothesis $\mathcal{H}^k_{ij}$. These datasets are grouped into five variations $k = 0$, and $D \leqslant k$ with $k = \{1, 2, 3, 5\}$ with questions requiring reasoning up to depths $0, 1, 2, 3, 5$ respectively.

As explained before, the RuleTakers dataset is part of the approaches that train models into a multi-hop setup, where different hops are mixed in the training set. Consequently, for our work we filtered these datasets to construct our probe. To this end, we construct single k-hop datasets with $k \in \{0, 1, 2, 3, 5\}$ for train and test splits, which we called *S-RT* dataset for easier reference. An example of a true hypothesis in the RuleTakers dataset is presented in Figure 4.2.

SHINET DATASET

We can observe from the example of the RuleTakers dataset (Figure 4.2), that there is no annotation provided in the context. This means that it is not straightforward to identify the relevant facts and/or the distractors in the context. This form of the dataset makes it difficult to measure their impact on the inference process. To overcome this limi-

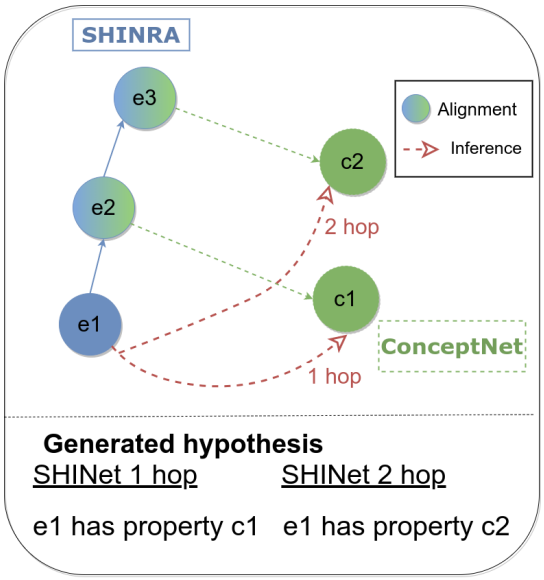


Figure 4.5: The SHINet dataset is created by aligning the SHINRA and ConceptNet datasets to generate a hypothesis. Nodes e_2 and e_3 exist in both datasets and link them.

tation, we created the SHINet dataset built upon the public SHINRA dataset. The SHINRA dataset contains facts with the form $(e_i, \text{is-a}, e_j)$, limited to the "is-a" relation.

Considering that the inference task heavily relies on the range of relationships and objects that the model has seen in the training phase (Wang et al., 2021a), we noticed that using only the SHINRA dataset is limited by the only relation present. Hence, we expanded the single relationship from SHINRA with the one present in ConceptNet, with each fact represented in the form (e_j, r', p_j).

We created *SHINet* dataset by sampling from SHINRA and ConceptNet based on a manual alignment of the intermediate nodes of SHINRA, done by a manual verification of finding e_j in both datasets, as represented in Figure 4.5. An example of a true hypothesis in the SHINet dataset with related distractors *s-inf*, *i-inf*, and *g-inf* is presented in Figure 4.4. Table 4.1 summarizes our generated datasets. We remark that we have train and test partitions for each SHINet strategy-based dataset.

4.4.2 Bias Reduction Algorithm

A limitation of building automatic generated datasets is the uncontrolled insertion of biases. As remarked in the work of Sakaguchi

Dataset	train	dev	test
1-hop *(s-inf)*	35,000	1,200	2,074
2-hop *(s-inf)*	35,000	1,200	5,994
2-hop *(i-inf)*	35,000	1,200	-
2-hop *(g-inf)*	35,000	1,200	-

Table 4.1: Number of samples for train/dev/test splits for each generated dataset.

et al. (2021) there exists dataset-specific biases which are not apparent for individual instances, they get introduced in the dataset where repetitive crafting strategies are used. Therefore, as recommended in the literature (Elazar et al., 2021; Sakaguchi et al., 2021), to avoid biases in our generated datasets that lead to an overestimation of the reasoning capabilities of PLMs, we applied on the test partitions the AFLITE algorithm (Sakaguchi et al., 2021).

THE AFLITE ALGORITHM. The AFLITE algorithm is an Adversarial Filtering (AF) algorithm initially proposed by Zellers et al. (2018) and later improved by Sakaguchi et al. (2021). This algorithm utilizes a dense representation of instances by employing their pre-computed neural network embeddings.

Let us assume we have a dataset consisting of 50k instances. To use the AFLITE algorithm, we perform the following steps:

1. We fine-tune RoBERTa (or any other PLM) on a small subset of the data, which includes 5k instances for training and 1k instances for validation.

2. After fine-tuning, RoBERTa pre-computes embeddings for the remaining instances (44k instances) and discards the initial 6k instances.

3. We proceed with using an ensemble of linear classifiers, specifically logistic regressions, which are trained on random subsets of the data. The goal is to assess whether the representation used in RoBERTa strongly indicates the correct answer option. If the representation is strong, we discard the corresponding instances and proceed with this process iteratively.

The algorithm requires pre-computed embeddings and labels as input, along with the ensemble size (n), training size (m) for classifiers in the ensemble, filtering cutoff size, and filtering threshold.

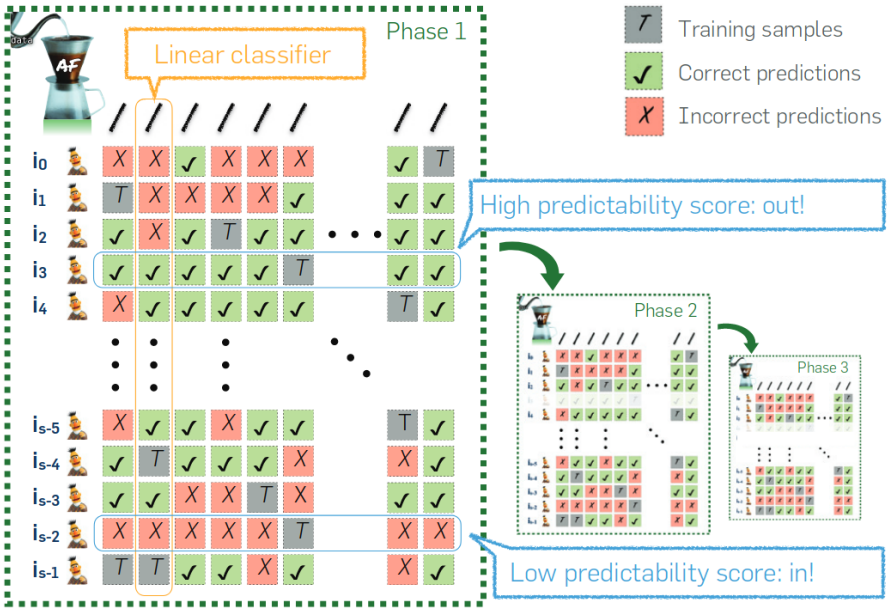


Figure 4.6: The AFLite algorithm. It uses a set of linear classifiers trained on different random partitions of the data. The algorithm filters out instances with highest scores, iteratively. Source Sakaguchi et al. (2021)

We train n linear classifiers during each filtering phase on different random data partitions. We then collect their predictions on their corresponding validation set. For each instance, we calculate its score as the ratio of correct predictions to the total number of predictions. We rank the instances based on their scores and remove the top-k instances whose score is above the threshold. We repeat this process until we remove fewer than k instances in a filtering phase or there are fewer than m remaining instances. Figure 4.6 illustrates the iterative process.

Our application used optimal parameters after grid-search with the final values of n (classifiers) = 64, m(samples) = 1000, top-k = 200, and threshold = 0.75. Comparing the original and filtered datasets, we filtered approximately 45% of the total samples.

4.4.3 Model Training Strategy

We used PLMs trained on the *single-hop* training partitions of our generated datasets. We remind the reader that this training protocol differs from the protocol used in previous approaches, where mixed datasets $\{0 \leqslant i\}$-hop datasets are simultaneously used for train-

ing multi-hop reasoning models based on a multitasking approach (Richardson and Sabharwal, 2020).

More specifically, we exploited the following:

1. We used several PLMs based on BERT (Devlin et al., 2019), XL-Net (Yang et al., 2019), and RoBERTA (Liu et al., 2019); even using similar architectures, they showed significant differences in performance results, especially on reasoning tasks (Talmor et al., 2020a);

2. As a building block, we used the training protocol proposed in previous work (Talmor et al., 2020b), removing the first fact from $\mathcal{I}^k_{ij}$ in 40% of the samples. Using this training protocol, we can provide insights into both the intrinsic strengths and limits of our proposed PLM since we exploit a recent state-of-the-art PLM that captures rich semantic information.

4.4.4 Implementation Details

SHINET CONSTRUCTION

Following previous work (Petroni et al., 2019), we transform each triplet fact element into a statement in natural language using linguistic patterns referred to as *fact templates*. For example, a triplet with the form (e_i, r, e_j) is textually represented as e_i *is a* e_j, when $r =$"is-a".

To generalize to all the relationships range available in our dataset, we aligned our templates according to previous work, particularly, the Hearst Patterns templates (Hearst, 1992; Roller, Kiela, and Nickel, 2018) and the ones proposed in Talmor et al. (2020b) as fact templates, considering that they also work in a ConceptNet resource (refer to Table 4.2). An example of the SHINet dataset is shown in Figure 4.4.

MODEL ARCHITECTURE

We train a transformer-based model with a set of input sequences of tokens with the following form: "[CLS] Context [SEP] Hypothesis [SEP]" (Figure 4.7). Where Hypothesis is the textual representation of the Hypothesis fact using the templates, and Context is the set of textual representations for each fact in the context set. Similarly, following previous approaches, we shuffled all the sentences in the Context, to avoid the PLM learning a naive sequence order inference.

Finally, we used the output representation of the [CLS] token and projected it into a binary classifier layer to obtain the probabilities that the hypothesis is true or false. For all of the models, we used the

Relation	Text template
HasProperty	"has the property"
UsedFor	"is used for"
RelatedTo	"is related to"
FormOf	"is form of"
Synonym	"is synonym of"
SimilarTo	"is similar to"
HasContext	"has contexto"

Table 4.2: ConceptNet relations and the text template used to create the phrases.

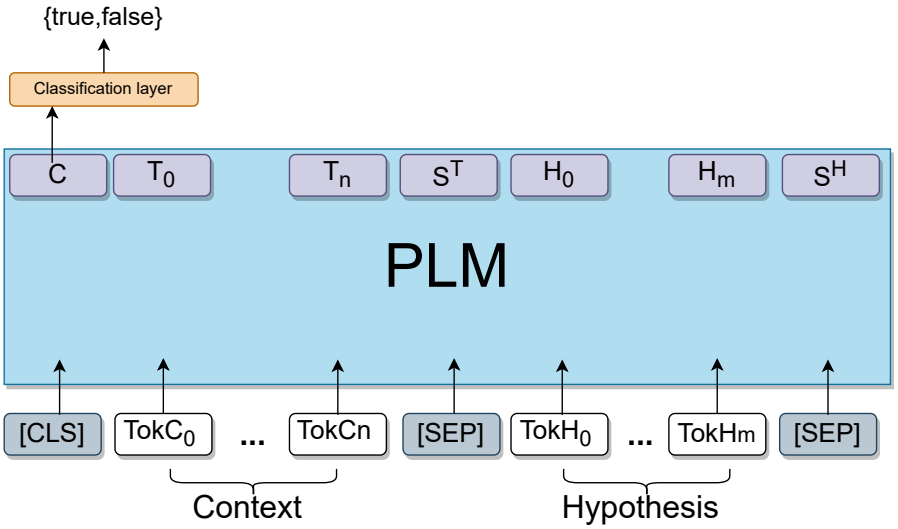


Figure 4.7: The model architecture used for training the reasoning models.

transformers' public implementation from HuggingFace (Wolf et al., 2020).

Main hyperparameters were set following standard setup or original authors' recommendations. In all the experiments where SHINet is used for training, we set a maximum word length to 256, batch size to 4, number of steps per batch to 729, number of epochs to 4, and Adam as optimizer with learning rate to 1e-5 and weight decay to 0.1. In the case of fine-tuning, each dataset uses its default hyperparameters. However, the parameters remain the same for each dataset regardless of the fine-tuning order.

For the QA tasks evaluation, we opted for the same parameters as when fine-tuning the SHINet datasets, except for the loss (categorical

Test (k-hop)	**k=0**	**k=1**	**k=2**	**k=3**	**k=5**
Train ↓ Models →	*RoBERTa*				
0-hop	**99.99**	*43.51*	*26.52*	*22.94*	*12.78*
1-hop	**90.11**	**98.16**	50.64	*37.30*	*23.07*
2-hop	66.92	64.62	**88.54**	**91.65**	**96.16**
3-hop	68.36	64.44	**88.47**	**91.35**	**96.11**
5-hop	63.32	63.11	**87.09**	**89.92**	**95.06**
$\leqslant$ **5-hop**	100.0	98.40	98.40	98.40	99.8

Table 4.3: Accuracy performance (in %) for a RoBERTa model trained on k-hop S-RT training set (rows) and tested on k-hop S-RT test set (columns). For a better reading, scores worse than random ($<$ 50%) are in *italic* and good results ($>$ 80%) are in **bold**.

cross entropy), and the number of steps per batch is set to 2,163. We trained and evaluated the models with 10 different random seeds and presented mean scores in our comparisons (see Appendix B for computational costs). To provide statistical significance to our results, we applied a test for Almost Stochastic Dominance (Dror, Shlomov, and Reichart, 2019) between test score distributions, using $\alpha = 0.05$.

4.5 RESULTS AND DISCUSSION

To address *RQ1* and *RQ2*, we perform our training and evaluations on the k-hop S-RT with $k \in \{0, 1, 2, 3, 5\}$ and the $\{D \leqslant 2\}$-hop, 1-hop, and 2-hop SHINet dataset. All the SHINet datasets are composed of the *object*, *relationship*, and *inference* distractors. Regarding the inference distractors, we evaluated with the three strategies: *i-inf*, *s-inf*, *g-inf*. In the rest of this work, we consider that the default setting uses the *(s-inf)* strategy unless it is explicitly mentioned otherwise.

To address *RQ3*, we used the MCQA (Richardson and Sabharwal, 2020), and RACE (Lai et al., 2017) datasets and tasks. MCQA is composed of 193,000 entries. Each entry is composed of a question and five possible answers, including reasoning tasks such as hypernymy, hyponymy, synonymy detection, and word sense disambiguation. RACE (Lai et al., 2017) consists of nearly 28,000 passages and 100,000 questions divided into Middle and High School sets and up to four possible answers.

Test (k-hop)	**k=1**	**k=2**	**k=1**	**k=2**	**k=1**	**k=2**
Train ↓ Models →	*XLNet*		*BERT*		*RoBERTa*	
Mixed	98.89	89.06	**99.23**	**94.63**	99.71	93.44
1-hop	**99.71**	86.00	98.96	89.75	**99.80**	96.23
2-hop	96.31	**99.82**	77.75	87.25	98.77	**99.99**

Table 4.4: Accuracy performances (in %) for mixed, 1-hop and 2-hop models using the SHINet dataset. The highest values are in **bold**.

4.5.1 Single-Hop PLMs and Incremental Reasoning

To answer *RQ1, Do Single-Hop Reasoning Models Incrementally Learn?*, we compare the performance of training models on different setups: single-hop and multi-hop. Following the previous work on the RuleTakers dataset, we used the RoBERTa model, and train separately single i-hop reasoning models using the S-RT dataset ($i \in \{0, 1, 2, 3, 5\}$). For $k = 4$, RuleTakers do not offer a training set, thus we did not add it for the test since it does not fit our analysis purposes. We keep the model, hyperparameters, and setting as proposed in Clark, Tafjord, and Richardson (2021). Similarly, we train different PLMs using the SHINet dataset on single 1-hop and 2-hop, and the *Mixed* models trained on the $\{k \leqslant 2\}$-hop SHINet dataset. In this case, we extend the PLMs including BERT, RoBERTa, and XLNET for further exploration on the impact of our strategy, optimizing the training hyperparameters for each model.

Table 4.3 and Table 4.4 report, respectively, the accuracy scores for the different single i-hop models using the S-RT dataset, and the accuracy scores of PLMs trained on 1-hop and 2-hop SHINet dataset, as well as the *Mixed* model.

ABOUT INCREMENTAL REASONING.

We take an empirical approach by assuming that incremental learning is observed when the models generalize from complex to less complex tasks. Overall, we can observe that when trained with larger hop depths, models struggle to solve even slightly less large reasoning tasks. Regarding specifically the *S-RT* dataset, it can be seen from Table 4.3 (green area), that the performance of the model trained and tested on the 2-hop (88.54) decreases to 66.92 and 64.62 respectively when tested on the 0-hop and 1-hop data. Similar behavior is observed for the model trained on 3-hop.

Looking at Table 4.4, obtained using the SHINet dataset, we can see that the results on the 2-hop test show a similar trend with the *S-RT* dataset: overall, the 2-hop PLMs exhibit better results when tested on 2-hop, but their performances decrease for a simpler task, 1-hop test, for all the three models while we expect at least stable performance. More precisely, we have observed a significant decrease in performance by a considerable margin. Specifically, the performance metrics have dropped by a significant amount. For instance, the performance metrics for XLNet have dropped from 99.82 to 96.31, which is a decrease of 3.51. Similarly, for BERT, the performance metrics have dropped from 87.25 to 77.75, which is a decrease of 9.5. Lastly, for RoBERTa, the performance metrics have dropped from 99.99 to 98.77, which is a decrease of 1.22. The same performance decrease trend is observed compared to the upper bound achieved when testing the 1-hop trained PLMs on the 1-hop test. This might reveal a counter-intuitive and uncontrollable behavior: having in mind the incremental human-style reasoning (Anderson, 1980), one could argue that the ability to solve a k-hop problem implies the ability to solve the {k-1}-hop one, but results indicate the contrary. These results are consistent in both datasets suggesting that *PLMs do not incrementally learn by accumulating knowledge*.

ON THE IMPACT OF IMPLICIT KNOWLEDGE IN PLMS.

In addition, looking at the compositionality generalization from simple to complex tasks, we can see from Table 4.3 (S-RT) that the performance of models trained on low-depth single-hops (rows $k = 0, 1$) significantly decreases when the hop is deeper (columns $k = 2, 3, 5$) in the test set (e.g., the 1-hop model performance decreases from 90.11 to 50.64 and 37.30 for columns $k = 2, 3$, respectively). However, for depth rows $k = 2, 3, 5$, this trend is less clear. Similarly, Table 4.4, using the SHINet dataset, shows that 1-hop models manage to obtain strong accuracy scores, over 86.00 in all datasets, indicating that they can deal with their tasks and complex ones. This behavior can be explained by the mix of implicit knowledge (from pre-training) and explicit knowledge (from training), filling the logic gap between tasks as shown by Talmor et al. (2020b). Moreover, we can observe that RoBERTa-based and XLNet-based PLMs are more effective in both 1-hop and 2-hop configurations, in contrast to BERT-based models, consistently with previous work (Talmor et al., 2020a).

4.5.2 Incremental Reasoning Strategies

To answer *RQ2. Can Reasoning Models be Guided Toward Incremental Reasoning?*, we compared our models to two different baselines: the *Mixed* model used in RQ1, and the *LoT* model from Talmor et al. (2020b). *LoT* is trained on a $\{k \leqslant 1\}$ using ConceptNet, WordNet, and Wikidata datasets to combine implicit knowledge acquired in pre-training with explicit rules and facts showing good performance on various types of reasoning tasks.

Additionally, we also trained the *hybrid* model by fine-tuning on the SHINet 2-hop (g-inf) dataset followed by the *LoT* train dataset (1-hop) to show the effect of jointly leveraging implicit knowledge, explicit knowledge (*LoT*), and incremental reasoning (*2-hop (g-inf)*).

We report our results computing the mean accuracy scores of the different distractors for each model in Table 4.5.

ON THE IMPACT OF DISTRACTORS.

At a first glance, we can see that our proposed guided model *2-hop (g-inf)* surpasses all its counterparts for 2 out of 3 settings, namely XLNet and RoBERTa. More precisely, by comparing the performance scores of *2-hop (g-inf)* to *2-hop (s-inf)*, we can fairly assess the positive impact of our proposed inference distractors presented in the Algorithm 3 to guide the training toward incremental learning across all the models. For instance, we observed significant improvements in the performance of *2-hop (g-inf)* model with *2-hop (s-inf)* models in the $k = 1$ test for XLNet, BERT, and RoBERTa. The improvements were 3.53, 6.41, and 1.17, respectively. Similarly, for the $k = 2$ test, we observed improvements of 0.17, 7.98, and 0.01. We further compare the accuracy scores of *2-hop (g-inf)* in comparison to *2-hop (i-inf)* to show the impactful role of the (s-inf) inference distractor to improve the reasoning inference. As it can be seen from Table 4.5, *2-hop (g-inf)* increases the performance on both tests by a difference greater than 21.0 in all models. The comparison between the *2-hop (g-inf)* model to a traditional multi-hop mixing strategy *Mixed* shows the advantage of the incremental inference for most of the settings.

Finally comparing our proposed guided model *2-hop (g-inf)* to the fine-tuned multi-hop strategy *hybrid*, we can observe that our guided model surpasses the hybrid model and *LoT* in 2 out of 3 models, namely (*XLNet* and *RoBERTa*). It is worth noting that this performance is achieved using fewer computational resources; the model can address both tests 2-hop and the simpler 1-hop in an incremental reasoning fashion.

In Figure 4.8, we show some hand-picked difficult examples from the *2-hop* test set for the *LoT* model that are especially helped by the guided model *2-hop (g-inf)*, using XLNet and RoBERTa. Specifically, we observed a positive impact of the distractors to solve false hypothesis examples using a negative phrase.

GENERALIZATION OF INCREMENTAL REASONING TO OTHER DATASETS.

Additionally, we compare our proposed models with those trained on the *S-RT* dataset used in *RQ1*. We particularly examine if the proposed models still exhibit the observed phenomenon highlighted in the green area from Table 4.3. To achieve this, we have conducted three runs and used the *inoculation* technique (see Section 2.3). The results with mean values are presented in Table 4.6.

We did a preliminary analysis of the learning curves for each task to determine the right amount of data to use. Figure 4.9 shows the learning curves when applying the inoculation technique for the MCQA and RC tasks. We selected 5,000 as the number of samples with the best performance and smaller training size. Similar analysis was done for the RACE and S-RT datasets with equal conclusion w.r.t. the number of samples.

We can see from Table 4.6 that even when the inoculation is used, the models relying on incremental reasoning (*2-hop (g-inf)* and *hybrid*) overpass the baseline results (*S-RT* and *2-hop (s-inf)*). Particularly, we see that guiding the model training over hops leads to improvements in lower hop levels ($k = 0, 1$) compared to traditional model training with mixed hops. For instance, for the test $k = 1$, the guided model *2-hop (g-inf)* improves by 0.56 and 0.36 the model trained with S-RT on hops $k = 2$ and $k = 3$, respectively.

4.5.3 Impact of Reasoning in QA tasks

To answer *RQ3. Do QA Tasks Leverage Incrementally Trained Reasoning Models?*, we used: 1) two QA tasks, namely Multiple Choice Question Answering (MCQA), and Reading Comprehension (RC). We applied the inoculation technique presented before to all the models; 2) the *Random* model, denoted *Rand*, the *RoBERTa* model, denoted *RoB*, and the *LoT* model as baselines. The *RoBERTa* model has been chosen, given its performance superiority as shown in the previous experiments (see Sections 4.5.1 and 4.5.2). To illustrate the tasks, Figure 4.11 shows an input sample for the MCQA task, while Figure

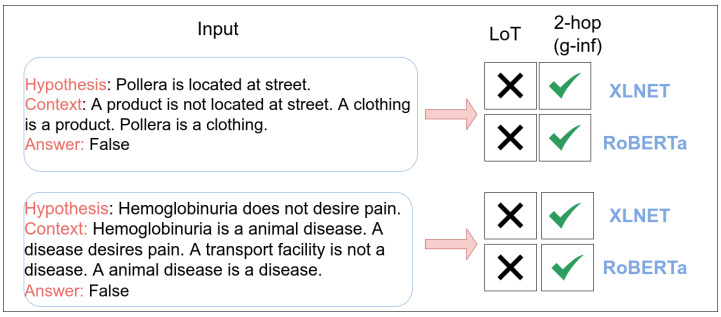


Figure 4.8: Two examples from the *2-hop* test set. Both examples are negative (false hypothesis), the first with a positive phrase and the second one with a negative phrase. The guided model correctly predicted both.

Test (k-hop)	**k=1**	**k=2**	**k=1**	**k=2**	**k=1**	**k=2**
Models →	*XLNet*		*BERT*		*RoBERTa*	
Train ↓						
LoT	98.37∗	99.17	86.28∗	**95.33**∗	99.15∗	98.96
Mixed	98.89	89.06	**99.23**	94.63	99.71	93.44
2-hop						
(i-inf)	60.81	56.77	62.37	57.16	60.48	56.23
(s-inf)	96.31	99.82†	77.75	87.25	98.77	99.99†
(g-inf)	**99.84**∗†	**99.99**∗†	84.16∗	95.23∗	**99.94**∗†	**100**∗†
hybrid	99.18∗†	99.67∗†	86.32∗†	93.85∗	99.31∗†	99.76†

Table 4.5: Accuracy performances (%) for 2-hop models by varying the inference distractor in the SHINet dataset. † and ∗ indicates statistical significance according to the Almost Stochastic Dominance test over *LoT* and *2-hop (s-inf)*, respectively.

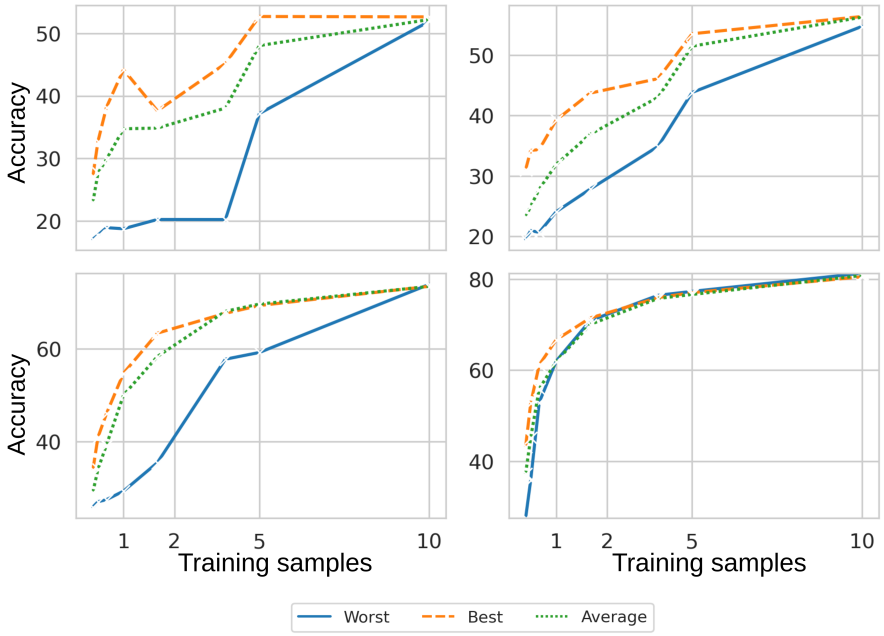


Figure 4.9: Learning curves for MCQA (upper) and RC (lower) tasks. For MCQA we show the Hypernymy (left) and Synonymy (right) dataset. For RC we show the Middle School (left) and High School (right) datasets. For all curves, the X axis represents the number of training samples (in thousands), and the Y axis, the accuracy score. Average values are reported with 5 runs for MCQA and 3 for RC.

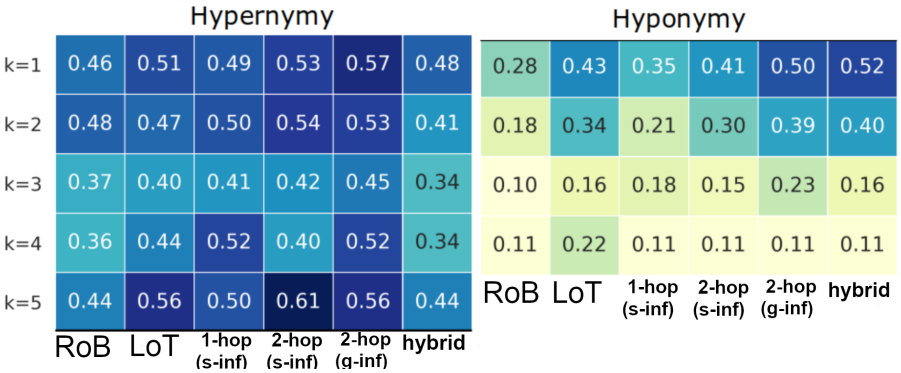


Figure 4.10: Accuracy values using the hypernymy and hyponymy subsets broken into number of hops k (rows) for the models (columns).

Train ↓ / Test(k-hop) →		k = 0	k = 1	k = 2
S-RT	2-hop	66.92	64.62	<u>88.54</u>
	3-hop	68.36	64.44	88.47
LoT	2-hop	<u>72.61</u>	65.37	88.40
	3-hop	72.46	64.81	88.56
2-hop	2-hop	67.15	64.88	88.42
(s-inf)	3-hop	67.06	64.58	88.25
2-hop	2-hop	68.06	65.18	88.05
(g-inf)	3-hop	67.89	64.8	88.11
hybrid	2-hop	71.75	<u>65.49</u>	88.44
	3-hop	**72.47**	**65.14**	**88.60**

Table 4.6: Accuracy comparing different reasoning models on the S-RT dataset. The best result for each test in <u>underlined</u> and **bold** for 2-hop and 3-hop models, respectively.

4.12 shows a sample for the RC task.

MULTIPLE CHOICE QUESTION ANSWERING (MCQA).

For MCQA, we re-used a publicly available code[1] as Richardson and Sabharwal (2020) and then applied the inoculation technique (Liu, Schwartz, and Smith, 2019). We plot the learning curves of each probe for the average of five different runs with random subsets (see Figure 4.9).

In Table 4.7, we report the results of our inoculated models. We can see that our models 2-*hop (g-inf)* and *hybrid* achieve the best average performance scores over all the baselines.

To deepen our analysis of the reasoning over increasing numbers of hops, we experimented with our models with the hypernymy and hyponymy subsets, up to 4 and 3 hops levels, respectively. By filtering the numbers of hops, we report the performance variation of our models in Figure 4.10. We can see that for the hypernymy (resp. hyponymy), the *hybrid* (followed by 2-*hop (g-inf)*) model outperforms all models in all depths but for $k = 4, 5$. Furthermore, we interestingly observe a positive trend toward reducing the performance decrease

1 https://github.com/yakazimir/semantic_fragments

MCQA - Hypernymy subset

Question:

"Given 'meat loaf', the text span or concept 'loaf' is best characterized as a kind or type of"

Choices:

a) yogurt (or yoghurt, yoghourt), defined as 'a custard-like food made from curdled milk'
b) solid, defined as 'matter that is solid at room temperature and pressure'
c) coconut (or coconut meat), defined as 'the edible white meat of a coconut; often shredded for use in e.g. cakes and curries'
d) meat, defined as 'the flesh of animals (including fishes and birds and snails) used as food'
e) produce (or garden truck, green goods), defined as 'fresh fruits and vegetable grown for the market'

Answer: b)

Figure 4.11: MCQA examples.

RACE - High school

Text:

"A nurse took the tired, anxious serviceman to the bedside. "Your son is here,"she said to the old man. .. the Marine interrupted her."Who was that man?"he asked. The nurse was surprised. "He was your father,"she answered. "No, he wasn't," the Marine replied. ...

Question:

Which of the following is NOT true according to the passage?

Choices:

a) The Marine didn't know the old man at all.
b) The nurse was careless and made a mistake.
c) The Marine happened to be the old man's son's friend.
d) The old man passed away peacefully and contentedly.

Answer: c)

Figure 4.12: RACE examples.

Model	Definition	Hypernymy	Hyponymy	Synonymy	DQA	Avg (%Imp.)
Random	19.90	19.90	20.20	19.80	20.00	19.96 (-38.6%)
RoBERTa	40.10	32.65	23.15	34.41	32.26	32.51 (-)
LoT	72.41	43.83	40.54	51.85	51.53	52.03 (+60.0%)
1-hop						
(s-inf)	62.65	52.69	38.58	45.80	43.68	48.68 (+49.7%)
2-hop						
(s-inf)	63.65	46.50	36.82	50.52	47.69	49.04 (+50.8%)
(g-inf)	70.86	**51.86**	43.03	**55.29**	52.65	**54.74** (+68.4%)
hybrid	**73.09**	44.70	**46.43**	51.87	**54.72**	*54.16* (+66.6%)

Table 4.7: Accuracy (%) scores for baselines and multi-hop reasoning models using the validation set of the MCQA dataset. Improvement percentages (%Imp) are given w.r.t. **RoB**ERTa (inoculated).

Models	**Middle (%Imp)**	**High (%Imp)**
RoBERTa	77.18 (-)	59.22 (-)
LoT	77.04 (-0.2%)	68.68 (+16.0%)
1-hop *(s-inf)*	76.56 (-0.8%)	68.94 (+16.0%)
2-hop *(s-inf)*	**77.32** (+0.2%)	**69.76** (+17.8%)
2-hop *(g-inf)*	75.65 (-2.0%)	68.72 (+16.0%)
hybrid	76.37 (-1.0%)	68.56 (+15.8%)

Table 4.8: Accuracy (%) comparing different reasoning models on the RACE dataset for middle school and high school. Improvement percentages (%Imp) are given w.r.t. RoBERTa.

rate between hop levels when using our proposed guided training approach. For instance, when comparing levels $k = 2$ and $k = 4$, we observe that performance decrease is reduced from 0.14 to 0.01 for *2-hop (s-inf)* and *2-hop (g-inf)* respectively. Similarly between the hyponymy levels $k = 2$ and $k = 3$ we can see a performance decrease reduced from 0.18 to 0.16 for *LoT* and *2-hop (g-inf)* models respectively. This observation clearly indicates the positive impact of incremental reasoning on performance.

READING COMPREHENSION (RC).

For RC, results under inoculation conditions are reported in Table 4.8. As can be seen, most of the models behave similarly for the Middle set, with *2-hop (s-inf)* as the most performing model. On the contrary, we can observe a clear improvement for all models on the High set when compared to *RoBERTa*. In this case, the most performing model is *2-hop (s-inf)* (69.76) closely followed by *2-hop (g-inf)* (68.72) and *hybrid* (68.56). Therefore, chains of reasoning seem to be a key component of the solution, even if most of the studied multi-hop models correctly capture the needed knowledge.

Finally, we compare the results from Table 4.7 and Table 4.8. We observe that model scores are very close on the RC task, even when using different distractors and the number of hops. The uniformity between all models' performances suggests that multi-hop reasoning is not a key component in solving these questions.

4.6 CONCLUSION

The use of PLMs for language reasoning tasks is becoming more popular, but the current training methods have produced inconsistent outcomes, which suggests certain limitations.

In this chapter, we have studied explicitly whether we can endow PLMs used in multi-hop reasoning tasks with the ability to incrementally acquire knowledge by following the inference path over the sequence of hops. Our goal is to better control the training of PLMs to create more understandable and predictable multi-hop reasoning models. To achieve this goal, we implemented single-hop training and evaluations using automatically constructed datasets, as presented in Section 4.4. Our findings complement previous research in the literature by showing that PLMs trained on 1-hop reasoning tasks can extrapolate the reasoning to 2-hop. However, 2-hop reasoning models struggle to generalize over slightly simpler 1-hop tasks.

Keeping in mind the human-style reasoning from simpler to complex tasks, we advocate incremental reasoning over the structure of the inference path as a step forward. In Section 4.3, we provided a training data generation strategy that relies critically on inference distractors connecting intermediate relevant facts in the reasoning path. By applying our approach, our models achieve higher or similar performance trends than fine-tuning multi-hop models but consume fewer resources. Furthermore, we show that the incrementally trained multi-hop PLMs are transferable to other QA-based tasks (Section 4.5).

Although our experimental settings are limited to low depths of inference ($k = 1, 2$), our findings demonstrate the feasibility and benefits of incremental reasoning and open up new research opportunities.

In this chapter, we explored implicit integration strategies of KG into PLMs for reasoning-based tasks. In the next chapter, we will explore explicit integration methods.

5

PROBING PRE-TRAINED LANGUAGE MODELS WITH HIERARCHY PROPERTIES

5.1 INTRODUCTION

From the previous chapter, we studied the implicit injection of knowledge from KG into PLMs for reasoning-based tasks. This implicit injection relies on transforming elements from the KG into plain text using text templates. We proposed and tested different strategies for adding information beyond the main reasoning chain and evaluated how this affects the PLMs performance.

Complementary, explicit injection methodologies are particularly useful to represent structural features of the KG, making them a better fit to represent their hierarchical structure. Learning knowledge representations with an underlying hierarchical structure is an active research topic (Rossi et al., 2021). Approaches include learning embedding representations from symbolic knowledge sources (e.g., knowledge graphs (KGs)) such as TransE (Bordes et al., 2013) and TransR (Ji et al., 2015) or constructing dedicated vector spaces based on hyperbolic spaces (Nickel and Kiela, 2017) and order embeddings (Vendrov et al., 2015)(Section 3.2.2). Particularly, for PLMs, learning underlying hierarchical structure is achieved by injecting triplets embeddings to capture entity and relation features (Liu et al., 2020; Zhang et al., 2019), as introduced in Chapter 3.

The learning of hierarchical structure in different models is particularly beneficial to NLP downstream tasks such as Taxonomy Reconstruction and Hypernymy Detection (Nickel and Kiela, 2017; Camacho-Collados, 2017), which are commonly used as evaluations. Consequently, recent work studied PLMs' adaptation to Taxonomy Reconstruction (Jain and Anke, 2022) and Hypernymy Detection tasks via sequence classification (Chen, Lin, and Klein, 2021). However, these evaluations rely on superficial hierarchical features (Camacho-Collados, 2017).

In this chapter, we delve into the development of an extensive evaluation of hierarchical knowledge encoded by PLMs. We aim to focus on the structural hierarchical features from taxonomies and analyze whether these features help enhance the representations. To conduct this evaluation, we propose to use the task diagnostic approach discussed in Chapter 2 and characterize the hierarchical structure using

hierarchy properties (presented in Section 5.3). Our proposition aims to create a task-agnostic setup suitable for PLMs that can measure various aspects of the hierarchy beyond the surface-level features, such as hypernymy relationships.

The hierarchy properties used to characterize the hierarchical structure rely on structural edge-based observations from the taxonomies. These properties are represented as the interaction between three elements in the taxonomy, which we call *ternaries*. These ternaries are later used as probes for language models to evaluate the notion of hierarchy in PLMs through these properties. We then explore the injection of these hierarchy properties to observe whether PLMs can learn them independently and integrate hierarchy. Finally, we evaluate the impact of learning these properties on improving the overall notion of hierarchy by testing PLMs on different hierarchy-related tasks such as taxonomy reconstruction and hypernymy discovery.

In the remaining part of this chapter, we have structured our discussion in the following way. First, we will overview the problem and our main contributions in Section 5.2. After that, we will explain our methodology, including the hierarchy properties and the design of our probes in Section 5.3. In Section 5.4, we will define our experimental setup along with datasets, metrics, and baselines. We will present and discuss our results in the subsequent section, i.e., Section 5.5. Finally, we show our conclusions in Section 5.6.

5.2 OVERVIEW AND CONTRIBUTIONS

The hierarchical representation of concepts is a fundamental aspect of human cognition and essential in performing numerous Information Retrieval (IR) (e.g., web search (Sieg et al., 2004)) and Natural Language Processing (NLP) tasks (e.g., hypernym discovery (Camacho-Collados et al., 2018)).

Therefore, works in this research direction have explored incorporating hierarchical knowledge extracted from knowledge graphs and taxonomies to refine models (Hu et al., 2015; Liu et al., 2020; Nickel and Kiela, 2017; Chen, Lin, and Klein, 2021). In particular, prior studies on static word embeddings showed that designing dedicated space representations to encode the hierarchy benefits the performance of various downstream tasks (Nickel and Kiela, 2017; Vendrov et al., 2015), giving rise to the importance of encoding concept hierarchy in these representations.

Most previous studies have adopted task-dependent evaluations to assess the extent to which models capture hierarchical linguistic knowledge. These evaluations are based on the premise that the

model's performance on downstream tasks sheds light on whether the model captures hierarchy. Specifically, such evaluations target tasks that require applying hierarchical knowledge to some degree, such as taxonomy reconstruction and hypernymy detection (Nickel and Kiela, 2017; Camacho-Collados, 2017).

However, existing task-dependent evaluations have two main limitations. First, they heavily rely on detecting a single relation type (hypernymy) and overlook the implicit taxonomy structure by ignoring other related relations, such as ancestors and siblings (Mao et al., 2018). As a result, none of the state-of-the-art (SOTA) evaluation methodologies are able to reveal this implicit yet essential hierarchical information (Alsuhaibani, Maehara, and Bollegala, 2018).

Second, particularly in the context of PLMs, task-dependent evaluations might conflate the model's understanding of a given target task and the model's understanding of hierarchy per se.

Hence, there is a need for more comprehensive evaluation methods and datasets that consider complex hierarchical relations and are able to reveal how well models capture those relations and apply them to downstream tasks.

The present contribution addresses the two aforementioned limitations. We propose a task-agnostic methodology to evaluate language models' understanding of hierarchical knowledge considering implicit and more complex taxonomic relations beyond the direct hypernymy (parent, ancestors, and siblings). In particular, our evaluation focuses on PLMs, such as BERT (Devlin et al., 2019), considering their ability to encode lexical knowledge, as well as their outstanding performance on different tasks (Qiu et al., 2020).

To the best of our knowledge, no task-agnostic methodology exists so far that enables us to reveal to what extent PLMs encode hierarchy relations intrinsically. We rely on the task diagnostic methodology (Chapter 2) to characterize the task into a set of properties, and we use a *probe*-based methodology (Pimentel et al., 2020) to evaluate whether SOTA models capture task-agnostic linguistic knowledge.

Our contributions seek to answer three main questions:

1. *RQ1*: *To which extent do PLM representations encode hierarchy w.r.t. hierarchy properties?*

2. *RQ2*: *Does injecting hierarchy properties into PLMs using a task-agnostic methodology impact their representations?*

3. *RQ3*:*Can hierarchy-enhanced PLM representations be transferred to downstream tasks?*

5.3 METHODOLOGY

This section presents our task-agnostic methodology for probing hierarchical knowledge of PLMs.

First, we define a set of intrinsic properties for a hierarchy (Section 5.3.1). Then, we describe how to design probes upon these properties and how to fine-tune PLMs to identify these properties by learning these probes (Section 5.3.2).

5.3.1 Intrinsic Hierarchy Properties

We characterize a hierarchical structure with a set of intrinsic properties that mirror the distribution of concepts within a given taxonomy. We rely on evaluating the taxonomic similarity to study the hierarchical relationship between words. From a linguistic perspective, this can be understood as evaluating the hierarchical relation based on the *paradigmatic* knowledge encoded by PLMs (Lapesa, Evert, and Walde, 2014; Kacmajor and Kelleher, 2020). To validate this similarity between two concepts, we represent concepts as nodes and use edge-based approaches, such as the shortest path between two nodes (Zhong et al., 2002; Budanitsky and Hirst, 2006). Please note that this differs from a *syntagmatic* approach, where only co-occurrence is considered.

Concretely, we define a set of *relations* and *properties* from edge-based observations. Then, we evaluate these properties empirically considering semantic distance methods, inspired by (Budanitsky and Hirst, 2006). We consider distance as the complement metric for similarity.

Notably, for a fixed node n, we define four basic relations –*parent*, *ancestor*, *sibling*, and *far relative*– in a taxonomy in the following way. A *parent* node p is a direct hypernym at a one-edge distance from n, while an *ancestor* node is an indirect hypernym at a two-edge distance. A *sibling* node shares a parent with a two-edge distance, and a *far relative* shares the ancestor but not the parent.

We use these relations and the corresponding edge-based distances to define hierarchy properties. Table 5.1 presents the six hierarchy properties considered in this work based on the four defined relations for all possible combinations. In order to verify these properties, we formulate them as inequalities that examine the distance between two distinct relations for a fixed node, as shown in Table 5.2. Our motivation for considering three nodes (*ternary*) in these evaluations comes from recent research on pattern-based relation extraction (Hovy, Kozareva, and Riloff, 2009; Liu, Cohn, and Frermann,

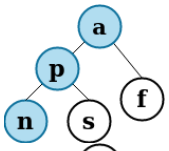
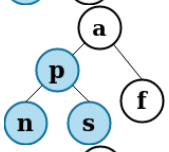
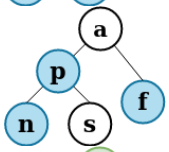
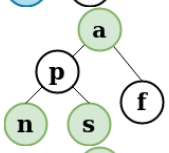
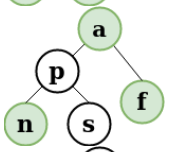
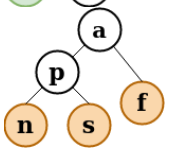
Table 5.1: The six hierarchy properties with their names and textual descriptions.

Property	Description
P-A	*"A node is closer to its parent than its ancestor."* The similarity between a pair of concepts is proportional to their path length under an edge-based approach (Resnik, 1995).
P-S	*"The distance between 'siblings' should be longer than between 'father' and 'son'."*. Similarly, it is a straightforward application of edge-counting, also found in other approaches such as scaling the taxonomy (Zhong et al., 2002).
P-F	*"The parent is the closest element to a node."* It generalizes P-S by comparing to further relations in a taxonomy. If a model does not satisfy this property, it struggles to differentiate the hypernym relation from others.
A-S	*"A node is closer to its ancestor than to its sibling."* We did not find an expected behavior about this property in the literature. We empirically choose the path-level evaluation proposed by (Gupta et al., 2016), favoring the correct edges in the hypernym path rather than adding incorrect elements that could cause a cascade of generalization errors.
A-F	*"The ancestor is the closest term for a node, except for the father."* This property is a generalization of the ancestor relationship evaluated in further relationships in a taxonomy.
S-F	*"The sibling is the closest element for a node, except for the father and the ancestor."* Based on edge-counting approaches, we should find a sibling node closer to other relations beyond the ancestor in a hierarchy.

2023), where the inclusion of a third "anchor" node has proven useful in capturing various relation types, including hypernymy and cohyponymy.

To facilitate the reading, we adopt a naming convention for the properties that highlights the two used relations, i.e. a **R**elation-**R**elation format. The relation identifier, denoted by **R**, can take one

Table 5.2: The hierarchy properties along with their distance-based definitions, and the three participant (colored) nodes in the taxonomy. Three *groups* are identified based on the left relation in the inequality: P-* (regrouping P-A, P-S, and P-F), A-* (regrouping A-S and A-F), and S-* (with only S-F).

Property	Definition	
P-A	$\text{dist}(n, p) < \text{dist}(n, a)$	
P-S	$\text{dist}(n, p) < \text{dist}(n, s)$	
P-F	$\text{dist}(n, p) < \text{dist}(n, f)$	
A-S	$\text{dist}(n, a) < \text{dist}(n, s)$	
A-F	$\text{dist}(n, a) < \text{dist}(n, f)$	
S-F	$\text{dist}(n, s) < \text{dist}(n, f)$	

of four possible values: **P**arent, **A**ncestor, **S**ibling, and **F**ar relative. Furthermore, we congregate the initial properties into three *groups*, P-*, A-*, and S-*, based on the left relation in the inequality. These *groups* serve as an aggregated representation of the properties for each type of relation.

5.3.2 Probing Hierarchical Representations

DESIGNING PROBES.

For this purpose, we align the properties into a single form. Given a taxonomy $T = (V, E)$, with V representing a set of nodes and E a set of edges, we define a *ternary* as $t = (x_n, x_l, x_r)$ encoding a hierarchy property $\mathbf{R}_l$-$\mathbf{R}_r$ where x_n represents a fixed node, x_l the node of the **l**eft **R**elation ($\mathbf{R}_l$), and x_r the node of the **r**ight **R**elation ($\mathbf{R}_r$). Note that each of the tuples (x_n, x_l) and (x_n, x_r) corresponds to one of the defined relations (parent **P**, ancestor **A**, sibling **S** or far relative **F**) and the taxonomic distances (edge-based) must satisfy the associated property $\mathrm{dist}(x_n, x_l) < \mathrm{dist}(x_n, x_r)$.

Each node within the ternary tuple is converted into textual representations through predefined phrases (prompts) (Section 5.4). These concept representations are used to compute a model representation within a given language model. Consequently, for a given model, we obtain a representation in the form $(\hat{x_n}, \hat{x_l}, \hat{x_r})$ for each ternary (x_n, x_l, x_r). Finally, we use a distance method to evaluate the inequality $d(\hat{x_n}, \hat{x_l}) < d(\hat{x_n}, \hat{x_r})$.

TEACHING HIERARCHY PROPERTIES TO A PLM.

To teach the hierarchy properties to PLMs, we leverage the concept representations derived from the probes, $(\hat{x_n}, \hat{x_l}, \hat{x_r})$, and train a model to satisfy the inequality $\mathrm{dist}(\hat{x_n}, \hat{x_l}) < \mathrm{dist}(\hat{x_n}, \hat{x_r})$. We assume that a model's performance on these probes indicates how well its representations align to a hierarchy-like distribution.

We employ the Sentence Transformer framework (Reimers and Gurevych, 2019) with a triplet network architecture and a pooling operation to the output of the PLM to generate the embedding of our concepts. This approach achieved promising results on lexical knowledge evaluations (Vulić et al., 2021).

In order to evaluate and teach our approaches, we use a pipeline illustrated in Figure 5.1. To teach a PLM the hierarchy property $P - S$, we take one triplet such as (tablespoon, spoon, wooden spoon). We convert these entities into text and obtain the PLM representations noted by the T, S, and W pointed boxes in the "Teaching a Property" module from Figure 5.1. We then train the inequality $\mathrm{dist}(\text{tablespoon}, \text{spoon}) < \mathrm{dist}(\text{tablespoon}, \text{wooden spoon})$ using the T, S, and W representations, where dist is the cosine distance.

Given the inner contrastive nature of our probes in a triplet form, we adopt a contrastive loss, specifically the Triplet loss (Dong and Shen, 2018). This loss fine-tunes the network to minimize the distance between related inputs, $(\hat{x_n}, \hat{x_l})$, while maximizing the distance for unrelated inputs, $(\hat{x_n}, \hat{x_r})$, by minimizing the following loss:

Figure 5.1: Pipeline for the evaluation and teaching of hierarchy properties. First, a property ternary is sampled from the dataset. Then, each ternary element is converted into text adapted to the PLM using prompts. For evaluation, we compute the embedding representation for each input sentence and compare their distances. For teaching, we use a triplet network to compute sentence representations and then apply a triplet loss.

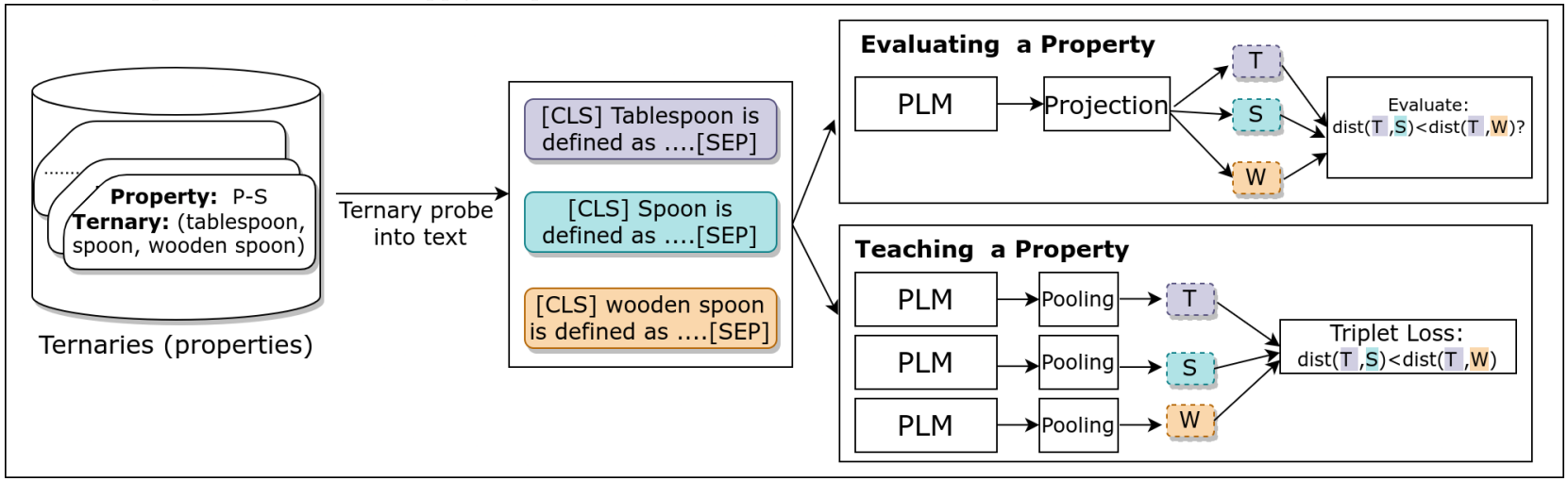


Table 5.3: Number of samples (ternaries) generated per property.

# ternaries	P-A	P-S	P-F	A-S	A-F	S-F
train	6,397	16,792	20,710	-	20,710	31,032
dev	1,443	2,016	4,352	2,016	4,352	4,112
test	1,435	7,658	14,795	7,658	14,795	64,906

$$\mathcal{L}(x_n, x_l, x_r) = -max(\|\hat{x_n} - \hat{x_l}\| - \|\hat{x_n} - \hat{x_r}\| + \alpha, 0) \tag{5.1}$$

5.4 EXPERIMENTAL SETUP

5.4.1 Datasets and Metrics

We used the Bansal et al. (2014) dataset to sample our probes, consisting of medium-sized taxonomies generated from WordNet subtrees (Bansal et al., 2014). This dataset comprises subtrees of height 3 (i.e., the longest path from the root to the leaf is 4 nodes) containing between 10 and 50 terms. Specifically, we used the version extended with WordNet definitions (Chen, Lin, and Klein, 2021).

We generated ternaries for each property, split into train/dev partitions to fine-tune PLMs with the hierarchy properties. Table 5.3 shows the number of ternaries generated per property for each split set. As mentioned before, we do not consider the property *A-S* for training due to its absence in the literature. For evaluation, we use the accuracy metric defined as the number of correct predictions where the ternary inequality was satisfied, divided by the total number of ternaries in the test set.

5.4.2 Baselines and Models

As baselines, we use three groups of models:

Group 1. Comprises random and non-contextualized models as a lower bound performance of the PLMs following previous work (Vulić et al., 2021; Talmor et al., 2020a):

- a) **Random:** we generated symmetrical random distances for all node pairs and report the average of ten random runs;
- b) **FastText:** to compare static word embeddings in a fair setup, i.e. avoiding the out-of-vocabulary problem, we used the

character-based version of **FastText** embeddings (Bojanowski et al., 2017) trained on Wikipedia (**FT-wiki**)

Group 2. Comprises a total of five PLMs available in the HuggingFace library:

- a) **BERT** (*bert-base-cased*);
- b) **BERT-L** (*bert-large-cased*);
- c) **RoB-L** (*roberta-large*);
- d) **S-RoB** (*all-distilroberta-v1*); and
- e) **S-MPNet** (*all-mpnet-base-v2*).

We notice that the first three models are cross-encoders and the last two models (**S-RoB** and **S-MPNet**) are bi-encoders. The bi-encoders are trained using a dual-encoder network and oriented towards semantic textual similarity tasks on multiple datasets. These characteristics give some advantages in the final results of our probes compared to classical PLMs. However, we evaluated these models to obtain insights on the most robust ones.

Group 3. Comprises two SOTA knowledge enhanced PLMs:

- **ERNIE** (Sun et al., 2019), a BERT-based trained on multiple tasks to capture lexical, syntactic and semantic aspects of information, and
- **CTP** (Chen, Lin, and Klein, 2021), a RoBERTa large model trained to perform hypernym prediction.

5.4.3 Concept and Ternary Representations

Each concept in a ternary is represented textually by its name and definition from a knowledge source to provide context information. We consider two vector-based methods, namely **cls** and **avg**, and prompt-based methods to generate the representation of a ternary to text with a PLM.

For vector-based methods, we use the last layer output of the PLM and represent each concept as a list of tokens with the form: *'[CLS] [concept name] is defined as [definition] [SEP]'*. The **cls** method uses the special token *[CLS]* and the **avg** method computes the average of all subtokens. We compute the distance between these representations based on cosinus (**cos**) and euclidean (**L2**) distances.

In contrast to vector-based methods, the prompt-based method condenses the ternary representation into a single textual representation.

Table 5.4: Accuracy scores for representation methods on different models for Property *P-A*, other properties follow similar trends. *cos* and *L2* stand for cosine and euclidean distances respectively. *cls* and *avg* refer to the type of representations used. LMScorer is not computed for bi-encoder models since they are not adapted for this task (Wang, Reimers, and Gurevych, 2021).

Model Repr.	**Distance Method**	BERT	BERT-L	RoB-L	S-RoB	S-MPNet
cls	cos	72.6	68.2	58.5	83.1	83.2
	L2	72.4	68.7	58.5	79.1	83.1
avg	cos	**75.8**	**78.8**	**73.2**	**85.2**	**85.6**
	L2	74.3	77.1	61.5	83.3	**84.9**
LMScorer	-	50.2	52.6	54.6	-	-

We use the LMScorer method (Jain and Anke, 2022) to score sentences for factual accuracy.

LMScorer computes a pseudo-likelihood score for each token in a sequence by iteratively masking it, considering past and future tokens. We recall its formulation (presented in Chapter 2):

$$\text{LMScorer}_{\text{PLM}}(\mathbf{W}) = \exp\left(\sum_{i=1}^{|\mathbf{W}|} \log P_{\text{PLM}}(w_i|\mathbf{W}_{\setminus i})\right) \tag{5.2}$$

We experiment with different templates for converting these ternaries and report the best-scoring template: "A is a B. C is a B. [definition a][definition b] [definition c].", where A, B, and C are nodes; different from the one for vector-based methods. For static word embeddings, we use only the **avg** method, as there is no *[CLS]* token.

5.5 RESULTS AND DISCUSSION

In this section, we report the results to answer our research questions. To simplify our analysis, we use the *groups* presented in Table 5.2 and report the average values. We only report the best configuration to facilitate the reading in our figures and tables, but we ensure we obtain the best configuration for each model.

5.5.1 Evaluation of Hierarchy Properties in PLMs

DEFINING THE BEST REPRESENTATION METHOD.

First, we carry out preliminary experiments to explore the best performing settings in terms of model representation and distance methods. Table 5.4 shows our results based only on the representative property *P-A* using various methods on all PLMs. Our findings indicate that the vector-based representations (75.0-85.6) outperform all the explored prompts. We also observe that the **avg** representation is always superior to **cls**. Moreover, the **avg** representation with **cos** distance frequently obtains slightly better scores (76.0) than **L2** (75.7). Under this evidence, we adopted the **avg** representation with **cos** distance for our remaining evaluations (Appendix C shows results obtained with **avg** representation and **L2** distance).

We now answer our first research question **RQ1:** *To which extent do PLM representations encode hierarchy w.r.t. hierarchy properties?*

COMPARING MODEL'S OVERALL PERFORMANCE.

Our analysis focuses on the model's overall performance in our evaluation (named *All*, as the average score from all properties) and provides insights into the property *groups* (*P-**, *S-**, and *F-**) captured by *Group 1* and *Group 2* models.

Considering the model's overall performance (*All*), we first test the quality of our probes from our baselines in *Group 1*. Table 5.5 shows that **FastText (FT-wiki)** embeddings perform better (60.3) than the **Random** approach (50.2). Aligned with previous work (Fu et al., 2014) claiming that static embeddings encode hypernymy-like relations to some degree, we argue that our probes help to capture these hierarchical relations.

Similarly, all *Group 2* models obtained *All* scores higher than **Random**. In particular, the **S-RoB** model obtained the highest score (70.3), closely followed by **S-MPNet** (70.0). Moreover, only these models outperformed the **FT-wiki** static embeddings.

ANALYZING KG-ENHANCED PLM'S PERFORMANCE.

From *Group 3* models, we can surprisingly see that **ERNIE** outperforms **CPT**, while the latter is a RoBERTa large model specifically trained on hypernym classification task with a classification layer on top of the PLM.

By comparing these models against their vanilla version, we can interestingly observe that while **ERNIE** showed constant improvement in all properties, with +2 score points on *All* (56.2) w.r.t. **BERT**, our probes suggest that **CTP**, trained on a task-dependent evaluation (37.7), degrades its initial representations w.r.t. **RoB-L** (53.3).

Moreover, **CTP** is the only PLM-based model with lower performance than **Random**. Overall, these results provide insights on the conflation problem faced with task-dependent evaluation regarding task understanding and hierarchy understanding of PLM's.

ANALYZING CHALLENGING TAXONOMIC RELATIONS.

Now, we analyze the results at *groups* level. Considering the semantic similarity approach in pre-training, we expect that *sibling* relations (*S*-*), akin to analogies, will exhibit better representations than other hierarchical relations.

Across all models, we can observe that the *siblings* relation (*S*-*), followed by the *parent* relation (*P*-*), obtained higher performances than the *ancestor* relation (*A*-*). Our results capture the preference of the semantic similarity of these models, particularly when compared to other *far* relations, because of the high values in *S-F* (min: $73.8 - \mathrm{max}$: 80.6).

Similarly, most *ancestor* representations are further than *parents* ($73.2 - 85.6$ in *P-A*). Based on property *P-S* ($37.2 - 42.0$), PLMs such as **BERT**, **BERT-L**, and **RoB-L** tend to represent *siblings* closer ($73.8 - 76.1$) than the *parent* ($60.5 - 66.1$). For the *ancestor* relation, higher scores on property *A-F* ($40.1 - 65.9$) than on *A-S* ($22.3 - 33.7$) imply that *far relatives* are easier than *siblings*. However, these scores are generally lower than those obtained with other properties. Besides, when comparing the *P-F* and *A-F* columns, we observe higher scores with the *parent*, one-edge distance, than the *ancestor*, two edges away (similarly, for the scores between *P-S* and *A-S*). These trends indicate that *ancestor* representations are not as clearly defined as *parents*.

Overall, our evaluation reveals that the analyzed PLMs struggle to capture some hierarchical relationships such as *sibling* and *ancestor*. Besides, the results confirm that the performance of PLM specifically trained on a hierarchy-aware task (e.g., **CTP**) is not a salient signal of PLM's understanding of hierarchy.

Table 5.5: Accuracy on each hierarchy property on the test dataset with method *avg(cos)*. Best results in **bold** for each probe per group. * shows 1 sample t-test > 0.05.

Model	**P-A**	**P-S**	**P-F**	**A-S**	**A-F**	**S-F**	**P-***	**A-***	**S-***	**All**
					Group 1: Baselines					
Random	50.5	49.8	50.1	50.1	50.4	50.1	50.1	50.3	50.1	50.2
FT-wiki	79.7	49.9	81.6	24.1	48.5	78.2	**70.4**	36.3	**78.2**	**60.3**
					Group 2: Vanilla PLMs					
BERT	75.8	37.9	71.8	22.3	43.0	74.5	61.8	32.6	74.5	54.2
BERT-L	78.8	42.0	77.6	24.6	47.6	76.1	66.1	36.1	76.1	57.8
RoB-L	73.2	37.2	71.1	24.6	40.1	73.8	60.5	32.4	73.8	53.3
S-RoB	85.2	68.6	90.4	33.7	64.8	78.8	**81.4**	**49.2**	78.8	**70.3**
S-MPNet	85.6	70.0	88.3	29.6	65.9	80.6	81.3	47.8	**80.6**	70.0
					Group 3: KG-Enhanced PLMs					
ERNIE	77.0	42.7	74.2	23.1	44.9	75.6	**64.6**	34.0	**75.6**	**56.2**
CTP	59.9	22.4	36.2	23.9	22.8	61.0	39.5	23.4	61.0	37.7
					Group 4: Hierarchy-Enhanced PLMs					
BERT_{hp}	79.9	56.2	82.6	29.2	58.4	75.2	72.9	43.8	75.2	64.0*
$\text{BERT-L}_{\text{hp}}$	85.7	69.1	91.0	31.6	67.6	79.3	81.9	49.6	79.3	70.3*
RoB-L_{hp}	82.0	64.3	86.0	35.1	61.9	76.1	77.4	48.5	76.1	67.4*
S-RoB_{hp}	87.7	74.9	93.9	34.5	73.0	81.6	85.5	53.8	**81.6**	**73.6***
$\text{S-MPNet}_{\text{hp}}$	84.0	80.0	92.9	35.5	72.3	80.4	**85.6**	**53.9**	80.4	73.3*

DISCUSSION ABOUT PROBING METHODOLOGY.

Our probing methodology requires a textual definition of concepts for evaluation. Hence, the resulting representations' quality can be impacted by the definition's length (Bouraoui, Camacho-Collados, and Schockaert, 2020) or absence. In the latter scenario, alternative approaches must be considered to obtain a suitable vector representation (Vulić et al., 2020). Additionally, the taxonomy's granularity can impact the representations' quality. Our methodology has been tested on low domain-specific levels (for example, a concept of "presenile dementia", with parent "dementia" and sibling "insanity"). However, coarse-grained taxonomies, with more abstract concepts in proportion, may not respect our target properties since relations between abstract concepts tend to be different from relations between more specific concepts, as suggested by Vulić et al. (2017).

5.5.2 Enhancing PLMs with hierarchy properties

In the following, we answer **RQ2:** *Does injecting hierarchy properties into PLMs using a task-agnostic methodology impact their representations?* Our underlying objective is to investigate the feasibility of our task-agnostic evaluation of PLMs w.r.t. hierarchy through the set of defined properties (see Section 5.3.1). To this end, we fine-tuned (see Section 5.4.3) and evaluated the models on the probes following the best evaluation setup for PLMs (discussed in Section 5.5). These fine-tuned models are referenced as hierarchy-enhanced PLMs (denoted PLM_{hp}) belonging to *Group 4* in Table 5.5, where we report the average accuracy after fine-tuning each model with 5 different random seeds and optimal hyper-parameters. Similar to RQ1, we consider the *All* for comparison, and we take as reference the corresponding vanilla model for our analysis.

Our results showed improvement in all models for the *All* column w.r.t. the original PLM. For instance, the models **BERT-L**$_{\text{hp}}$, **S-RoB**$_{\text{hp}}$, and **S-MPNet**$_{\text{hp}}$ are improved by 12.5, 3.3, and 3.3 points, respectively in comparison to their vanilla counterparts.

In particular, the property labeled as *A-S* was not included in the training dataset. However, it is noteworthy that all models demonstrated a consistent improvement in this particular property. Specifically, the performance score increased between $+0.8$ for **S-RoB**$_{\text{hp}}$ and $+10.5$ for **RoB-L**$_{\text{hp}}$. For instance, the models **BERT-L**$_{\text{hp}}$, **S-RoB**$_{\text{hp}}$, and **S-MPNet**$_{\text{hp}}$ showed an increase in performance from their respective initial scores of 24.6, 33.7, and 29.6 to their respective final scores of 31.6, 34.5, and 35.5.

These findings align with our hypothesis that *A-S* complements other property criteria without compromising initial performance. In contrast, the trained property *S-F* seems to have a more modest improvement, ranging from a slight degradation of -0.2 for the **S-MPNet**$_{\text{hp}}$ model to an increase of $+3.2$ for the **BERT-L**$_{\text{hp}}$ model. This suggests that, while our fine-tuning approach yields promising results, it still struggles to differentiate *sibling* and *far relation* representations effectively.

QUALITATIVE ANALYSIS

We further deepen our understanding of the impact of considering hierarchy on PLM's representations by comparing between **S-RoB** and **S-RoB**$_{\text{hp}}$. We particularly analyze the failures, in terms of wrong predictions of **S-RoB** using properties *P-S* and *A-S*. In Figure 5.2 (left) shows that both *ancestor* and *parent* relations initially exhibit greater distances, dist(n,p)= 52.1 and dist(n,a)= 75.5, for **S-RoB**, which subsequently decrease to 51.2, 63.0, respectively, after fine-tuning, for **S-RoB**$_{\text{hp}}$. Figure 5.2 (right), we selected some examples to demonstrate how the enhanced PLMs accurately reverse the trend of the distances between representations with respect to the evaluated properties.

To sum up, our experiments confirmed the feasibility of injecting hierarchy properties into PLMs, particularly for **BERT-L**, **S-RoB**, and **S-MPNet** models, with overall performance higher than all the evaluated vanilla PLMs.

5.5.3 Analyzing the transfer of hierarchy knowledge of PLMs to downstream tasks

Finally, we answer **RQ3:** *Can hierarchy-enhanced PLMs representations be transferred to downstream tasks?* We investigate in a sequential mode whether our hierarchy-enhanced PLMs using probes retain and leverage their knowledge when fine-tuned to perform a given downstream task.

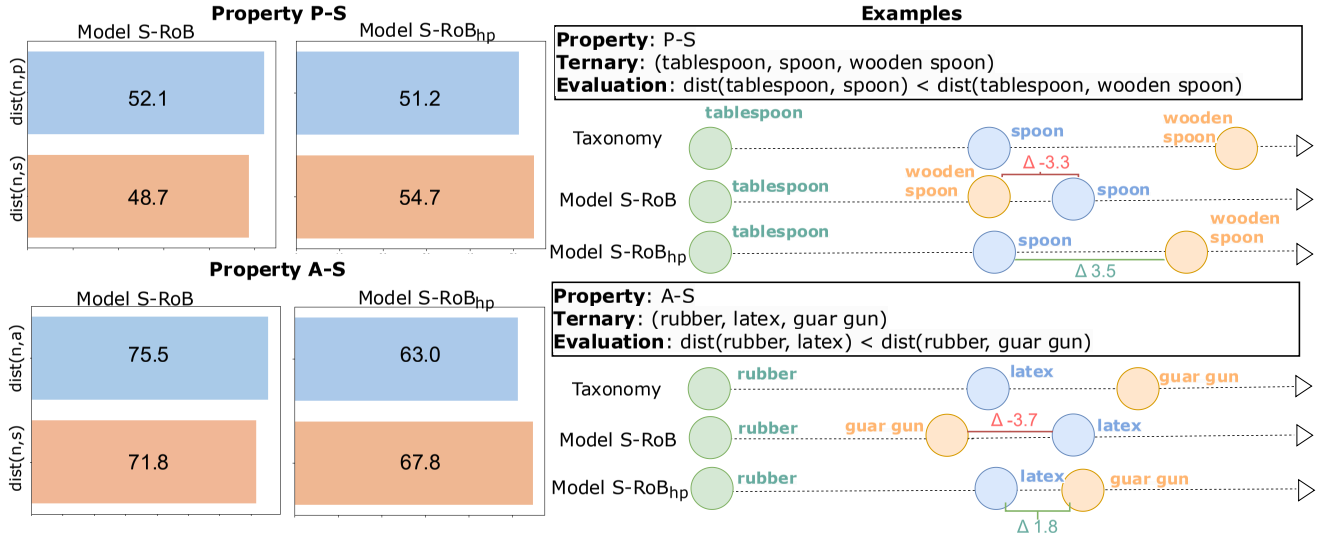
In the following, we introduce each downstream task and report our results.

5.5.3.1 *Downstream tasks*

HYPERNYM DISCOVERY.

This task evaluates an input term as a query and retrieves (or discovers) its suitable hypernyms from a target corpus. We used the

Figure 5.2: Avg. distance in the property evaluation for properties *P-S* and *A-S* (left). One example for each property, showing the distances of different concepts for a fixed node with models S-RoB and S-RoB$_{hp}$. We indicate the Δ distance between concepts (right).



SemEval-2018 Task 9 benchmark (Camacho-Collados et al., 2018) consisting of five subtasks covering general-purpose and domain-specific tasks.

Specifically, we consider the English subtask 1A which includes a textual corpus, a vocabulary including all valid hypernyms, and a training and testing set of hyponyms and gold hypernyms. Due to the size of the corpus and queries, we employed a re-ranking approach involving two steps.

In order to demonstrate the objective of this task, let us consider an example from the test set where the word query is "manga". The anticipated outcome is for the model to identify and retrieve the hypernyms such as "serial", "television series", "tv program series", "television program", and "tv show" from the given vocabulary.

Firstly, we considered two different models as first rankers to retrieve the top 1000 most relevant candidates: a semantic-retrieval approach with FT-wiki embeddings and cosine similarity, and BM25 with RM3 (Abdul-Jaleel et al., 2004). Then, we re-ranked these filtered results using our hierarchy-enhanced and vanilla PLMs, which were fine-tuned with a binary classification layer using the CTP architecture (for details, please refer to Section 3.3). We further explored an Oracle first ranker which extends the first ranker with missing golden candidates and acts as an upper bound of the performance of PLMs. We report MAP and P@5 ranking measures as proposed in Camacho-Collados et al. (2018).

TAXONOMY RECONSTRUCTION.

This task aims to construct a hierarchical taxonomy from a given set of words. We used the SemEval *TexEval-II* (Bordea, Lefever, and Buitelaar, 2016) dataset to compare with previous work (Chen, Lin, and Klein, 2021; Jain and Anke, 2022). We used the English-language version of the taxonomies *environment* and *science*. The dataset consists of an evaluation set with edge-based accuracy as the metric. We followed the CTP approach by training the vanilla and our hierarchy-enhanced PLMs on the Bansal et al. dataset, omitting overlapping terms with the evaluation dataset. We report standard Precision, Recall, and F1.

READING COMPREHENSION.

We use the RACE (Lai et al., 2017) dataset, consisting of English exams for middle and high school Chinese students with up to four possible answers. The questions are split into Middle and High sets,

where the High set is the most difficult. Reported values are given in terms of accuracy.

5.5.3.2 *Evaluation results*

We examine the transferability of the learned hierarchical representations to downstream tasks. Tables 5.6, 5.7, and 5.8 show results for each task. For Hypernym Discovery, our initial MAP scores were 2.2, 3.4, and 3.7 for FT-wiki, BM25, and BM25+RM3, respectively. Additionally, re-rankers **BERT-L**$_{hp}$ and **RoB-L**$_{hp}$ improved +5.4 and +5.1 w.r.t. vanilla versions for MAP and P@5 based on BM25+RM3 initial ranking. However, we noticed a slight degradation in performance for **S-RoB**$_{hp}$ and **S-MPNet**$_{hp}$ (similarly for FT-wiki column). We also observed that all models improved their results for the Oracle results, with smaller margin (+0.1) for **S-RoB**$_{hp}$ and **S-MPNet**$_{hp}$. These results motivate us to explore better first rankers. Moreover, considering the fact that this particular task heavily relies on hypernym relations (*group P-**), we assume that the enhanced representations are easily transferable to this task. For Taxonomy Reconstruction, we report the average metrics from both taxonomies. Three models, **BERT-L**$_{hp}$, **S-RoB**$_{hp}$, and **S-MPNet**$_{hp}$, showed improvements in F1 scores w.r.t their vanilla version (+0.4, +0.7, and +17.2, respectively), while **RoB-L**$_{hp}$ degraded its performance by −1.2. The improvements were primarily driven by better precision (0.7 − 32.0), with a smaller penalization on the recall (0.8 − 14.8). Considering higher precision as an indication of better quality in the concept representations, we argue that the enhanced representations from *groups P-**, *A-**, and *S-** are transferable for this task. For Reading Comprehension, learned representations may not be fully transferable. Specifically, we observed improvements for **RoB-L**$_{hp}$, but not for other models. This suggests that hierarchy knowledge is either forgotten (Kirkpatrick et al., 2017) or penalizing for this task. Thus, an appropriate setup is called since a sequential fine-tuning might lead to the hierarchy representations drift for this task.

Table 5.9 shows our results w.r.t. SOTA models. For the tasks of Taxonomy Reconstruction and Reading Comprehension[1], **S-MPNet**$_{hp}$ and **RoB-L**$_{hp}$ achieve competitive performances. However, the main trend is that specialized models, such as RRM, TaxoRL, and CoLISA, outperform fine-tuned enhanced models, which sheds light to the desirable room for improvement that a suitable transfer learning of hierarchy knowledge to downstream tasks would achieve. Overall, our empirical findings suggest that the enhanced PLM representations

1 All score recalculated for RoBERTa as suggested in the fairseq GitHub repository.

Table 5.6: Hypernym Discovery results for vanilla, enhanced PLMs. We report MAP and P@5 using *only* the FT-Wiki results (*default*), and including the gold results (*w/gold*). Best values are presented in **bold**.

	Hypernym Discovery					
	FT-wiki		**BM25+RM3**		**Oracle**	
Model	**MAP**	**P@5**	**MAP**	**P@5**	**MAP**	**P@5**
BERT-L	5.9	4.6	7.0	5.5	42.6	37.2
RoB-L	2.0	1.3	1.8	1.2	18.3	15.7
S-RoB	10.4	8.7	11.6	9.8	45.5	39.8
S-MPNet	9.9	7.8	10.6	8.6	46.1	39.6
BERT-L_{hp}	**11.3**	**9.7**	**12.4**	**11.0**	**53.6**	**48.5**
RoB-L_{hp}	5.0	4.1	5.5	4.5	32.2	27.0
S-RoB_{hp}	9.2	7.8	10.5	9.1	45.6	39.9
S-$MPNet_{hp}$	9.4	7.7	10.5	8.8	47.9	42.0

Table 5.7: Taxonomy Reconstruction results for vanilla, enhanced PLMs. We report the average of Precision (P), Recall (R), and F1 from all taxonomies. w.r.t. vanilla counterparts are presented in () and best values in **bold**.

Taxonomy Reconstruction			
Model	**P**	**R**	**F1**
BERT-L	15.5	48.0	22.9
RoB-L	17.3	46.7	24.8
S-RoB	17.2	33.6	22.6
S-MPNet	9.4	41.9	15.1
BERT-L_{hp}	16.2	45.0	23.3(+0.4)
RoB-L_{hp}	15.9	**49.4**	23.6
S-RoB_{hp}	18.3	32.8	23.3
S-$MPNet_{hp}$	**41.4**	27.1	**32.3**

Table 5.8: Reading Comprehension results for vanilla and enhanced. We report the accuracy. Results are average of three runs and best values in **bold**.

Reading Comprehension			
Model	**Middle**	**High**	**All**
BERT-L	75.0	66.3	68.8
RoB-L	86.3	80.0	81.9
S-RoB	57.9	51.4	53.3
S-MPNet	74.4	67.8	69.7
BERT-L$_{hp}$	74.3	63.4	66.6
RoB-L$_{hp}$	**87.8**	**81.4**	**83.2**
S-RoB$_{hp}$	57.3	51.0	52.8
S-MPNet$_{hp}$	46.2	42.2	43.4

Table 5.9: Results for Hypernym Discovery, Taxonomy Reconstruction and Reading Comprehension tasks. We present the best performing enhanced model from our experiments and the SOTA models for each task. Best values in **bold**.

Hypernym Discovery	**MAP**	**P@5**	
BERT-L$_{hp}$	12.4	11.0	
CRIM (Bernier-Colborne and Barrière, 2018)	19.8	19.0	
RMM (Bai et al., 2021)	**27.1**	**23.4**	
Taxonomy Reconstruction	**P**	**R**	**F1**
S-MPNet$_{hp}$	**41.4**	27.1	32.3
TaxoRL (Mao et al., 2018)	35.1	**35.1**	**35.1**
CTP (Chen, Lin, and Klein, 2021)	26.3	25.9	26.1
LMScorer (Jain and Anke, 2022)	29.8	28.6	29.1
Reading Comprehension	**Middle**	**High**	**All**
RoB-L$_{hp}$	87.8	81.4	83.2
RoBERTa (Liu et al., 2019)	86.5	81.8	82.8
DeBERTA (He, Gao, and Chen, 2021)	90.5	86.8	87.5
CoLISA(Dong et al., 2023)	**90.8**	**86.9**	**87.9**

are moderately transferable in a sequential mode of probe then finetune for hierarchy-aware tasks such as Hypernymy Discovery and Taxonomy Reconstruction, but detrimental for Reading Comprehension.

5.6 CONCLUSION

In this chapter, we aimed to evaluate whether the PLMs encode their knowledge representation hierarchically. Our research found that current evaluations of PLMs only consider superficial hierarchical features, which can lead to neglecting higher levels of hierarchical knowledge. To better understand the limitations of these models and provide insights into what information is helpful for PLMs to leverage their representations, comprehensive methods are needed to evaluate the hierarchy.

To address this gap, our work proposed a task-agnostic methodology for probing the capability of PLMs to capture hierarchy. We relied on the task diagnostic approach, which decomposes a given task into properties or steps to better understand the underlying mechanism leading to a model prediction.

We first identified hierarchy properties from taxonomies, capturing how concepts are distributed. More specifically, our properties rely on four taxonomic relationships: parent, ancestor, sibling, and far relative, and we rely on their taxonomic similarity to study the hierarchical relationship between words. Then, we constructed probes encoding these properties and a setup to evaluate the hierarchy knowledge in PLMs.

We further conducted experiments using probes to integrate explicit hierarchy knowledge into PLMs. Overall, our experiments showed that the evaluated PLMs, such as cross-encoder models like BERT, struggle to capture hierarchical relations, such as siblings and ancestor representations. However, we were able to improve the representations of PLMs by injecting hierarchy properties into them, which resulted in better performance on our evaluation.

Finally, we tested the transferability of the hierarchy knowledge acquired by our hierarchy-aware PLMs on different NLP downstream tasks, including hypernym discovery, taxonomic reconstruction, and reading comprehension. Our finding reveals that a kind of catastrophic forgetting can occur, leading to performance results under upper-bound performance of PLMs trained specifically on hierarchy-aware tasks.

Part III

CONCLUSION

6 CONCLUSION

6.1 SUMMARY

In this dissertation, we investigated different approaches that enhanced the knowledge representation of Pre-trained Language Models (PLMs) using external knowledge resources, particularly knowledge bases (KBs). Using KBs has proven beneficial to incorporating the structured and factual knowledge that PLMs typically ignore. Different methodologies have mainly focused on injecting textual information from knowledge as a factual and specialized knowledge source. However, these methodologies do not explicitly incorporate the structural aspects that KBs inherently possess. Consequently, we explored methodologies and adapted evaluation protocols to integrate this structural knowledge into PLMs.

In Chapter 2, we provided relevant background on knowledge bases, their sources, and components typically used in approaches that employ PLMs. We also reviewed different approaches that inject external knowledge, which can either be implicit or explicit. The main difference between these two approaches is how the structure is used and translated to deal with PLMs. Moreover, we presented common evaluation approaches used in the literature to determine the quality and quantity of knowledge injected into the model. These approaches typically depend on the PLM's knowledge capacity. We further introduced a less common evaluation approach called task diagnostic, which aims to decompose the target task and identify which parts are beneficial using the enhanced model. This adds more transparency to the model's performance.

As previously mentioned, the previous methods focused solely on injecting factual information from KBs and did not consider the hierarchical structure of KBs. In Chapter 3, we reviewed the main approaches that explicitly leverage the hierarchical structural information from KB. We start by reviewing static-word embeddings, which have been extensively explored in the literature. Then, we introduced some of the recent approaches, which are still limited due to the ongoing development of this research field.

In Chapter 4, we focused on a specific NLP task, the reasoning task, which is particularly improved with implicit injection methodologies. We found that the learning methodologies previously used

for this task struggled to find a human-based generalization. To address this issue, we proposed new learning methodologies that gradually teach different complexities of the reasoning task inspired by how humans learn incrementally. We evaluated the effectiveness of our proposed methodology by testing it on similar multi-hop reasoning tasks and related tasks such as question-answering and reading comprehension.

Finally, in Chapter 5, we extended our study of hierarchical representation under explicit hierarchical signals. We found that the current methods for evaluating hierarchy representation are limited and inherited from classical methods for static word embeddings. Therefore, we propose a task diagnostic approach in response to the need for more comprehensive evaluations for hierarchy. This task diagnostic characterized the hierarchy into hierarchy properties that capture the notion of hierarchy based on specific patterns in the structure. We also introduced an evaluation setup for PLMs using these properties. Additionally, we taught these properties to the models and assessed whether they improved performance on their representations and related NLP tasks.

6.2 CONTRIBUTIONS

Considering that the integration of knowledge into PLMs is a recent but rapidly evolving domain, we have done a comprehensive review of the most relevant knowledge integration methods from the SOTA. In Chapter 2, we surveyed methods for knowledge integration for PLMs based on the methodology approach used (implicit and explicit) and provided a brief introduction to the domain shift in the context of LLMs. We also presented the main evaluation methodologies. Similarly, in Chapter 3, we surveyed the main strategies for hierarchy injection, covering methods oriented for static word embeddings and the most recent methods for PLMs

Regarding our contributions, our research focused on the structural component of the injected knowledge, particularly exploring the benefits and limitations of different methodologies and the relevance of injecting this type of knowledge by evaluating downstream NLP tasks.

Our first contribution delved into the reasoning task and proposed different learning methodologies. Throughout the exploration of this venue, we highlight the following contributions:

1. The creation of the SHINet dataset, a synthetic dataset for multi-hop reasoning. It includes 1-hop and 2-hop levels for training

and evaluation. This dataset has been created by manually aligning the SHINRA and ConceptNet datasets. It has also been filtered to provide a more challenging version of reasoning, considering the bias typically introduced when creating synthetic datasets.

2. We proposed the training and evaluations of single-hop reasoning models, which are trained on a specific difficulty level. The objective is to understand the strengths and limitations of the reasoning abilities of PLMs by observing their contribution after learning different reasoning depths at varying difficulties. To achieve this, we have developed single-hop datasets that can be used to train and evaluate existing datasets in the literature (Single RuleTakers and SHINet).

3. We provided an algorithm to sample different training strategies based on the context information. Particularly, our algorithm defines different types of distractors according to their participation in the reasoning chain. Using this algorithm, we analyzed their impact on the task and the most helpful technique to leverage the training of PLMs.

4. We have replicated various previous research models to ensure a fair comparison. Specifically, we have reproduced the results of the Leap-of-Thought model (Talmor et al., 2020b) for reasoning tasks under various setups proposed in the original work. Additionally, we have replicated the work of Richardson and Sabharwal (2020) for question-answering. In doing so, we have updated the deprecated libraries with current ones, achieved comparison results in reproductions, and reorganized the datasets.

5. We provide code and trained models from our experience in a public repository for the sake of reproducibility in our work.

Our second contribution involved evaluating and integrating explicit hierarchical signals within the context of PLMs. This has resulted in the following contributions:

1. We proposed a comprehensive method for evaluating hierarchy based on hierarchy properties. These properties reflect the hierarchy distribution of concepts similar to taxonomies. Instead of simply relying on a hypernym link between two entities, our approach considers the relative placement of three entities, mimicking the substructure of a taxonomy.

2. We generated an evaluation dataset based on alignments of WordNet and Bansal et al. taxonomies to instantiate the hierarchy properties. The dataset contains concepts and contextualized definitions based on the sub-taxonomy.

3. An evaluation setup for these hierarchies is provided for PLMs. Mainly, we explored different PLM representations based on different tokens, layers, and aggregation methods to obtain the best configuration that favors these hierarchical representations.

4. We provided a training setup to teach PLMs about hierarchy properties. We relied on the Triplet Loss and triplet networks to this end. During the exploration, we also tried experimented with different architecture networks, such as cross-encoders and bi-encoder architectures

5. We analyzed multiple SOTA models using these hierarchy properties. This helped us to spot the main limitations of PLMs in dealing with hierarchical distributions, considering relations such as parent, ancestor, sibling, and others. Additionally, we trained these models by leveraging these properties, expanding our analysis to these more comprehensive representations.

6. Similar to the previous contribution, we provide the code, datasets, and models of all the resources developed in this work to assist future research in this domain.

6.3 PERSPECTIVES AND FUTURE DIRECTIONS

The knowledge integration field is an extensive topic, and different aspects could not be explored within the limited time span of this thesis. In the following, we will review some ideas related to the topics presented that could serve as a starting point for further work in this domain. Our perspectives on future work can be grouped into two categories: exploring novel and useful injection methods and adapting or improving architectural propositions.

Our perspectives for further work on different aspects of injection methodologies are the following:

IMPROVED PROMPTS AS IMPLICIT INTEGRATION FOR LLMS.

Using few-shot learning through prompting is a common approach to adapting Large Language Models (LLMs) for specific tasks. Currently, KBs are used as a resource to verify the accuracy of the information generated by LLMs, with training modules being used to

correct any inaccuracies. However, we believe that constructing richer prompts that capture structural elements can lead to capturing different hierarchical nuances of relationships. These richer prompts can leverage current text-patterns approaches and use adapted learning strategies.

BETTER METRICS FOR MULTI-HOP REASONING.

The evaluation of multi-hop reasoning relies on simple metrics such as accuracy, which determines whether the overall inference is correct or not. This evaluation approach can reveal if a model can correctly reason over a given inference problem at a higher level. However, this evaluation approach has limitations, as it does not consider the difficulty level (number of hops) of the reasoning problem evaluated. Moreover, it does not provide information about which parts of the reasoning chain the model could obtain. Although multi-hop reasoning is not the core subject of this thesis, we consider that better evaluation approaches could be helpful. Particularly, they could provide valuable information about the reasoning process, which could be helpful for interpretability approaches to elucidate whether a model gives a good answer based on the correct choices. Therefore, there is a need for better metrics that recall the inference path.

ENCODE HIERARCHY IN RELATIONS.

Our second contribution explored the use of explicit hierarchical signals, which mainly focused on different taxonomical relations, such as parent, ancestor, and sibling relations. These relations aimed to complement current evaluations that rely only on the parent relationship. However, we encourage to explore other explicit hierarchical signals that could provide essential knowledge to enhance representations. One promising direction for further exploration is the different natures of relationships in a knowledge base, which can be mainly hierarchical and non-hierarchical (Krackhardt, 2014). Previous work has explored some methods to characterize the degree of hierarchy of these relationships. Moreover, the binary distinction between these two types of relationships has shown promising results in knowledge-based embeddings (Du et al., 2022). Therefore, considering degrees of hierarchy when injecting information in PLMs could provide an additional dimension in the latent representation space towards enhancing this aspect of language, leading to capturing hierarchy characteristics such as granularity and direction.

As incremental contributions related to architectural challenges, we present the following:

ADAPTATION TO HYPERBOLIC SPACES.

The use of hyperbolic spaces showed an interesting approach in works related to static word embeddings. These spaces are useful for encoding the hierarchical distribution of KBs. However, there are limited studies exploring the use of these spaces in PLMs. One possible future direction for research is to consider these spaces in the architecture of the models. It would be interesting to evaluate the relevance of the hierarchy properties proposed in our second contribution using these spaces.

Another interesting approach to analyze is the learning projection of contextualized representations for evaluation purposes. Learning projections on final and intermediate representations in PLMs could help improve the hierarchical modeling of these models and comply with a hierarchical reference. However, implementing these methods would require redefining basic operations commonly available in libraries. Developing tools adapted to the most commonly used frameworks, such as AllenNLP or HuggingFace, is a crucial step in advancing further research in this domain.

SPAN REPRESENTATION FOR ENTITY AND RELATION

The way entities and relations are represented in PLMs has only been explored to a limited extent. In KEPLMs, most methods use simple approaches for span representations, such as considering the first token of the mention (ERNIE) or using average pooling for the participating entities (ERICA). However, these methods are limited and may miss important information related to the entities and relations. Moreover, they do not consider that an entity or relation can be expressed through different textual mentions. For example, an entity could be expressed by a mention in a text or using higher-level representations such as paragraphs or documents. Similarly, relationships could be formed by the interaction between two entities in a phrase or between sentences or documents. Previous research has examined various ways to capture these mixed representations, but the structural distribution of these representations has not been given much attention.

DEVELOPING DEDICATED TAXONOMIC SIMILARITY MODULES.

In our second contribution, we observed that incorporating hierarchical representation into PLMs improved the performance of tasks that rely on hierarchy-like structures. However, we also observed that this method did not work well for other NLP tasks less related to hierarchy. There could be two reasons for this outcome: either the injection strategy is not optimal for these types of problems, or there is an incompatibility in the representation space between semantic and taxonomic similarity.

We suggest testing this incompatibility hypothesis as an interesting approach to continue. It could be achieved by adding complementary modules at the architecture level that can learn different types of similarities. For instance, by training the Adapters module to integrate taxonomical similarity into PLMs already trained on semantic similarity, we can conduct experiments to gain more insight into this matter.

HIERARCHICAL LOSSES.

In language models, learning semantic relationships typically involves contrastive learning. Similar to the LEAR approach, the idea is to group similar words and calculate the distance between dissimilar words based on our semantic relationship criteria. The triplet or negative ranking loss emerged as derived losses from this approach. However, these losses do not consider different aspects of the hierarchy, such as granularity and direction or interactions between multiple entities for hierarchical injection purposes. In computer vision, recent losses were proposed to capture different nuances of the hierarchy better. It may be worthwhile to evaluate the adaptation of these losses in the text domain or propose more appropriate losses.

Part IV

APPENDIX

A

BI-ENCODER MODELS

This appendix provides a detailed definition of bi-encoder models and showcases the training details of the models in our second contribution.

A.1 DEFINITION

The introduction of Sentence-BERT (Reimers and Gurevych, 2019) has improved the embedding representations of the PLM network through siamese and triplet networks, referred to as bi-encoders. The refined version of BERT produces semantically meaningful sentence embeddings, making it useful in new domains such as large-scale semantic similarity comparison, clustering, and information retrieval through semantic search.

The conventional BERT architecture uses a cross-encoder paradigm for two-sentence input, but it is impractical for tasks involving numerous sentence combinations. To overcome this limitation, typical approaches rely on feeding individual sentences into BERT to extract fixed-size sentence embeddings. These embeddings are typically achieved by averaging the BERT output layer or using the [CLS] token, which may not always provide good-quality representations.

The incorporation of siamese network architecture facilitates the extraction of fixed-sized vectors for input sentences, overcoming the challenge of efficiently handling varying sentence pairs. This paradigm shift enhances the flexibility of BERT in accommodating diverse tasks and extends its applicability to scenarios where nuanced sentence-level embeddings are essential.

A.2 BI-ENCODERS VS CROSS-ENCODERS

Cross-encoder and bi-encoder models represent contrasting approaches, each presenting distinctive advantages and drawbacks. Figure A.1 shows both types of model architecture.

Cross-encoders consider the global context of both sentences simultaneously and provide a comprehensive perspective on their relationship. This approach is advantageous in tasks where the interplay between sentences depends on their overall context. How-

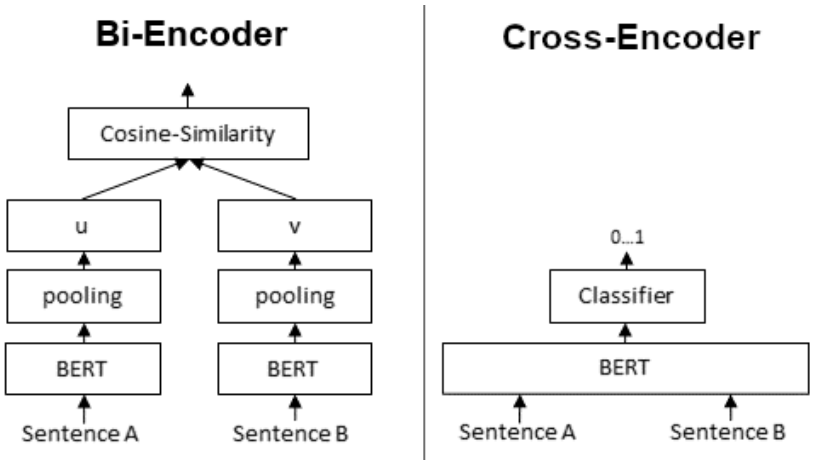


Figure A.1: Network architectures for cross-encoders and bi-encoder models. Source:

ever, cross-encoders require concurrent processing of both sentences, which makes them computationally intensive and limits scalability. Also, the sequential nature of cross-encoder processing hinders efficient parallelization, affecting training and inference efficiency. Additionally, cross-encoders do not produce sentence embeddings that can be used independently.

In contrast, bi-encoders process each sentence independently, facilitating parallelization and making them highly scalable. This approach makes bi-encoders a good fit for applications dealing with extensive datasets. Bi-encoders are also versatile in adapting to various tasks independently. However, their emphasis on the local context within individual sentences might limit their effectiveness in tasks requiring a broader understanding of the global relationship between sentences. Additionally, bi-encoders may not perform optimally in explicit pairwise classification tasks compared to their cross-encoder counterparts.

To sum up, the choice between cross-encoder and bi-encoder models depends on the specific requirements of the task. Cross-encoders excel in tasks where a holistic understanding of the relationship between sentences is crucial, while bi-encoders offer efficiency and adaptability for a broader range of tasks, particularly those emphasizing local context or requiring scalable solutions.

Table A.1: Pre-trained Bi-encoder Models available on the HuggingFace hub.

Model	PLM	Embedding dimension	Max length
all-mpnet-base-v2	MPNet	768	512
all-distilroberta-v1	RoBERTa	768	512
all-MiniLM-L12-v2	MiniLM (distilled BERT)	384	256

A.3 PRE-TRAINED BI-ENCODER MODELS

The pre-trained models specifications are shown in Table A.1. Please note that all these models are trained using the same dataset collection with over 1 billion of sentence pairs.

This dataset collection is composed of the following dataset: Reddit comments (2015-2018), S2ORC Citation pairs (Abstracts), WikiAnswers Duplicate question pairs, PAQ (Question, Answer), S2ORC Citation pairs (Titles), S2ORC (Title, Abstract), Stack Exchange (Title, Body), MS MARCO triplets, GOOAQ: Open Question Answering with Diverse Answer Types, Yahoo Answers (Title, Answer), Code Search, COCO Image captions, SPECTER citation triplets, Yahoo Answers (Question, Answer), Yahoo Answers (Title, Question), SearchQA, Eli5, Flickr 30k, Stack Exchange Duplicate questions (titles), AllNLI (SNLI and MultiNLI), Stack Exchange Duplicate questions (bodies), Stack Exchange Duplicate questions (titles+bodies), Sentence Compression, Wikihow, Altlex, Quora Question Triplets, Simple Wikipedia, Natural Questions (NQ), SQuAD2.0, TriviaQA.

B

COMPUTING INFRASTRUCTURE AND BUDGET

All experiments were performed in a server Dell R740 bi pro Intel Xeon 2630 using Nvidia RTX6000 graphic card. A single training and test took around 100 and 120 minutes under this infrastructure. In summary, to compute the results of RQ1 and RQ2 we used approximately 120 GPU hours.

To compute the results of RQ3, including the inoculation technique, we used approximately 1,730 GPU hours.

C

RESULTS OF HIERARCHY PROPERTIES

Table C.1 shows the accuracy on each hierarchy property on the test dataset with method *avg* token representation and using the L2 distance.

Table C.1: Accuracy on each hierarchy property on the test dataset with method *avg* token representation and using the L2 distance.

Model	**P-A**	**P-S**	**P-F**	**A-S**	**A-F**	**S-F**	**P-***	**A-***	**S-***	**All**
					Group 1: Baselines					
FT-wiki	73.2	48.9	75.2	29.3	47.0	73.2	65.8	38.2	73.2	59.1
					Group 2: Vanilla PLMs					
BERT	74.3	37.6	67.8	22.8	41.4	72.6	59.9	32.1	72.6	54.9
BERT-L	77.1	41.0	75.2	25.1	46.7	74.8	64.4	35.9	74.8	58.4
RoB-L	61.5	38.2	60.5	32.4	42.6	65.9	53.4	37.5	65.9	52.3
S-RoB	83.3	63.0	87.8	30.6	59.8	77.3	78.0	45.2	77.3	66.8
S-MPNet	84.9	68.2	87.7	28.2	65.5	80.8	80.3	46.9	80.8	69.3
					Group 3: KG-Enhanced PLMs					
ERNIE	74.4	41.8	70.4	25.0	43.5	73.2	62.2	34.3	73.2	56.6
CTP	60.4	21.9	35.9	23.8	22.7	61.0	39.4	23.3	61.0	41.2

BIBLIOGRAPHY

Abdul-Jaleel, Nasreen, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade (2004). "UMass at TREC 2004: Novelty and HARD." In: *Computer Science Department Faculty Publication Series*, p. 189.

Alsuhaibani, Mohammed, Takanori Maehara, and Danushka Bollegala (2018). "Joint learning of hierarchical word embeddings from a corpus and a taxonomy." In: *Automated Knowledge Base Construction (AKBC)*.

Aly, Rami, Shantanu Acharya, Alexander Ossa, Arne Köhn, Chris Biemann, and Alexander Panchenko (2019). "Every Child Should Have Parents: A Taxonomy Refinement Algorithm Based on Hyperbolic Term Embeddings." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4811–4817.

Amigó, Enrique, Alejandro Ariza-Casabona, Victor Fresno, and M Antònia Martí (2022). "Information Theory–based Compositional Distributional Semantics." In: *Computational Linguistics* 48.4, pp. 907–948.

Anderson, John R. (1980). *Cognitive Psychology and Its Implications*. W.H.Freeman & Co Ltd.

Apidianaki, Marianna (2023). "From word types to tokens and back: A survey of approaches to word meaning representation and interpretation." In: *Computational Linguistics*, pp. 465–523.

Bai, Yuhang, Richong Zhang, Fanshuang Kong, Junfan Chen, and Yongyi Mao (2021). "Hypernym discovery via a recurrent mapping model." In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2912–2921.

Balazevic, Ivana, Carl Allen, and Timothy Hospedales (2019). "Multi-relational poincaré graph embeddings." In: *Advances in Neural Information Processing Systems* 32.

Bansal, Mohit, David Burkett, Gerard De Melo, and Dan Klein (2014). "Structured learning for taxonomy induction with belief propagation." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1041–1051.

Baroni, Marco and Alessandro Lenci (2011). "How we BLESSed distributional semantic evaluation." In: *Proceedings of the GEMS 2011*

Workshop on GEometrical Models of Natural Language Semantics, pp. 1–10.

Battaglia, Peter W, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. (2018). "Relational inductive biases, deep learning, and graph networks. arXiv 2018." In: *arXiv preprint arXiv:1806.01261*.

Bernier-Colborne, Gabriel and Caroline Barrière (June 2018). "CRIM at SemEval-2018 Task 9: A Hybrid Approach to Hypernym Discovery." In: *Proceedings of the 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana, pp. 725–731.

Bizer, Christian, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann (2009). "Dbpedia-a crystallization point for the web of data." In: *Journal of web semantics* 7.3, pp. 154–165.

Bodenreider, Olivier (2004). "The unified medical language system (UMLS): integrating biomedical terminology." In: *Nucleic acids research* 32.suppl_1, pp. D267–D270.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching Word Vectors with Subword Information." In: *TACL 2017* 5, pp. 135–146.

Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor (2008). "Freebase: a collaboratively created graph database for structuring human knowledge." In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250.

Bordea, Georgeta, Paul Buitelaar, Stefano Faralli, and Roberto Navigli (2015). "Semeval-2015 task 17: Taxonomy extraction evaluation." In: *Proceedings of the 9th International Workshop on Semantic Evaluation*.

Bordea, Georgeta, Els Lefever, and Paul Buitelaar (2016). "Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2)." In: *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pp. 1081–1091.

Bordes, Antoine, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko (2013). "Translating embeddings for modeling multi-relational data." In: *Advances in neural information processing systems* 26.

Bosselut, Antoine, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi (2019). "COMET: Commonsense Transformers for Automatic Knowledge Graph Construction." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4762–4779.

Bouraoui, Zied, Jose Camacho-Collados, and Steven Schockaert (2020). "Inducing relational knowledge from BERT." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05, pp. 7456–7463.

Budanitsky, Alexander (1999). *Lexical semantic relatedness and its application in natural language processing*. Tech. rep. technical report CSRG-390, Department of Computer Science, University of Toronto.

Budanitsky, Alexander and Graeme Hirst (2006). "Evaluating wordnet-based measures of lexical semantic relatedness." In: *Computational linguistics* 32.1, pp. 13–47.

Camacho-Collados, Jose (2017). "Why we have switched from building full-fledged taxonomies to simply detecting hypernymy relations." In: *arXiv preprint arXiv:1703.04178*.

Camacho-Collados, Jose, Claudio Delli Bovi, Luis Espinosa Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion (2018). "SemEval-2018 task 9: Hypernym discovery." In: *Proceedings of the 12th international workshop on semantic evaluation*, pp. 712–724.

Camacho-Collados, Jose and Mohammad Taher Pilehvar (2018). "From word to sense embeddings: A survey on vector representations of meaning." In: *Journal of Artificial Intelligence Research* 63, pp. 743–788.

Câmara, Arthur and Claudia Hauff (2020). "Diagnosing BERT with retrieval heuristics." In: *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42*. Springer, pp. 605–618.

Caruana, Rich (July 1997). "Multitask Learning." In: *Mach. Learn.* 28.1, 41–75. ISSN: 0885-6125. DOI: 10.1023/A:1007379606734. URL: https://doi.org/10.1023/A:1007379606734.

Chaabouni, Rahma, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni (July 2020). "Compositionality and Generalization In Emergent Languages." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4427–4442. DOI: 10.18653/v1/2020.acl-main.407. URL: https://aclanthology.org/2020.acl-main.407.

Chandrasekaran, Dhivya and Vijay Mago (2021). "Evolution of semantic similarity—a survey." In: *ACM Computing Surveys (CSUR)* 54.2, pp. 1–37.

Chang, Yingshan, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk (2022). "Webqa: Multihop and multi-

modal qa." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16495–16504.

Chen, Bohan and Andrea L Bertozzi (2023). "AutoKG: Efficient Automated Knowledge Graph Generation for Language Models." In: *arXiv preprint arXiv:2311.14740*.

Chen, Boli, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing (2020a). "Probing BERT in Hyperbolic Spaces." In: *International Conference on Learning Representations*.

Chen, Catherine, Kevin Lin, and Dan Klein (2021). "Constructing Taxonomies from Pretrained Language Models." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4687–4700.

Chen, Fenxiao, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo (2020b). "Graph representation learning: a survey." In: *APSIPA Transactions on Signal and Information Processing* 9, e15.

Chen, Yi, Jiayang Cheng, Haiyun Jiang, Lemao Liu, Haisong Zhang, Shuming Shi, and Ruifeng Xu (2022). "Learning from sibling mentions with scalable graph inference in fine-grained entity typing." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2076–2087.

Choudhary, Nurendra and Chandan K Reddy (2023). "Complex Logical Reasoning over Knowledge Graphs using Large Language Models." In: *arXiv preprint arXiv:2305.01157*.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. (2023). "Palm: Scaling language modeling with pathways." In: *Journal of Machine Learning Research* 24.240, pp. 1–113.

Clark, Peter, Oyvind Tafjord, and Kyle Richardson (2021). "Transformers as soft reasoners over language." In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 3882–3890.

Das, Rajarshi, Ameya Godbole, Manzil Zaheer, Shehzaad Dhuliawala, and Andrew McCallum (Nov. 2019). "Chains-of-Reasoning at TextGraphs 2019 Shared Task: Reasoning over Chains of Facts for Explainable Multi-hop Inference." In: *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*. Hong Kong: Association for Computational Linguistics, pp. 101–117. DOI: 10.18653/v1/D19-5313. URL: https://aclanthology.org/D19-5313.

Dash, Sarthak, Md Faisal Mahbub Chowdhury, Alfio Gliozzo, Nandana Mihindukulasooriya, and Nicolas Rodolfo Fauceglia (2020).

"Hypernym detection using strict partial order networks." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05, pp. 7626–7633.

Dekang, LIN (1998). "An information-theoretic definition of similarity." In: *Machine Learningˆ* Proceedings of the Fifteenth International Conference (ICML'98)*, pp. 296–304.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Dhingra, Bhuwan, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl (2018). "Embedding Text in Hyperbolic Spaces." In: *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, pp. 59–69.

Ding, Ming, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang (July 2019). "Cognitive Graph for Multi-Hop Reading Comprehension at Scale." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 2694–2703. DOI: 10.18653/v1/P19-1259. URL: https://aclanthology.org/P19-1259.

Dong, Mengxing, Bowei Zou, Yanling Li, and Yu Hong (2023). "CoLISA: Inner Interaction Via Contrastive Learning For Multi-Choice Reading Comprehension." In: *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 264–278. ISBN: 978-3-031-28243-0. DOI: 10.1007/978-3-031-28244-7_17.

Dong, Xingping and Jianbing Shen (2018). "Triplet loss in siamese network for object tracking." In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 459–474.

Dror, Rotem, Segev Shlomov, and Roi Reichart (2019). "Deep dominance-how to properly compare deep neural models." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2773–2785.

Du, Yuntao, Xinjun Zhu, Lu Chen, Baihua Zheng, and Yunjun Gao (2022). "Hakg: Hierarchy-aware knowledge gated network for recommendation." In: *Proceedings of the 45th International ACM SI-*

GIR Conference on Research and Development in Information Retrieval, pp. 1390–1400.

Elazar, Yanai, Hongming Zhang, Yoav Goldberg, and Dan Roth (Nov. 2021). "Back to Square One: Artifact Detection, Training and Commonsense Disentanglement in the Winograd Schema." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 10486–10500. DOI: 10.18653/v1/2021.emnlp-main.819. URL: https://aclanthology.org/2021.emnlp-main.819.

Elkan, Charles and Russell Greiner (1993). *Building large knowledge-based systems: Representation and inference in the cyc project: DB Lenat and RV Guha*.

Elsahar, Hady, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl (2018). "T-rex: A large scale alignment of natural language with knowledge base triples." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Espinosa-Anke, Luis, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion (2016). "Supervised distributional hypernym discovery via domain adaptation." In: *Conference on Empirical Methods in Natural Language Processing; 2016 Nov 1-5; Austin, TX. Red Hook (NY): ACL; 2016. p. 424-35*. ACL (Association for Computational Linguistics).

Faruqui, Manaal, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer (Aug. 2016). "Problems With Evaluation of Word Embeddings Using Word Similarity Tasks." In: *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany: Association for Computational Linguistics, pp. 30–35. DOI: 10.18653/v1/W16-2506. URL: https://aclanthology.org/W16-2506.

Févry, Thibault, Livio Baldini Soares, Nicholas Fitzgerald, Eunsol Choi, and Tom Kwiatkowski (2020). "Entities as Experts: Sparse Memory Access with Entity Supervision." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4937–4951.

Fu, Ruiji, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu (2014). "Learning semantic hierarchies via word embeddings." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1199–1209.

Ganea, Octavian, Gary Bécigneul, and Thomas Hofmann (2018a). "Hyperbolic entailment cones for learning hierarchical embed-

dings." In: *International Conference on Machine Learning*. PMLR, pp. 1646–1655.

— (2018b). "Hyperbolic neural networks." In: *Advances in neural information processing systems* 31.

Geffet, Maayan and Ido Dagan (2005). "The distributional inclusion hypotheses and lexical entailment." In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 107–114.

Gulcehre, Caglar, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. (2018). "Hyperbolic Attention Networks." In: *International Conference on Learning Representations*.

Gupta, Amit, Rémi Lebret, Hamza Harkous, and Karl Aberer (2017). "Taxonomy induction using hypernym subsequences." In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1329–1338.

Gupta, Amit, Francesco Piccinno, Mikhail Kozhevnikov, Marius Pasca, and Daniele Pighin (2016). "Revisiting Taxonomy Induction over Wikipedia." In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2300–2309.

Guu, Kelvin, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang (2020). "Retrieval augmented language model pretraining." In: *International conference on machine learning*. PMLR, pp. 3929–3938.

Hadsell, Raia, Sumit Chopra, and Yann LeCun (2006). "Dimensionality reduction by learning an invariant mapping." In: *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*. Vol. 2. IEEE, pp. 1735–1742.

Hamilton, William L, Rex Ying, and Jure Leskovec (2017). "Representation Learning on Graphs: Methods and Applications." In.

Han, Jiuzhou, Nigel Collier, Wray Buntine, and Ehsan Shareghi (2023). "PiVe: Prompting with Iterative Verification Improving Graph-based Generative Capability of LLMs." In: *arXiv e-prints*, arXiv–2305.

Harris, Zellig S (1954). "Distributional structure." In: *Word* 10.2-3, pp. 146–162.

He, Bin, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu (2020). "BERT-MK: Integrating Graph Contextualized Knowledge into Pre-trained Language Models." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2281–2290.

He, Pengcheng, Jianfeng Gao, and Weizhu Chen (2021). "Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing." In: *arXiv preprint arXiv:2111.09543*.

He, Tianxing, Kyunghyun Cho, and James Glass (2021). "An Empirical Study on Few-shot Knowledge Probing for Pretrained Language Models." In: *arXiv e-prints*, arXiv–2109.

Hearst, MA (1992). "Automatic acquisition of hyponyms from large text corpora in proc." In: *14th International Conference Computational Linguistics, Nantes France*.

Hendrycks, Dan and Kevin Gimpel (2016). "Gaussian error linear units (gelus)." In: *arXiv preprint arXiv:1606.08415*.

Hewitt, John and Percy Liang (2019). "Designing and Interpreting Probes with Control Tasks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743.

Hewitt, John and Christopher D Manning (2019). "A structural probe for finding syntax in word representations." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138.

Hou, Yifan, Guoji Fu, and Mrinmaya Sachan (2022). "What has been Enhanced in my Knowledge-Enhanced Language Model?" In: *EMNLP 2022*.

Houlsby, Neil, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly (2019). "Parameter-efficient transfer learning for NLP." In: *International Conference on Machine Learning*. PMLR, pp. 2790–2799.

Hovy, Eduard, Zornitsa Kozareva, and Ellen Riloff (2009). "Toward completeness in concept extraction and classification." In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 948–957.

Hu, Edward J, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. (2021). "LoRA: Low-Rank Adaptation of Large Language Models." In: *International Conference on Learning Representations*.

Hu, Linmei, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li (2023). "A survey of knowledge enhanced pre-trained language models." In: *IEEE Transactions on Knowledge and Data Engineering*.

Hu, Zhiting, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric Xing (2015). "Entity hierarchy embedding." In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1292–1300.

Hwang, Jena D, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi (2021). "(Comet-) atomic 2020: on symbolic and neural commonsense knowledge graphs." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 7, pp. 6384–6392.

Jain, Devansh and Luis Espinosa Anke (2022). "Distilling Hypernymy Relations from Language Models: On the Effectiveness of Zero-Shot Taxonomy Induction." In: *The 11th Joint Conference on Lexical and Computational Semantics*, p. 151.

Ji, Guoliang, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao (July 2015). "Knowledge Graph Embedding via Dynamic Mapping Matrix." In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (NAACL)*. Beijing, China, pp. 687–696.

Jiang, Jay J and David W Conrath (1997). "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy." In: *Proceedings of the 10th Research on Computational Linguistics International Conference*, pp. 19–33.

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

Kacmajor, Magdalena and John D Kelleher (2020). "Capturing and measuring thematic relatedness." In: *Language Resources and Evaluation* 54.3, pp. 645–682.

Kassner, Nora, Benno Krojer, and Hinrich Schütze (2020). "Are Pretrained Language Models Symbolic Reasoners over Knowledge?" In: *Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 552–564.

Kassner, Nora and Hinrich Schütze (2020). "Negated and Misprimed Probes for Pretrained Language Models: Birds Can Talk, But Cannot Fly." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7811–7818.

Kaushik, Prakhar, Adam Kortylewski, Alex Gain, and Alan Yuille (2021). "Understanding Catastrophic Forgetting and Remembering in Continual Learning with Optimal Relevance Mapping." In: *Fifth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*.

Ke, Pei, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang (2020). "SentiLARE: Sentiment-Aware Language Representation Learning with Linguistic Knowledge." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6975–6988.

Khan, Asifullah, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi (2020). "A survey of the recent architectures of deep convolutional neural networks." In: *Artificial intelligence review* 53, pp. 5455–5516.

Kiela, Douwe, Laura Rimell, Ivan Vulic, and Stephen Clark (2015). "Exploiting image generality for lexical entailment detection." In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. ACL; East Stroudsburg, PA, pp. 119–124.

Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell (2017). "Overcoming catastrophic forgetting in neural networks." In: *Proceedings of the National Academy of Sciences* 114.13, pp. 3521–3526. DOI: 10.1073/pnas.1611835114. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1611835114. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

Kozareva, Zornitsa, Ellen Riloff, and Eduard Hovy (2008). "Semantic class learning from the web with hyponym pattern linkage graphs." In: *Proceedings of ACL-08: HLT*, pp. 1048–1056.

Krackhardt, David (2014). "Graph theoretical dimensions of informal organizations." In: *Computational organization theory*. Psychology Press, pp. 107–130.

Krathwohl, David R (2002). "A revision of Bloom's taxonomy: An overview." In: *Theory into practice* 41.4, pp. 212–218.

Lai, Guokun, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy (2017). "RACE: Large-scale ReAding Comprehension Dataset From Examinations." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794.

Lapesa, Gabriella, Stefan Evert, and Sabine Schulte im Walde (Aug. 2014). "Contrasting Syntagmatic and Paradigmatic Relations: Insights from Distributional Semantic Models." In: *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*. Ed. by Johan Bos, Anette Frank, and Roberto Navigli. Dublin, Ireland: Association for Computational Linguistics

and Dublin City University, pp. 160–170. DOI: 10.3115/v1/S14-1020. URL: https://aclanthology.org/S14-1020.

Leacock, Claudia (1998). "Combining local context and WordNet similarity for word sense identification." In: *WordNet: A Lexical Reference System and its Application*, pp. 265–283.

Lerner, Paul, Olivier Ferret, Camille Guinaudeau, Hervé Le Borgne, Romaric Besançon, José G Moreno, and Jesús Lovón Melgarejo (2022). "ViQuAE, a dataset for knowledge-based visual question answering about named entities." In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3108–3120.

Levine, Yoav, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham (2020). "SenseBERT: Driving Some Sense into BERT." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4656–4667.

Levy, Omer, Steffen Remus, Chris Biemann, and Ido Dagan (2015). "Do supervised distributional methods really learn lexical inference relations?" In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 970–976.

Li, Jiacheng, Yannis Katsis, Tyler Baldwin, Ho-Cheol Kim, Andrew Bartko, Julian McAuley, and Chun-Nan Hsu (2022). "SPOT: Knowledge-Enhanced Language Representations for Information Extraction." In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 1124–1134.

Li, Zhongyang, Xiao Ding, Ting Liu, J Edward Hu, and Benjamin Van Durme (2021). "Guided generation of cause and effect." In: pp. 3629–3636.

Lin, Jimmy, Rodrigo Nogueira, and Andrew Yates (2021). *Pretrained Transformers for Text Ranking: BERT and Beyond*. Morgan & Claypool Publishers.

Lin, Yankai, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu (2015). "Learning entity and relation embeddings for knowledge graph completion." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 29. 1.

Lin, Yongjie, Yi Chern Tan, and Robert Frank (2019). "Open Sesame: Getting inside BERT's Linguistic Knowledge." In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 241–253.

Liu, Chunhua, Trevor Cohn, and Lea Frermann (July 2023). "Seeking Clozure: Robust Hypernym extraction from BERT with Anchored

Prompts." In: *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*. Ed. by Alexis Palmer and Jose Camacho-collados. Toronto, Canada: Association for Computational Linguistics, pp. 193–206. DOI: 10.18653/v1/2023.starsem-1.18. URL: https://aclanthology.org/2023.starsem-1.18.

Liu, Jingping, Menghui Wang, Chao Wang, Jiaqing Liang, Lihan Chen, Haiyun Jiang, Yanghua Xiao, and Yunwen Chen (2021). "Learning term embeddings for lexical taxonomies." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 7, pp. 6410–6417.

Liu, Nelson F., Roy Schwartz, and Noah A. Smith (June 2019). "Inoculation by Fine-Tuning: A Method for Analyzing Challenge Datasets." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2171–2179. DOI: 10.18653/v1/N19-1225. URL: https://aclanthology.org/N19-1225.

Liu, Weijie, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang (2020). "K-bert: Enabling language representation with knowledge graph." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 03, pp. 2901–2908.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." In.

Lovón-Melgarejo, Jesús, José G Moreno, Romaric Besançon, Olivier Ferret, and Lynda Tamine (2022). "Can We Guide a Multi-Hop Reasoning Language Model to Incrementally Learn at Each Single-Hop?" In: *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 1455–1466.

Lovón-Melgarejo, Jesús, Jose G Moreno, Romaric Besançon, Olivier Ferret, and Lynda Tamine (2024). "Probing Pretrained Language Models with Hierarchy Properties." In: *Advances in Information Retrieval*. Springer International Publishing.

Lovón-Melgarejo, Jesús, Laure Soulier, Karen Pinel-Sauvagnat, and Lynda Tamine (2021). "Studying catastrophic forgetting in neural ranking models." In: *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*. Springer, pp. 375–390.

Ma, Kaixin, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari (2021). "Knowledge-driven data

construction for zero-shot evaluation in commonsense question answering." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 15, pp. 13507–13515.

Manning, Christopher D, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy (2020). "Emergent linguistic structure in artificial neural networks trained by self-supervision." In: *Proceedings of the National Academy of Sciences* 117.48, pp. 30046–30054.

Mao, Yuning, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han (2018). "End-to-End Reinforcement Learning for Automatic Taxonomy Induction." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2462–2472.

Mao, Yuning, Jingjing Tian, Jiawei Han, and Xiang Ren (Nov. 2019). "Hierarchical Text Classification with Reinforced Label Assignment." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, pp. 445–455. DOI: 10.18653/v1/D19-1042. URL: https://aclanthology.org/D19-1042.

Maudslay, Rowan Hall, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell (2020). "A Tale of a Probe and a Parser." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7389–7395.

Mawson, Christopher Orlando Sylvester (1911). *Roget's international thesaurus of English words and phrases*. Crowell.

Meng, Yu, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han (2019). "Spherical text embedding." In: *Advances in neural information processing systems* 32.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed representations of words and phrases and their compositionality." In: *Advances in neural information processing systems* 26.

Mikolov, Tomáš, Wen-tau Yih, and Geoffrey Zweig (2013). "Linguistic regularities in continuous space word representations." In: *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pp. 746–751.

Miller, George A (1995). "WordNet: a lexical database for English." In: *Communications of the ACM* 38.11, pp. 39–41.

Min, Sewon, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi (2019). "Multi-hop Reading Comprehension through Question Decomposition and Rescoring." In: *ACL*.

Minsky, Marvin (1974). *A framework for representing knowledge*.

Mrkšić, Nikola, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young (2017). "Semantic Specialization of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints." In: *Transactions of the Association for Computational Linguistics* 5. Ed. by Lillian Lee, Mark Johnson, and Kristina Toutanova, pp. 309–324. DOI: 10.1162/tacl_a_00063. URL: https://aclanthology.org/Q17-1022.

Nickel, Maximilian, Xueyan Jiang, and Volker Tresp (2014). "Reducing the rank in relational factorization models by including observable patterns." In: *Advances in Neural Information Processing Systems* 27.

Nickel, Maximillian and Douwe Kiela (2017). "Poincaré embeddings for learning hierarchical representations." In: *Advances in neural information processing systems* 30.

Onoe, Yasumasa, Michael Boratko, Andrew McCallum, and Greg Durrett (2021). "Modeling Fine-Grained Entity Types with Box Embeddings." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Pan, Shirui, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu (2023). "Unifying Large Language Models and Knowledge Graphs: A Roadmap." In: *arXiv preprint arXiv:2306.08302*.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

Peters, Matthew E, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith (2019). "Knowledge Enhanced Contextual Word Representations." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 43–54.

Petroni, Fabio, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller (2019). "Language Models as Knowledge Bases?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and*

the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2463–2473.

Pfeiffer, Jonas, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych (2021). "AdapterFusion: Non-Destructive Task Composition for Transfer Learning." In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503.

Pfeiffer, Jonas, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder (2020). "MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7654–7673.

Pimentel, Tiago, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell (2020). "Information-Theoretic Probing for Linguistic Structure." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Plebe, Alessio (2011). *Self-Organization of Object Categories in a Cortical Artificial Model*. DOI: 10.5772/13291.

Poerner, Nina, Ulli Waltinger, and Hinrich Schütze (2020). "E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 803–818.

Qin, Yujia, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou (2021). "ERICA: Improving Entity and Relation Understanding for Pre-trained Language Models via Contrastive Learning." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3350–3363.

Qiu, Xipeng, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang (2020). "Pre-trained Models for Natural Language Processing: A Survey." In: *CoRR* abs/2003.08271. arXiv: 2003.08271. URL: https://arxiv.org/abs/2003.08271.

Quillian, M Ross (1966). *Semantic memory*. Air Force Cambridge Research Laboratories, Office of Aerospace Research ...

Rada, Roy, Hafedh Mili, Ellen Bicknell, and Maria Blettner (1989). "Development and application of a metric on semantic nets." In: *IEEE transactions on systems, man, and cybernetics* 19.1, pp. 17–30.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. (2018). "Improving language understanding by generative pre-training." In.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392.

Raman, Natraj, Sameena Shah, and Manuela Veloso (2022). "Structure and Semantics Preserving Document Representations." In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 780–790.

Rei, Marek and Ted Briscoe (2014). "Looking for hyponyms in vector space." In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pp. 68–77.

Reimers, Nils and Iryna Gurevych (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992.

Resnik, Philip (1995). "Using information content to evaluate semantic similarity in a taxonomy." In: *Proceedings of the 14th international joint conference on Artificial intelligence-Volume 1*, pp. 448–453.

Richardson, Kyle, Hai Hu, Lawrence Moss, and Ashish Sabharwal (2020). "Probing natural language inference models through semantic fragments." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05, pp. 8713–8721.

Richardson, Kyle and Ashish Sabharwal (2020). "What does my qa model know? devising controlled probes using expert knowledge." In: *Transactions of the Association for Computational Linguistics* 8, pp. 572–588.

Roberts, Adam, Colin Raffel, and Noam Shazeer (2020). "How Much Knowledge Can You Pack Into the Parameters of a Language Model?" In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5418–5426.

Roller, Stephen, Katrin Erk, and Gemma Boleda (2014). "Inclusive yet selective: Supervised distributional hypernymy detection." In: *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pp. 1025–1036.

Roller, Stephen, Douwe Kiela, and Maximilian Nickel (2018). "Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 358–363.

Rossi, Andrea, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo (2021). "Knowledge Graph Embedding

for Link Prediction: A Comparative Analysis." In: *ACM Trans. Knowl. Discov. Data* 15.2. ISSN: 1556-4681.

Saha, Swarnadeep, Prateek Yadav, and Mohit Bansal (2021). "multiPRover: Generating Multiple Proofs for Improved Interpretability in Rule Reasoning." In: *NAACL*.

Sakaguchi, Keisuke, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi (2021). "Winogrande: An adversarial winograd schema challenge at scale." In: *Communications of the ACM* 64.9, pp. 99–106.

Salazar, Julian, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff (July 2020). "Masked Language Model Scoring." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, pp. 2699–2712. DOI: 10.18653/v1/2020.acl-main.240. URL: https://aclanthology.org/2020.acl-main.240.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." In: *arXiv e-prints*, arXiv–1910.

Sap, Maarten, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi (2019). "Atomic: An atlas of machine commonsense for if-then reasoning." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01, pp. 3027–3035.

Sato, Naomi, Masaru Isonuma, Kimitaka Asatani, Shoya Ishizuka, Aori Shimizu, and Ichiro Sakata (2022). "Lexical Entailment with Hierarchy Representations by Deep Metric Learning." In: *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3517–3522.

Saxena, Apoorv, Aditay Tripathi, and Partha Talukdar (2020). "Improving multi-hop question answering over knowledge graphs using knowledge base embeddings." In: *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4498–4507.

Schick, Timo and Hinrich Schütze (2020). "Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05, pp. 8766–8774.

Schnabel, Tobias, Igor Labutov, David Mimno, and Thorsten Joachims (Sept. 2015). "Evaluation methods for unsupervised word embeddings." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Ed. by Lluís Màrquez, Chris Callison-Burch, and Jian Su. Lisbon, Portugal: Association for

Computational Linguistics, pp. 298–307. DOI: 10.18653/v1/D15-1036. URL: https://aclanthology.org/D15-1036.

Sekine, Satoshi (2008). "Extended Named Entity Ontology with Attribute Information." In: *LREC*.

Sekine, Satoshi, Akio Kobayashi, and Kouta Nakayama (2018). "Shinra: Structuring wikipedia by collaborative contribution." In: *Automated Knowledge Base Construction (AKBC)*.

Shang, Chao, Sarthak Dash, Md Faisal Mahbub Chowdhury, Nandana Mihindukulasooriya, and Alfio Gliozzo (2020). "Taxonomy construction of unseen domains via graph-based cross-domain knowledge transfer." In: *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pp. 2198–2208.

Shen, Jiaming, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T Vanni, Brian M Sadler, and Jiawei Han (2018). "Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion." In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2180–2189.

Shen, Yuxin, Zhao Li, Xin Wang, Jianxin Li, and Xiaowang Zhang (2021). "Datatype-aware knowledge graph representation learning in hyperbolic space." In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1630–1639.

Shin, Taylor, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh (2020). "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235.

Shwartz, Vered, Yoav Goldberg, and Ido Dagan (2016). "Improving Hypernymy Detection with an Integrated Path-based and Distributional Method." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2389–2398.

Sieg, Ahu, Bamshad Mobasher, Steve Lytinen, and Robin Burke (2004). "Using concept hierarchies to enhance user queries in web-based information retrieval." In: *Artificial Intelligence and Applications (AIA)*.

Snow, Rion, Daniel Jurafsky, and Andrew Ng (2004). "Learning syntactic patterns for automatic hypernym discovery." In: *Advances in neural information processing systems* 17.

Song, Mingyang, Yi Feng, and Liping Jing (2022a). "A preliminary exploration of extractive multi-document summarization in hyperbolic space." In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4505–4509.

— (2022b). "Hyperbolic Relevance Matching for Neural Keyphrase Extraction." In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5710–5720.
— (2023). "HISum: Hyperbolic Interaction Model for Extractive Multi-Document Summarization." In: *Proceedings of the ACM Web Conference 2023*, pp. 1427–1436.
Speer, Robyn, Joshua Chin, and Catherine Havasi (2017). "Conceptnet 5.5: An open multilingual graph of general knowledge." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1.
Su, Yusheng, Xu Han, Zhengyan Zhang, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun (2021). "Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models." In: *AI Open* 2, pp. 127–134.
Suchanek, Fabian M, Gjergji Kasneci, and Gerhard Weikum (2007). "Yago: a core of semantic knowledge." In: *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706.
Sun, Tianxiang, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuan-Jing Huang, and Zheng Zhang (2020a). "CoLAKE: Contextualized Language and Knowledge Embedding." In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3660–3670.
Sun, Yu, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. (2021). "Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation." In: *arXiv preprint arXiv:2107.02137*.
Sun, Yu, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu (2019). "Ernie: Enhanced representation through knowledge integration." In: *arXiv preprint arXiv:1904.09223*.
Sun, Yu, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang (2020b). "Ernie 2.0: A continual pre-training framework for language understanding." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 05, pp. 8968–8975.
Sun, Zhiqing, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang (2018). "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space." In: *International Conference on Learning Representations*.
Sussna, M (1993). "Word sense disambiguation for free-text indexing using a massive semantic network." In: *Proc. CIKM'93*, pp. 67–74.
Takeoka, Kunihiro, Kosuke Akimoto, and Masafumi Oyamada (Nov. 2021). "Low-resource Taxonomy Enrichment with Pretrained Lan-

guage Models." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 2747–2758. DOI: 10.18653/v1/2021.emnlp-main.217. URL: https://aclanthology.org/2021.emnlp-main.217.

Talmor, Alon, Yanai Elazar, Yoav Goldberg, and Jonathan Berant (2020a). "oLMpics-on what language model pre-training captures." In: *Transactions of the Association for Computational Linguistics* 8, pp. 743–758.

Talmor, Alon, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant (2019). "CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158.

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

Tenney, Ian, Dipanjan Das, and Ellie Pavlick (2019). "BERT Rediscovers the Classical NLP Pipeline." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601.

Tomkins, Silvan S (1978). "Script theory: differential magnification of affects." In: *Nebraska symposium on motivation*. University of Nebraska Press.

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

Vendrov, Ivan, Ryan Kiros, Sanja Fidler, and Raquel Urtasun (2015). "Order-embeddings of images and language." In: *arXiv preprint arXiv:1511.06361*.

Verga, Pat, Haitian Sun, Livio Baldini Soares, and William Cohen (2021). "Adaptable and interpretable neural memoryover symbolic knowledge." In: *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 3678–3691.

Vincent, Pascal, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol (2008). "Extracting and composing robust

features with denoising autoencoders." In: *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103.

Voita, Elena and Ivan Titov (2020). "Information-Theoretic Probing with Minimum Description Length." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 183–196.

Vrandečić, Denny and Markus Krötzsch (2014). "Wikidata: a free collaborative knowledgebase." In: *Communications of the ACM* 57.10, pp. 78–85.

Vu, Tu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant (2022). "Overcoming Catastrophic Forgetting in Zero-Shot Cross-Lingual Generation." In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9279–9300.

Vulić, Ivan, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen (2017). "Hyperlex: A large-scale evaluation of graded lexical entailment." In: *Computational Linguistics* 43.4, pp. 781–835.

Vulić, Ivan and Nikola Mrkšić (2018). "Specialising Word Vectors for Lexical Entailment." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1134–1145.

Vulić, Ivan, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš (2021). "LexFit: Lexical fine-tuning of pretrained language models." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5269–5283.

Vulić, Ivan, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen (2020). "Probing Pretrained Language Models for Lexical Semantics." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7222–7240.

Wang, Chengyu, Xiaofeng He, and Aoying Zhou (2017). "A short survey on taxonomy learning from text corpora: Issues, resources and recent advances." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1190–1203.

Wang, Kexin, Nils Reimers, and Iryna Gurevych (Nov. 2021). "TSDAE: Using Transformer-based Sequential Denoising Auto-Encoderfor Unsupervised Sentence Embedding Learning." In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 671–688.

Wang, Meng, Haofen Wang, Guilin Qi, and Qiushuo Zheng (2020). "Richpedia: a large-scale, comprehensive multi-modal knowledge graph." In: *Big Data Research* 22, p. 100159.

Wang, Peifeng, Filip Ilievski, Muhao Chen, and Xiang Ren (2021a). "Do Language Models Perform Generalizable Commonsense Inference?" In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3681–3688.

Wang, Xiaozhi, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang (2021b). "KEPLER: A unified model for knowledge embedding and pre-trained language representation." In: *Transactions of the Association for Computational Linguistics* 9, pp. 176–194.

Wang, Zhen, Jianwen Zhang, Jianlin Feng, and Zheng Chen (2014). "Knowledge graph embedding by translating on hyperplanes." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 28. 1.

Weber, Leon, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel (2019). "NLProlog: Reasoning with Weak Unification for Question Answering in Natural Language." In: *CoRR* abs/1906.06187. arXiv: 1906.06187. URL: http://arxiv.org/abs/1906.06187.

Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. (2022a). "Emergent Abilities of Large Language Models." In: *Transactions on Machine Learning Research*.

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

Wei, Xiaokai, Shen Wang, Dejiao Zhang, Parminder Bhatia, and Andrew Arnold (2021). "Knowledge enhanced pretrained language models: A compreshensive survey." In: *arXiv preprint arXiv:2110.08455*.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online:

Association for Computational Linguistics, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wu, Wentao, Hongsong Li, Haixun Wang, and Kenny Q Zhu (2012). "Probase: A probabilistic taxonomy for text understanding." In: *Proceedings of the 2012 ACM SIGMOD international conference on management of data*, pp. 481–492.

Wu, ZHIBIAO (1995). "VERB SEMANTICS AND LEXICAL SELECTION." In.

Xu, Bo, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Yanghua Xiao (2017). "CN-DBpedia: A never-ending Chinese knowledge extraction system." In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, pp. 428–438.

Xu, Hongyuan, Yunong Chen, Zichen Liu, Yanlong Wen, and Xiaojie Yuan (July 2022). "TaxoPrompt: A Prompt-based Generation Method with Taxonomic Context for Self-Supervised Taxonomy Expansion." In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Ed. by Lud De Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, pp. 4432–4438. DOI: 10.24963/ijcai.2022/615. URL: https://doi.org/10.24963/ijcai.2022/615.

Xu, Peng and Denilson Barbosa (2018). "Neural Fine-Grained Entity Type Classification with Hierarchy-Aware Loss." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 16–25.

Yamada, Ikuya, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto (2020a). "Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 23–30.

Yamada, Ikuya, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto (2020b). "LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6442–6454.

Yang, Jian, Gang Xiao, Yulong Shen, Wei Jiang, Xinyu Hu, Ying Zhang, and Jinghui Peng (2021). "A Survey of Knowledge Enhanced Pre-trained Models." In: *CoRR* abs/2110.00269. arXiv: 2110.00269. URL: https://arxiv.org/abs/2110.00269.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le (2019). "XLNet: Generalized

Autoregressive Pretraining for Language Understanding." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

Yao, Liang, Chengsheng Mao, and Yuan Luo (2019). "KG-BERT: BERT for Knowledge Graph Completion." In: *arXiv e-prints*, arXiv–1909.

Yu, Donghan, Chenguang Zhu, Yiming Yang, and Michael Zeng (2022). "Jaket: Joint pre-training of knowledge graph and language understanding." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 10, pp. 11630–11638.

Yu, Yong, Xiaosheng Si, Changhua Hu, and Jianxun Zhang (2019). "A review of recurrent neural networks: LSTM cells and network architectures." In: *Neural computation* 31.7, pp. 1235–1270.

Zellers, Rowan, Yonatan Bisk, Roy Schwartz, and Yejin Choi (2018). "SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii. Brussels, Belgium: Association for Computational Linguistics, pp. 93–104. DOI: 10.18653/v1/D18-1009. URL: https://aclanthology.org/D18-1009.

Zhang, Chengkun and Junbin Gao (2021). "Hype-han: Hyperbolic hierarchical attention network for semantic embedding." In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 3990–3996.

Zhang, Hongming, Daniel Khashabi, Yangqiu Song, and Dan Roth (2021). "TransOMCS: from linguistic graphs to commonsense knowledge." In: pp. 4004–4010.

Zhang, Hongming, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung (2020). "ASER: A large-scale eventuality knowledge graph." In: *Proceedings of the web conference 2020*, pp. 201–211.

Zhang, Zhengyan, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu (2019). "ERNIE: Enhanced Language Representation with Informative Entities." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1441–1451.

Zhao, Wayne Xin, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. (2023). "A Survey of Large Language Models." In: *arXiv e-prints*, arXiv–2303.

Zhong, Jiwei, Haiping Zhu, Jianming Li, and Yong Yu (2002). "Conceptual graph matching for semantic search." In: *Conceptual Structures: Integration and Interfaces: 10th International Conference on Conceptual Structures, ICCS 2002 Borovets, Bulgaria, July 15–19, 2002 Proceedings 10*. Springer, pp. 92–106.

Zhong, Zexuan, Dan Friedman, and Danqi Chen (2021). "Factual Probing Is [MASK]: Learning vs. Learning to Recall." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5017–5033.

Zhu, Yudong, Di Zhou, Jinghui Xiao, Xin Jiang, Xiao Chen, and Qun Liu (2020). "HyperText: Endowing FastText with Hyperbolic Geometry." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1166–1171.

Zhu, Yukun, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). "Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books." In: *The IEEE International Conference on Computer Vision (ICCV)*.

Zuo, Xinyu, Haijin Liang, Ning Jing, Shuang Zeng, Zhou Fang, and Yu Luo (2022). "Type-enriched Hierarchical Contrastive Strategy for Fine-Grained Entity Typing." In: *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 2405–2417.