Constructing Knowledge Graphs with Language Models and Learning Hierarchies from Graphs using Probabilistic Topic Modeling

by

Yujia Zhang

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Knowledge graphs leverage a data model structured as a graph or topology to represent and manipulate data. Knowledge graphs, abbreviated as KGs, consist of interconnected factual statements, conceptualized as distinct entities referred to as the *subject* and *object*, linked by a specified relation known as the *predicate*. These graphs find applications in recommendation systems, logical reasoning, and question-answering mechanisms. They empower machines to comprehend the relationships between different entities and draw conclusions based on the structured information they encompass. Constructing, revising, and augmenting such KGs warrants particular scholarly attention.

KG construction is fundamental to organizing and representing structured knowledge from unstructured text data. The KGs can be constructed more effectively with advanced language models with substantial computational capabilities. The models' effectiveness lies in understanding textual data, extracting facts, and synthesizing the content. Our study focuses on evaluating the capacity of these models to identify entities and relationships that contain contextual semantics. Through the utilization of these capabilities, the quality and comprehensiveness of KGs can be improved. Moreover, incorporating sophisticated methods such as transformers and their fine-tuning enables these models to adapt to specific domains, consequently enhancing the relevance and accuracy of the extracted knowledge.

The hierarchical analysis of knowledge graphs (KGs) is instrumental in uncovering the latent structures inherent in knowledge base data. Drawing inspiration from probabilistic topic modeling, which analyzes text corpora by identifying latent topics that represent the underlying themes and content patterns in documents, our research aims to adapt and

extend these analytical frameworks for the hierarchical exploration of KGs. Specifically, models are introduced within a nonparametric and probabilistic context, offering adaptability in comprehending the arrangement of the hierarchy. We have adapted the Hierarchical Latent Dirichlet Allocation algorithm and the Nested Hierarchical Dirichlet Process to construct the models. We evaluate these models quantitatively and qualitatively by analyzing the topics and distributions of words that define the hierarchical structure of complex KGs. By doing so, we aim to enhance our understanding of the intricate connections and dependencies within KGs, facilitating more robust and scalable knowledge representation. Furthermore, our research seeks to identify potential improvements in the algorithms used for hierarchical analysis, ultimately contributing to more efficient methods for managing and utilizing large-scale knowledge bases. This approach provides deeper insight into the structural dynamics of KGs and paves the way for semantic search, ontology development, and automated reasoning.

Preface

The research in this thesis represents the culmination of years of dedicated research, experiments, and collaboration in the knowledge graphs construction and hierarchy analysis. This research reflects not only my personal academic growth but also the invaluable guidance and support of my supervisor Prof. Marek Reformat.

Chapter 3 of this thesis includes the materials accepted by the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024) workshop *Knowledge Graphs and Large Language Models (KaLLM)*. As the principal author of this work, I conducted the design and execution of the experiments.

Chapter 4 of this thesis builds upon research published in the 19th European Semantic Web Conference (ESWC 2022). As the principal author of this work, I conducted the design of model and evaluation of experiments.

Chapter 5 of this thesis contains unpublished experimental methodology that we intend to submit to Information Processing and Management. As the principal author of this work, I conducted the design of model and evaluation of experiments.

Acknowledgements

First of all, I want to express my sincerest gratitude to my supervisor, Prof. Marek Reformat, for his unwavering support, invaluable guidance, and scholarly insight throughout this doctoral journey. During the past five years, his encouragement and insight have significantly contributed to my academic growth. I am deeply grateful for his unwavering support. I also wish to thank the other members of my supervisory committee, Prof. Petr Musilek and Prof. Witold Pedrycz, for their thorough reviews of my work and their constructive feedback.

I am profoundly thankful to my collaborators and colleagues, whose contributions have enriched this work immeasurably. Special thanks to Marcin Pietrasik, Wenjie Xu, Zheng Yu, Tyler Sadler, Mohammad Reza Taesiri for their dedication, expertise, and collaborative spirit. Their support and innovative ideas have been pivotal in advancing this research, and I am deeply appreciative of their commitment to our shared goals.

I am also indebted to my family for their unwavering support, encouragement, and understanding throughout this journey. Their love, patience, and belief in me have been a constant source of inspiration and motivation. Without their steadfast presence and sacrifices, this achievement would not have been possible.

Lastly, I extend my heartfelt thanks to all researchers, educators, and institutions whose work has paved the way for mine. It is through their collective efforts that the boundaries of knowledge continue to expand, and I am honored to contribute to this ongoing pursuit. Their foundational research has provided the critical foundations for my work, and I am grateful for the opportunity to build upon their legacy.

Table of Contents

List of Tables

List of Figures

List of Symbols

Knowledge Graphs

$\mathcal{E}$ Entities Sets

$\mathcal{G}$ Knowledge Graphs

$\mathcal{O}$ Objects Sets

$\mathcal{P}$ Predicates Sets

$\mathcal{S}$ Subjects Sets

Large Language Models

K Keys Matrix

Q Query Matrix

V Values Matrix

Hierarchical Topic Modelling for KGs

α Parameter of Dirichlet Prior for Topic Distribution over Levels

β^p Predicates Topic Distribution

β^t Tags Topic Distribution

η_p Parameter of Dirichlet Prior for Predicates

η_t Parameter of Dirichlet Prior for Tags

γ Parameter of nested Chinese Restaurant Process

θ_i Topic Distribution over Levels in the Tree

$p_{i,j}$ Specific Predicates

$t_{i,j}$ Specific Tags

$z_{i,j}$ Level Indicators shared among Predicated and Tags

c_i Paths Sampled for Each Subject

nested Hierarchical Dirichlet Process for KGs

$\theta_{\boldsymbol{i}_l,j}$ An Atom θ_{i_l} at Level l in Dirichlet Process

$\theta_{\boldsymbol{i}}$ Topic Probability Vector for Node $\boldsymbol{i}$

$c_{s,n}$ Topic Indicator for Word n in Subject s,

G_0 Continuous Base Distribution

$G_{\boldsymbol{i}_l}$ Stick Breaking Construction to get the Hierarchy

$G^s_{\boldsymbol{i}_l}$ Subject Specific Distribution over Paths in the Global Shared Tree

$q(\theta_{\boldsymbol{i}})$ Prior Distribution for Topic Probability

$q(c_{s,n})$ Prior Distribution for Topic Indicator

$q(U_{s,\boldsymbol{i}})$ Prior Distribution for Switch Probability

$q(V_{\boldsymbol{i},j})$ Prior Distribution for Global Stick Proportion

$q(V_{\boldsymbol{i},j})$ Prior Distribution for Stick Proportion in Local DP

$U_{s,\boldsymbol{i}}$ Switch Probability for Node $\boldsymbol{i}$

$V^{(s)}_{\boldsymbol{i},j}$ Stick Proportion for Local DP for Node $\boldsymbol{i}$

$V_{\boldsymbol{i},j}$ Stick Proportion for the Global DP for node $\boldsymbol{i}$

$z^{(s)}_{\boldsymbol{i},j}$ Index Pointer to Atom in Global DP $G_{\boldsymbol{i}}$ for jth break in $G^{(s)}_{\boldsymbol{i}}$

Abbreviations

ARI Adjusted Rand Index.

BTQ Branch Topic Quality.

EE Exact Evaluation.

GPT Generative Pre-trained Transformers.

hLDA hierarchical Latent Dirichlet Allocation.

HTQ Hierarchical Topic Quality.

KGs Knowledge Graphs.

LDA Latent Dirichlet Allocation.

LLMs Large Language Models.

LOL nested Hierarchical Dirichlet Process.

LORA Low-Rank Adaptation.

LTQ Level Topic Quality.

MCMC Markov Chain Monte Carlo.

nCRP nested Chinese Restaurant Process.

NLP Nature Language Processing.

NMI Normalized Mutual Information.

PaLM Pathways Language Model.

PE Partical Evaluation.

PEFT Parameter-Efficient Fintuning.

PMI Pointwise Mutual Information.

RDF Resource Description Framework.

SE Strict Evaluation.

TE Type Evaluation.

XML Extensible Markup Language.

Chapter 1

Introduction

The thesis presents the outcomes of the research on knowledge graphs, particularly their construction and inducing hierarchical structures based on them. The work's results are documented in three papers, forming three thesis chapters. Additionally, the thesis includes the motivation and objectives of the research, emphasizing its contributions to the scientific community. It also provides relevant background information and related works, positioning the research at the intersection of knowledge graphs, large language models, and probabilistic topic models. The thesis concludes with potential directions for future research.

1.1 Motivation

The development of the internet and digital platforms has led to a massive growth of web data repositories. Such a high volume of web data poses formidable challenges for computer systems and human users. Machine systems have difficulty computing efficiency and data storage while processing and analyzing volumes of data. To successfully harness invaluable insights concealed within a vast amount of web data, continuous improvement of machine learning and data processing techniques is necessary. Understanding natural language, ambiguity resolution, and context comprehension present significant obstacles for these algorithms. On the other hand, human users experience cognitive overload due to the amount of information available. Additionally, it is becoming increasingly difficult

to identify the reliability of data sources, thereby fostering the spread of false information. Consequently, addressing the mentioned challenges requires cutting-edge research to provide machines and humans with the skills and approaches to navigate the data-rich environment successfully.

Knowledge graphs (KGs) represent a novel knowledge format founded on graph theory in a domain that provides an intuitive way to understand and navigate the world. A KG is a data structure representing real-world entities and the relationships between them in the format of a triple, e.g., <head entity, relation, tail entity> or <subject, predicate, object> [1]. In precise terminology, a KG is a directed (mostly acyclic) graph (or DAG). A KG can have cyclic or transitive relationships, but most are subsumptive or representing inverse relations. KGs can represent different types of data, including facts, opinions, and events. For example, a KG in an organization is a hierarchical data structure that describes the relationships between entities and their attributes, such as customers, products, employees, and suppliers.

Popular, open to the public, KGs such as WikiData [2], Freebase [3], YAGO [4], and DBpedia [5] have been widely used to support a variety of applications such as search engines [6], recommendation systems [7], and question answering mechanisms[8]. Not only in the research fields of Computer Science, Artificial Intelligence, and the Semantic Web but also in some real-world products, like Google's Knowledge Graph and Microsoft's Satori, KGs have demonstrated a substantial ability to offer more effective services.

However, KGs need to be revised and updated with new information, while old data must be discarded. Therefore, constructing triples in the structured format <subject, predicate, object> is necessary to keep KGs current. Building KG is a challenging task. It is primarily supervised and mandates that humans extract all facts from plain text, connect them and build a graph out of those facts. As a result, KG construction and improvement are time-consuming and costly. One of the proposed aims is to develop a model to extract facts with semantic relationships automatically from plain text. It should provide helpful information

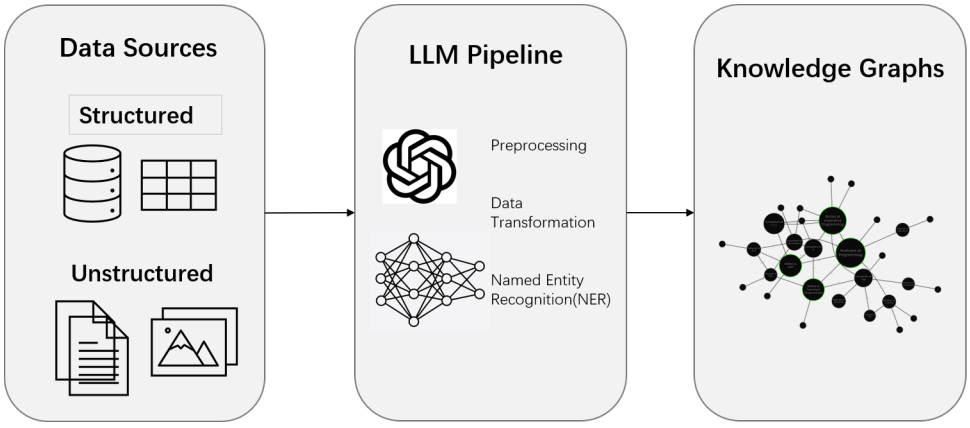


Figure 1.1: Knowledge Graphs Construction Pipeline

for the proposed here hierarchy analysis.

Large Language Models (LLM) are essential for natural language processing (NLP). These models [9], which are based on deep learning techniques and exhibit outstanding language generation and understanding capabilities, enable a wide range of applications in KG construction, text generation, translation, sentiment analysis, and question-answering systems. An outline of the pipeline for constructing a knowledge graph is shown in Figure 1.1.

Learning hierarchies from KGs is motivated by various factors. The most essential benefit of hierarchical structures is that we, as humans, naturally categorize and order information to make it easier to comprehend and recall. The hierarchical organization allows us to see the connections between concepts and ideas. For example, a hierarchical KG integrates information in a hierarchical structure and illustrates parent-child relationships between entities. Such a graph organizes the entities into higher-level categories and subcategories, making it possible to see the knowledge hierarchy. Thanks to hierarchical KGs, knowledge can be categorized and organized, efficiently navigated, and supported by semantic inference and reasoning. A common characteristic of KGs is the semantic hierarchy. The triple <England, /location/location/contains, Pontefract/Lancaster> found in Freebase [3], where "Pontefract/Lancaster" is at a lower semantic level than "England" in the hierarchy, is an example of hierarchical relation. DBpedia, on the other hand, provides ontology, i.e., hier-

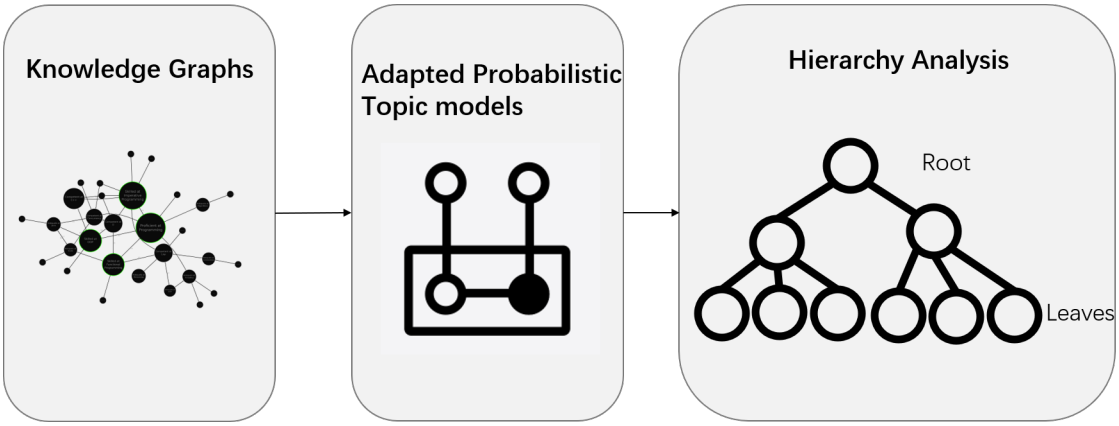


Figure 1.2: Knowledge Graphs Hierarchy Analysis Pipeline

archically organized information about classes and concepts. Although some studies have considered hierarchy structures [10], they typically call either extra information or a different method to collect the hierarchy information. Finding a process that can automatically and effectively build the semantic hierarchy is thus still challenging.

There are several challenges associated with extracting hierarchy KGs. First of all, for granularity and flexibility, the stiffness and granularity of the hierarchical structure might restrict hierarchical information. To overcome that, graphs' entities must fit into established categories and subcategories, which may only sometimes match the data's specifics entirely. It might not be easy to balance the need for adaptability to accept various relationships and maintain a consistent hierarchical structure. Secondly, for scalability and upkeep, maintaining the hierarchical structure can be challenging as the knowledge network becomes larger and more complicated. In particular, when numerous entities and relationships are involved, updating or changing the hierarchy may entail much work. Practical challenges can arise while ensuring data consistency and integrity while scaling the graph. Last but not least, for Semantic ambiguity, the interpretation and assignment of entities to specific categories and subcategories are the foundation of the hierarchy for KG. However, semantic ambiguity can appear when an entity has several legitimate classification possibilities or when an entity spans several categories. Resolving such semantic ambiguity can be difficult and require further context or domain-specific information.

To overcome these issues, we are investigating novel and manageable methods for extracting hierarchical information from KGs. One possible strategy is to incorporate probabilistic topic algorithms that can extract the latent hierarchy structure of KGs. These algorithms can help reveal hidden relationships and dependencies within the KGs based on statistical analysis, shedding light on the underlying organizational principles governing interconnected entities. Figure 1.2 shows the pipeline of knowledge graphs hierarchy analysis. Another is to employ more adaptable language models that allow longer-term fine-tuning, customization, and modification.

1.2 Objectives

The research topics addressed in the thesis can be framed as three objectives.

The first objective – *Knowledge Graph Construction based on Large Language Models* – is to prepare high-quality data with semantic relationships. The aim is to organize and extract structured knowledge from unstructured textual information. The intention is to apply state-of-the-art large language models to process text, identify entities and relationships between them, and transform them into a structured KG representation. The generated KG, compromised by factual triples, will be evaluated using Precision, Recall, and F1 metrics.

The second objective – *Hierarchical Topic Modeling for Knowledge Graphs* – is to develop a non-parametric hierarchical generative model for KGs that draws inspiration from probabilistic methods used in topic modeling. The goal is to discover the latent probability distributions of a KG and organize its elements into a tree of abstract topics. The goal is to develop a method to perform a hierarchical clustering of knowledge graph subjects and learn membership distributions of predicates and entities to topics. Three standard datasets should be used to evaluate the proposed approach quantitatively and qualitatively.

Ultimately, the third objective – *Construction of Topic Hierarchy with Subtree Representation for Knowledge Graphs* – is to develop a non-parametric probabilistic model for hierarchical clustering of KGs. The model should uncover the latent subject-specific distri-

butions on paths within the hierarchy and a subtree for each subject. An entire tree should be a collection of local trees representing each subject. The method should provide the opportunity to identify cross-thematic topics, while keeping individual topics for subjects in separate subtrees. Therefore, the developed method is intended to cluster subject entities, corresponding predicates, and object entities and deliver their distributions over subtrees. The aim is to evaluate the model on four semantically real-world datasets. It will be essential to perform the qualitative assessment of the induced hierarchy.

1.3 Outline

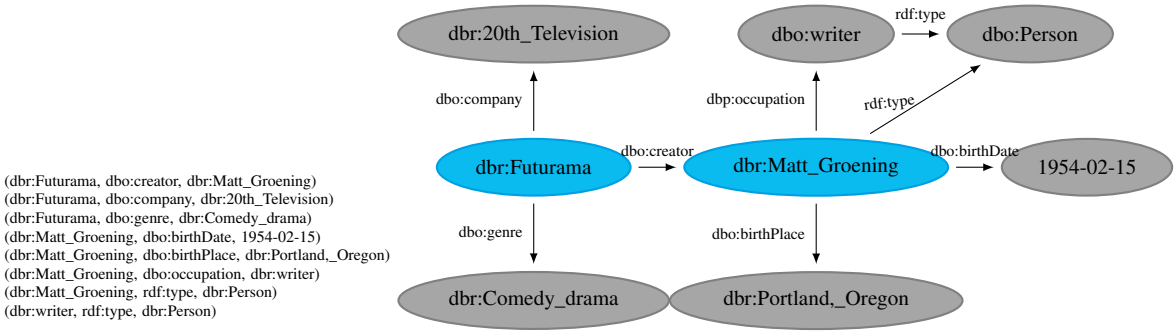
After the introductory chapter, this thesis continues with a brief overview of the necessary preliminaries and background information in Chapter 2. Chapter 3 presents the knowledge graph construction methods and experiments, it is the content of the paper *Fine-tuning Language Models for Triple Extraction with Data Augmentation*. The subject clustering based on probabilistic topic modeling from paper *Hierarchical Topic Modelling for Knowledge Graphs* is included in Chapter 4. Chapter 5 comprises of the paper *Construction of Topic Hierarchy with Subtree Representation for Knowledge Graphs*. Finally, this thesis is summarized and future work is discussed in Chapter 6.

Chapter 2

Background

2.1 Knowledge Graphs

There are two definitions of knowledge graph in the survey paper[1]: A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge. A knowledge graph is a multi-relational graph composed of entities and relations which are regarded as nodes and different types of edges, respectively.



(a) Factual Triples (b) Knowledge Graph

Figure 2.1: Factual Triples and Knowledge Graph

Following previous literature, a knowledge graph can be defined as $\mathcal{G} = \{\langle s, p, o\rangle \in \mathcal{E} \times \mathcal{P} \times \mathcal{E}\}$ where $\langle s, p, o\rangle$ is a triple, $\mathcal{E}$ is the set of entities in $\mathcal{G}$, and $\mathcal{P}$ is the set of predicates in $\mathcal{G}$. KG comprises real-world facts that can be represented as triples $\langle s, p, o\rangle$, where s, p, and o stand for the subject entity, predicate, and object entity, respectively. The examples of KGs are illustrated in Figure 2.1. In this example, the prefixes dbr, dbo, rdf, and $rdfs$ are commonly used in DBpedia to provide an organized and uniform method of

describing information and relationships between resources. The "dbr" stands for DBpedia Resource and is used to denote specific resources within the DBpedia dataset. The name or identifier of the resource usually follows it. The "dbo" stands for "DBpedia Ontology," and it is used to denote different classes or categories of resources in the DBpedia ontology. Usually, the class or type name comes afterward.

In my thesis, a subject can be described by all its predicate-object pairs $\langle p, o \rangle$, name it tags. From this view, each subject, s_i is annotated by its related tags, $t_j \in \mathcal{T}_i$, here $\mathcal{T}_i$ is the set of tags. The set of all subjects is denoted as $\mathcal{S} \subseteq \mathcal{E}$ which means it is a subset of all entities. Tags, denoted as $t := \langle p, o \rangle$, belong to the set of all tags, name it vocabulary, $\mathcal{V}$ whicn means $\mathcal{T}_i \subseteq \mathcal{V}$. For example in Figure 2.1, the entitiy of *dbr:Futurama* is described by the tags:$\langle$*dbo:creator, dbr:Matt_Groening*$\rangle$, $\langle$*dbo:company, dbr:20th_Television*$\rangle$, and $\langle$*dbo:genre, dbr:Comedy_drama*$\rangle$.

2.1.1 Resource Description Framework

The Resource Description Framework (RDF) [1] serves as a fundamental framework for the management of metadata; it facilitates interoperability among applications that transmit machine-readable information across the Web developed by the World Wide Web Consortium [2]. RDF focuses on providing mechanisms that support the automated handling of Web resources. This framework can be employed across various domains; for instance, in resource discovery to enhance search engine functionalities, in cataloging to articulate the content and interrelations of materials found on specific Websites, pages, or digital libraries.

Representation of RDF metadata [3] as well as a syntax for the encoding and transmission of this metadata in a manner that optimizes the interoperability of independently developed Web servers and clients [11, 12]. The syntax proposed herein utilizes the Extensible

[1] https://www.w3.org/RDF
[2] https://www.w3.org/
[3] https://www.w3.org/TR/PR-rdf-syntax/Overview.html

Markup Language [XML]: one of the objectives of RDF is to enable the specification of semantics for data grounded in XML in a standardized and interoperable format. RDF and XML are mutually supportive: RDF constitutes a model of metadata and only indirectly addresses many encoding challenges that transportation and file storage entail (such as internationalization, character sets, etc.). For these challenges, RDF depends on the capabilities provided by XML. It is equally crucial to recognize that this XML syntax represents merely one of several possible syntaxes for RDF, and alternative methods for representing the same RDF data model may arise.

The RDF data model provides a syntax-agnostic framework for the representation of RDF expressions. This model is instrumental in assessing semantic equivalence.

The fundamental data model encompasses three categories of objects:

- Resources: In RDF expressions, all entities are referred to as resources. A resource may represent an entire webpage, a segment within a webpage, a collection of pages, or even an entity not directly accessible via the web, such as a printed book. Resources are identified by URIs, potentially supplemented by anchor IDs, enabling the identification of any conceivable entity.

- Properties: A property constitutes a particular aspect, characteristic, attribute, or relationship utilized to delineate a resource. Each property possesses a specific meaning, delineates its permissible values, identifies the types of resources it can describe, and clarifies its interrelations with other properties.

- Statements: An RDF statement consists of a resource, a property, and the property's value, known as the subject, predicate, and object, respectively. The object can be either another resource, identified by a URI, or a literal, such as a string or primitive datatype defined by XML. In RDF, a literal may include XML markup, but it is not further evaluated by the RDF processor. Specific syntactic rules dictate how markup within literals can be expressed.

In Figure 2.1, the description of *dbr: Matt_Groening* is written as below:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:dbo="http://dbpedia.org/ontology/"
         xmlns:dbp="http://dbpedia.org/property/"
         xmlns:dbr="http://dbpedia.org/resource/">

    <rdf:Description rdf:about="dbr: Matt_Groening">
        <rdf:type rdf:resource="dbo:Person"/>
        <dbo:occupation rdf:resource="dbo:Writer"/>
        <dbo:birthDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1954-02-15</dbo:birthDate>
        <dbo:birthPlace rdf:resource="dbr:Portland,_Oregon"/>
    </rdf:Description>

</rdf:RDF>
```

It comprises notations and formats for serializing triples and is built on the triple structure. It offers the fundamental building blocks for creating ontologies and outlining the connections between resources.

2.1.2 Ontologies

Ontologies represent structured aggregations of knowledge that delineate a specific domain and its constituent entities, arranged in a hierarchical framework characterized by sets that possess shared attributes. In paper [13], ontology encompasses classes, relationships, constraints, and instances. The ontologies are employed for the purposes of knowledge representation, decision-making assistance, and modeling endeavors. In paper [14], the ontologies offer a standardized lexicon pertinent to a particular domain, enhance the application of synonyms, and aid in the resolution of syntactic ambiguities. In paper [15], the implementation, sharing, and reuse of ontologies are notably adaptable, particularly when integrated with web technologies and applications. They facilitate a non-formal articulation of knowledge, rendering them appealing to a diverse audience, irrespective of their programming proficiency or educational background.

The relationships incorporated within an ontology are not fixed a priori, thereby allowing for any real-world relationship to be logically defined and employed to interlink terms, thereby mirroring reality. There exist two fundamental relationship categories frequently utilized in numerous ontologies: *is_a* and *part_of* [16] [17]. The *is_a* relationship facilitates

straightforward, hierarchical connections among terms. For instance encompassing the terms "heart," "gills," and "brain". These terms are interconnected to the term "organ," which is subsequently linked to the term "anatomical structure," via an is_a hierarchy. Consequently, a query for "all mutants that influence zebrafish organs" could utilize the is_a relationships to yield results for any mutants displaying phenotypes in the heart, gills, or brain.

The *part_of* relationship serves to elucidate how the constituents of a biological system are organized. This may denote physical components where the brain is categorized into the hindbrain, forebrain, etc. It is noteworthy that each segment of the brain may be further subdivided, with the subcomponents related through a part_of relationship—for instance, the cerebellum is part_of the hindbrain, which in turn is part_of the entirety of the brain. Furthermore, a part_of relationship can also pertain to processes, such as those represented within the GO biological process ontology. For example, in that ontology, prophase, anaphase, metaphase, and telophase are all part_of the mitotic cell cycle.

Various relationship types can be appended to an ontology to enhance the knowledge it encompasses. The develops_from relationship, for instance, is employed to depict the developmental lineage of the organism and its components. Thus, the brain develops_from the neural tube.

The DBpedia ontology is the foundational structure for DBpedia [4]. Initially created in 2008 as a manually curated ontology from the most commonly used Wikipedia infoboxes, it has evolved into a successful crowdsourced project, resulting in a broad, yet somewhat superficial, cross-domain ontology. The DBpedia community continually refines the ontology schema and the mappings from infoboxes to the ontology through active participation in the DBpedia Mappings Wiki. Automated daily snapshots of these specifications are available via the DBpedia Databus, with the monthly DBpedia dataset release based on the most recent snapshot at the start of the dataset generation process.

[4] https://www.dbpedia.org/resources/ontology/

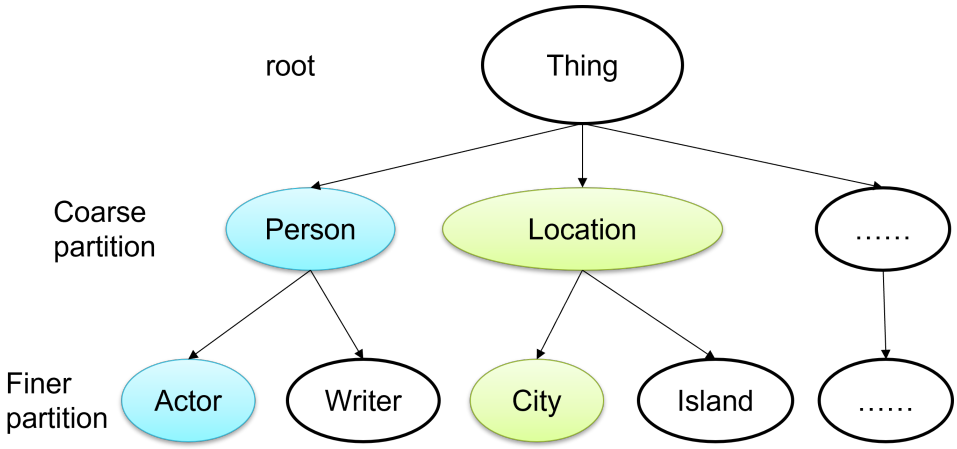


Figure 2.2: Ontology Example in DBpedia

The DBpedia ontology currently includes 768 classes organized within a subsumption hierarchy, featuring 3,000 distinct properties and approximately 4,233,000 instances. As illustrated in Figure 2.2, the ontology includes two Person classes and two Location classes. This hierarchical structure enhances the knowledge graph by enabling the inference that entities classified under the Actor class also belong to the Person class. Moreover, it provides a theoretical framework for understanding the relationships among various classes. For example, Actor and Writer are conceptually closer compared to the relationship between Actor and City.

2.2 Knowledge Graph Embedding

Knowledge graph embedding in Figure 2.3 maps knowledge graphs from the discrete graph space to a continuous vector space by translation distance models or semantic matching models. Such a representation is useful as it allows knowledge graphs to be easily integrated with common machine learning and deep learning methods. In the context of our work, knowledge graph embeddings may be used in conjunction with hierarchical clustering methods, allowing for benchmark comparison.

The process of embedding is outlined as follows in paper [18]. When provided with a Knowledge Graph (KG), the entities and relations are initially represented randomly in a

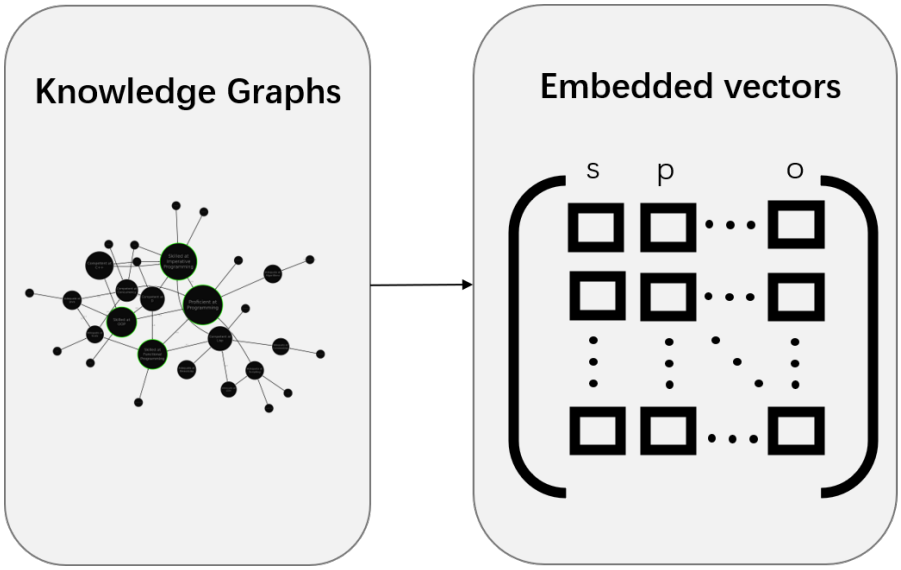


Figure 2.3: Knowledge Graph Embedding

lower-dimensional vector space, and a metric is established to assess the credibility of each fact triplet. During each iteration, the embedding vectors of entities and relations can be updated by maximizing the overall credibility of facts through an optimization algorithm. While numerous successful studies [19, 20] have been conducted in modeling relational facts, the majority of them are limited to training an embedding model using observed triplets data. Consequently, there is a growing body of research focusing on enhancing KG embedding models to be more generalized by incorporating supplementary information, such as entity types, relation paths, and textual descriptions.

2.3 Large Language Models

The development of large language models (LLMs) is a significant breakthrough in natural language processing (NLP). A wide range of applications in text generation, translation, sentiment analysis, and question-answering systems are made possible by these models[9], which are based on deep learning techniques and exhibit outstanding language generation and understanding capabilities. We could find the most latest LLMs in the leaderboard [21]. In this section, we initially introduce the prevalent architectures employed for LLMs. Upon

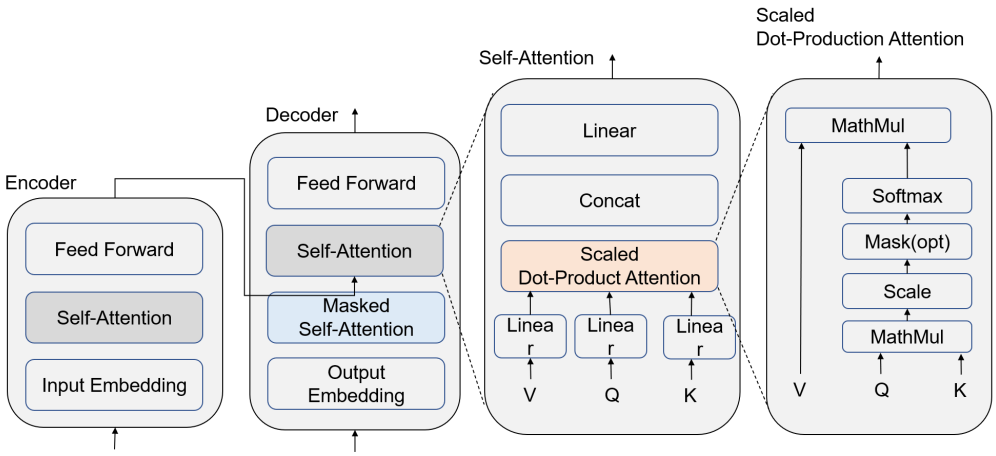


Figure 2.4: Transformer-based LLM with self-attention

the selection of the model architecture, the principal steps implicated in the training of an LLM encompass: data preprocessing (which involves collection, cleaning, and deduplication), tokenization, model pre-training (executed in a self-supervised learning manner), instruction tuning, and alignment.

2.3.1 LLM Architectures Transformer

The majority of large language models rely on transformer architectures [22], shown in Figure 2.4, which employ self-attentional mechanisms to extract contextual dependencies and relationships between words in a text. These models are trained on enormous volumes of text data from many sources, allowing them to obtain statistical patterns and representations of a language's semantics. The model consists of an encoder and a decoder. The encoder comprises six identical transformer layers consisting of two sublayers. The output of each sub-layer is represented as $\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$, where $\mathrm{Sublayer}(x)$ denotes the function executed by the sub-layer itself. Once the input embedding is fed in, it will pass through the self-attention layer and position-wise fully connected feed-forward layer. The decoder has masked self-attention for output embedding, and then the output from the encoder and the mask self-attention will be fed into the next transformer layers. Specifically, the Scaled Dot-Product Attention (SDPT) is mapping a query and key-value

pairs. So it generates the output as a weighted sum of the values, the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The input consists of queries and keys of dimension d_k, and values of dimension d_v. It computes the dot products of the query with all keys, divides each by $\sqrt{d_k}$, and applys a softmax function to obtain the weights on the values.

In practice, the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. We compute the matrix of outputs as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

LLMs can be categorized into three groups based on the architecture structure, which are encoder-only LLMs, decoder-only LLMs, and encoder-decoder LLMs [23] [24].

Encoder-only LLMs, in each phase, the attention layers have the capability to access all the words within the original sentence. The pre-training process for these models typically involves introducing some form of corruption to a provided sentence (like randomly masking certain words) and assigning the model the task of recovering or reconstructing the original sentence. Encoder models excel in tasks that demand a comprehensive grasp of the entire sequence, such as identifying sentence types, recognizing named entities, and answering extractive questions. BERT (Bidirectional Encoder Representations from Transformers) [25] is a notable single case of encoder model.

Decoder-only LLMs, on the other hand, enable the attention layers to only access words that precede a specific word in the sentence at each stage. These models are also known as autoregressive models. The pretraining approach for such models usually revolves around predicting the subsequent word (or token) in the sequence. Decoder-only models are particularly well-suited for tasks centered on text generation. Notably, the GPT models serve as a prominent example within this model category.

Encoder-Decoder LLMs, incorporating encoder and decoder, are sometimes referred to as sequence-to-sequence models. In each phase, the attention layers in the encoder can

scrutinize all the words in the original sentence, while the attention layers in the decoder are limited to words positioned prior to a specific word in the input. These models are typically pretrained based on the objectives of either encoder or decoder models, albeit with a more intricate approach. For instance, certain models undergo pretraining by replacing random text spans (which may consist of multiple words) with a single mask special word, with the subsequent objective being to predict the text represented by this mask word. Encoder-decoder models are especially effective for tasks involving the generation of new sentences conditioned on a given input, such as summarization, translation, or generative question answering.

The well-known three LLM families [24] are GPT, LLaMA, and PALM. The GPT Family, comprising Generative Pre-trained Transformers (GPT), are a series of decoder-only Transformer-based language models established by OpenAI. This family encompasses GPT-1, GPT-2, GPT-3, InstrucGPT, ChatGPT, GPT-4. While earlier iterations like GPT-1 and GPT-2 are available as open-source, more recent versions such as GPT-3 and GPT-4 [26] are proprietary and can only be interacted with through APIs.

The LLaMA Family, on the other hand, is a set of foundational language models introduced by Meta. Differing from GPT models, LLaMA models are open-source, meaning that model weights are made accessible to the academic community under a noncommercial license. Consequently, the LLaMA family is expanding rapidly as these models find extensive utility in numerous research endeavors aimed at creating improved open-source LLMs to rival proprietary ones or to engineer task-specific LLMs for critical applications. The initial series of LLaMA models [27] was introduced in February 2023, with parameter ranges from 7B to 65B. These models have undergone pre-training on vast amounts of tokens sourced from publicly accessible datasets. LLaMA adopts the transformer architecture of GPT-3, with several minor adjustments to its design, such as (1) utilizing a SwiGLU activation function in place of ReLU, (2) employing rotary positional embeddings rather than absolute positional embeddings, and (3) utilizing root-mean-squared layernormaliza-

tion instead of standard layer-normalization. The open-source LLaMA-13B model demonstrates superior performance compared to the proprietary GPT-3 (175B) model across various benchmarks, positioning it as a reliable baseline for further LLM studies.

The PaLM (Pathways Language Model) family, created by Google, introduced its initial model [28] in April 2022 and kept it private until March 2023. This model, a 540B parameter transformer-based LLM, was trained on a high-quality text corpus with 780 billion tokens covering various natural language tasks and scenarios. Utilizing the Pathways system, PaLM underwent pre-training on 6144 TPU v4 chips, enabling efficient training across multiple TPU Pods. Through scaling, PaLM achieved remarkable few-shot learning outcomes on numerous language understanding and generation benchmarks. Notably, PaLM540B surpassed state-of-the-art fine-tuned models in multi-step reasoning tasks and even matched human performance on the recent BIG-bench benchmark. The U-PaLM models, ranging from 8B to 540B, are continuously trained on PaLM using UL2R, a method for ongoing LLM training with UL2's denoiser-based objective. This approach is reported to yield approximately a 2x computational savings rate.

2.4 Probabilistic Topic Models

2.4.1 Probability Theory

Bayes' Theorem Many problems require calculating $p(\theta|X)$ given $p(X|\theta)$. Such problems can be addressed using Bayes' theorem, which describes the relationship between $p(\theta|X)$ and $p(X|\theta)$. Bayes' theorem can be expressed as follows:

$$p(\theta|X) = \frac{p(X|\theta)\, p(\theta)}{p(X)} \tag{2.2}$$

In this equation, $p(\theta)$ is known as the prior probability of θ because it is not influenced by the variable X. Correspondingly, $p(\theta|X)$ is known as the conditional or posterior probability of θ given x. The probability $p(X|\theta)$ is known as the likelihood, and $p(X)$ is known as the marginal or prior probability of X, typically used as a normalization factor. Therefore,

Bayes' theorem can also be expressed in the following manner:

$$\text{posterior probability} = \frac{\text{likelihood} \times \text{prior probability}}{\text{normalizing constant}}$$

Conjugate Distribution If a posterior probability $p(\theta|X)$ and a prior probability $p(\theta)$ of a random variable θ belong to the same distribution family, then $p(\theta|X)$ and $p(\theta)$ are known as the conjugate distribution, and $p(\theta)$ is known as the conjugate prior of the likelihood function $p(X|\theta)$. Conjugate distribution is a key characteristic of the exponential family.

Case 1: The beta distribution and binomial distribution are conjugate. Consider a random variable θ with a prior:

$$p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{2.3}$$

The likelihood function is:

$$p(X|\theta) = \binom{n}{k}\theta^{k}(1-\theta)^{n-k} \tag{2.4}$$

According to Bayes' theorem, the posterior is:

$$p(\theta|X) = \frac{\Gamma(\alpha+\beta+n)}{\Gamma(\alpha+k)\Gamma(\beta+n-k)}\theta^{\alpha+k-1}(1-\theta)^{\beta+n-k-1} \tag{2.5}$$

This result shows that the posterior $p(\theta|X)$ belongs to the beta distribution.

Case 2: The Dirichlet and multinomial distributions are conjugate. Consider a random variable θ with a prior in its Dirichlet form:

$$p(\theta; \alpha_1, \ldots, \alpha_K) = \frac{1}{B(\alpha)}\prod_{i=1}^{K}\theta_i^{\alpha_i-1} \tag{2.6}$$

where $B(\alpha)$ is the Beta function and $\alpha = (\alpha_1, \ldots, \alpha_K)$. The likelihood function is:

$$p(X|\theta) = \frac{n!}{X_1!\cdots X_k!}\theta_1^{X_1} \times \cdots \times \theta_k^{X_k} \tag{2.7}$$

According to Bayes' theorem, the posterior is:

$$p(\theta|X) = \frac{1}{B(\alpha+X)}\prod_{i=1}^{K}\theta_i^{\alpha_i+X_i-1} \tag{2.8}$$

This matches the form of a Dirichlet distribution Dirichlet$(\alpha_1 + X_1, \ldots, \alpha_K + X_K)$. This result shows that the posterior $p(\theta|X)$ belongs to the Dirichlet distribution. These two examples demonstrate that for a given likelihood function, the difficulty of finding a posterior probability depends on the selection of a prior distribution. Appropriate selection of the conjugate prior distribution and the likelihood function allows the posterior probability distribution to take the same form as the prior probability distribution, enabling a closed-form solution to be directly obtained.

Divergence Kullback-Leibler (KL) divergence is a common measure that defines the difference between two probability distributions $p(x)$ and $q(x)$:

$$D_{\mathrm{KL}}(p \parallel q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)} \tag{2.9}$$

This definition shows that the KL divergence cannot be negative. Moreover, if and only if $q(x)$ and $p(x)$ are equal, $D_{\mathrm{KL}}(p \parallel q) = 0$. Note that KL divergence is not symmetric when measuring the difference between two probability distributions, that is, $D_{\mathrm{KL}}(p \parallel q) \neq D_{\mathrm{KL}}(q \parallel p)$.

2.4.2 Common Probability Distributions

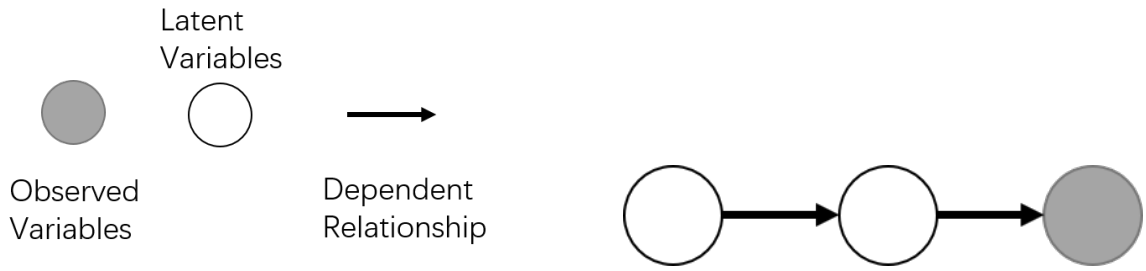
Beta Distribution The beta distribution is a continuous probability distribution family that operates within the range $[0, 1]$ and is characterized by two positive shape parameters, commonly referred to as α and β. It represents a particular scenario of the Dirichlet distribution, which is defined by only two parameters. Given that the Dirichlet distribution serves as the conjugate prior to the multinomial distribution, the beta distribution functions as the conjugate prior to the binomial distribution. Within Bayesian statistics, it can be interpreted as the posterior distribution of the parameter p in a binomial distribution following the observation of $\alpha - 1$ independent events with probability p, and $\beta - 1$ events with probability $1 - p$, assuming an initial uniform prior distribution for p.

Dirichlet Distribution The Dirichlet distribution, commonly represented as $Dir(\alpha)$, is a set of continuous multivariate probability distributions characterized by the positive real

vector α. It serves as the multivariate extension of the beta distribution and functions as the conjugate prior for both the categorical distribution and multinomial distribution within Bayesian statistics. Specifically, its probability density function expresses the confidence in the probabilities of rival events E_i, given that each event has been observed $\alpha_i - 1$ times.

Multinomial Distribution The multinomial distribution serves as a broader form of the binomial distribution, which represents the likelihood of achieving a certain number of "successes" in a series of n independent Bernoulli trials with equal success probabilities. In the context of a multinomial distribution, the categorical distribution is similar to the Bernoulli distribution, where each trial produces one of a set number k of possible outcomes with probabilities $p_1, ..., p_k$ (such that $p_i \geq 0$ for $i = 1, ..., k$ and the sum of all probabilities equals 1), and there are n independent trials. The random variables X_i denote the frequency of occurrence of outcome i across the n trials. A vector $U = (U_1, ..., U_k)$ conforms to a multinomial distribution with parameters n and p, where $p = (p_1, ..., p_k)$.

Graph plate notations Graph plate notations serve as a graphical method for under-



(a) Basic elements (b) A Bayesian network of three variables

Figure 2.5: Examples of Bayesian networks

standing topic models. Within graph plate notations, the presence of shaded and unshaded variables distinguishes between observed and latent variables, respectively. Arrows are utilized to denote conditional dependencies between variables, while plates represent repeated sampling, with the number of repetitions specified by the variable at the base. The symbols used in graph plate notations can be referenced in Figure 2.5.

2.4.3 Latent Dirichlet Allocation

Probabilistic topic models have become an effective technique for identifying latent semantic patterns in large document sets in the fields of natural language processing (NLP). In order to analyze text corpora to discover latent topics that can represent the underlying themes and content patterns in the documents, A probabilistic framework is proposed.

The assumption behind probabilistic topic models[29], which are generative models, is that each document contains a variety of latent topics, each of which is a probability distribution over words. The fundamental premise is that documents are generated via a two-step generative process: first, a topic is selected from the document's topic distribution, and then words are produced from the chosen topic's word distribution. This method enables the flexible representation of documents as a mixture of various topics, facilitating the discovery of latent thematic structures. Formally, the subsequent terms are defined. A word is considered the fundamental unit of discrete data, identified as an element from a vocabulary indexed by $1, ..., V$. Words are represented using unit-basis vectors with a sole component set to one, while all other components are set to zero. In this manner, denoting components with superscripts, the v_{th} word in the vocabulary is depicted by a V-vector w such that $w^v = 1$ and $w^u = 0$ for $u \neq v$. A document is outlined as a succession of N words signified by $w = (w_1, w_2, ..., w_N)$, where w_n represents the nth word in the sequence. A corpus signifies a compilation of M documents designated by $D = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$. It is our objective to devise a probabilistic model of a corpus that not only assigns a high likelihood to elements within the corpus but also assigns a high probability to other analogous documents.

The Latent Dirichlet Allocation (LDA)[30] topic model is one of the most used probabilistic topic models. In Figure 2.6, the workflow of LDA is illustrated. Documents are modeled by LDA as distributions over topics, while topics are modeled as distributions over words. In order to create a smoothing effect and manage the sparsity of the resulting topic assignments, it makes the assumption that the document-topic and topic-word distributions

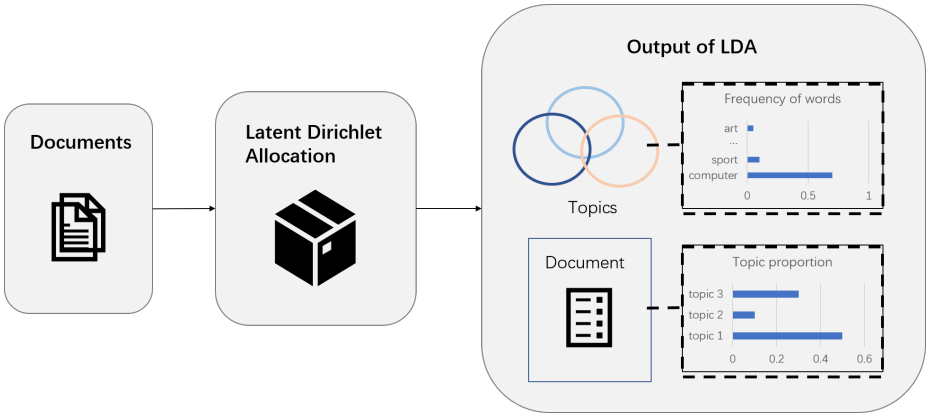


Figure 2.6: The Latent Dirichlet Allocation Workflow

are subject to a Dirichlet prior distribution. Given the observed data (documents), inference in LDA entails calculating the posterior distribution of latent variables (topics), generally using variational inference or Markov Chain Monte Carlo (MCMC) techniques.

Chapter 3

Fine-tuning Language Models for Triple Extraction with Data Augmentation

This chapter presents the work on fine-tuning language models for triple extraction with data augmentation and analysis of their performance compared to GPT-4. The implemented pipeline included the following steps:

- Data Augmentation and Preparation: We used diverse data augmentation techniques to fine-tune large language models (LLMs) for extracting triples (subject, predicate, object) from text. This process involved creating enlarged and enriched training WebNLG datasets.
- Model Training: Eleven models, each with seven billion parameters, were fine-tuned using different trainers from HuggingFace. These models were trained on the augmented datasets WebNLG and then benchmarked against ChatGPT and GPT-4.
- After analysing the performance of different LLMs on original WebNLG and augmentated WebNLG, we evaluated the best LLMs of 7b parameters, orca-mini-3, and the other base model of LLama, llama-2-13b, on the other real-world datasets like SKE, DocRed, FewRel, and KELM.
- Evaluation: The models were evaluated based on type, partial, exact, and strict accuracy in extracting triples. The evaluation showed that smaller, fine-tuned models could outperform the baselines set by GPT family models, including GPT-4.

Key Findings:

1. Effectiveness of Data Augmentation: The procedures to build various prompts and augment the datasets led to significant improvements in model performance.

2. Model Performance: Fine-tuned models with seven billion parameters performed better in triple extraction tasks than GPT-4, particularly for the WebNLG dataset.

3. Importance of High-Quality Data: The quality and size of the training data were critical in achieving high performance in triple extraction tasks.

Limitations:

1. Hallucination: The models occasionally hallucinated, especially on well-known topics like the Jeff Bezos Wikipedia article, often providing false information about Bezos's birthplace.

2. Looping issues: generating continuous output until reaching the token limit.

This work demonstrates that through effective data augmentation and fine-tuning, smaller LLMs of 7b parameters can achieve or exceed the performance of big LLMs like GPT-4 in specific tasks such as triple extraction.

This research was conducted by Yujia Zhang, Tyler Sadler, Mohammad Reza Taesiri, Wenjie Xu, Marek Reformat at the University of Alberta, and was accepted by the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024) workshop Knowledge Graphs and Large Language Models (KaLLM).
As the first author, I was responsible for formulating and executing the experiments, with oversight provided by Prof Marek Reformat.

Fine-tuning Language Models for Triple Extraction with Data Augmentation

Abstract:

Advanced language models with impressive capabilities to process textual information can more effectively extract high-quality triples, which are the building blocks of knowledge graphs. Our work examines language models' abilities to extract entities and the relationships between them. We use a diverse data augmentation process to fine-tune large language models to extract triples from the text. Fine-tuning is performed using a mix of trainers from HuggingFace and five public datasets, such as different variations of the WebNLG, SKE, DocRed, FewRel, and KELM. Evaluation involves comparing model output with test set triples based on several criteria, such as type, partial, exact, and strict accuracy. The obtained results outperform ChatGPT and even match or exceed the performance of GPT-4.

3.1 Introduction

Knowledge graphs (KGs) represent knowledge in a semantically rich and intuitive way, enabling one to better understand and utilize gathered information. A KG is a data structure representing real-world entities and the relationships between them in the format of a triple, e.g., $\langle$*head entity, relation, tail entity* $\rangle$ or $\langle$ *subject, predicate, object* $\rangle$ [1].

The majority of available knowledge is composed of unstructured textual data. The need to 'convert it' into a structured format via extracting entities and relationships between them drives the construction of KGs. Large language models, like ChatGPT or GPT-4, have a remarkable capacity for understanding and generating text. It makes them useful tools for automating the process of knowledge extraction from textual sources. They can capture nuances and complexities of language, allowing for a deeper comprehension of the text's meaning. Therefore, they can be employed to create KGs that accurately and fully capture

complicated semantic relations and the meaning of texts.

Extracting triples from texts poses several challenges [31]. Finding accurate and comprehensive entities and representative relationships from the text can be difficult, especially with various language usage, implicit references, and context-dependent interpretations. Additionally, processing and analyzing enormous quantities of text can be computationally demanding and resource-intensive.

Therefore, methods for capturing reliable contextual information are paramount for KG's growth and development. Advanced context-aware techniques must be developed to identify and separate contextual references, capture relationships, and identify implicit connections.

This work aims to tune large language models (LLMs) to perform triple extraction from text. We have conducted several experiments using various models and datasets of different quality and sizes. The construction of triples adhering to the DBpedia ontology format has been particularly interesting. The WebNLG dataset [32], predominantly using the DBpedia vocabulary for its entities and properties or emulating its ontological style, serves as the basis for our training data.

We have introduced a set of procedures to generate various prompts, instructing models about different processes related to triple extraction and understanding. This has led to the augmentation of the original WebNLG data and the creation of various versions of training datasets.

Eleven models, each with seven billion parameters, have been trained. Their efficacy has been evaluated in comparison with GPT-3.5 and GPT-4 on WebNLG. Additionally, we have preliminary assessed larger models with thirteen, thirty, and thirty-three billion parameters and trained them similarly.

The ultimate objective is to propose and illustrate a training methodology capable of elevating domain-specific models to or beyond the proficiency of leading-edge models.

The findings of the work that constitute our contributions are:

- the reasonably sized large language models, such as ones with seven billion parameters, can be successfully tuned to extract triples from text;
- the proposed procedures to build a variety of prompts lead to the generation of enlarged and enhanced (enriched with information that improves training) datasets;
- small, fine-tuned models can outperform the baselines set up by GPT family models: ChatGPT and GPT-4.
- high-quality data is essential for the triple generation task; many datasets in the triple extraction space focus on extracting only specific relationships from text rather than all possible relationships or do not follow particular vocabulary, like DBpedia ontology.

3.2 Related Work

In the field of triple extraction, LSTM is a conventional technique to explore. Seq2Rdf [33] employs an LSTM-based sequence-to-sequence model to map natural language text to RDF triples in one step, using pre-trained word and knowledge graph embeddings for initialization. However, it is limited to extracting single triples and cannot handle multi-triple extraction. The ChatIE framework [34] achieves zero-shot information extraction by promoting ChatGPT, without requiring any labeled data for training. It allows interactively querying the model to extract structured information piece by piece in a multi-turn conversational format. The ChatIE relies on LLM like ChatGPT which is not open source. The performance depends heavily on how well the prompts are engineered and provides many details.

The Head to Tail benchmark [35] provides a systematic way to evaluate how knowledgeable LLM are about facts in diverse domains(movies, books, academics). The benchmark is still limited in size and diversity compared to the vast world knowledge, 18k QA pairs may not comprehensively cover all entity types, relationships, and knowledge domains.

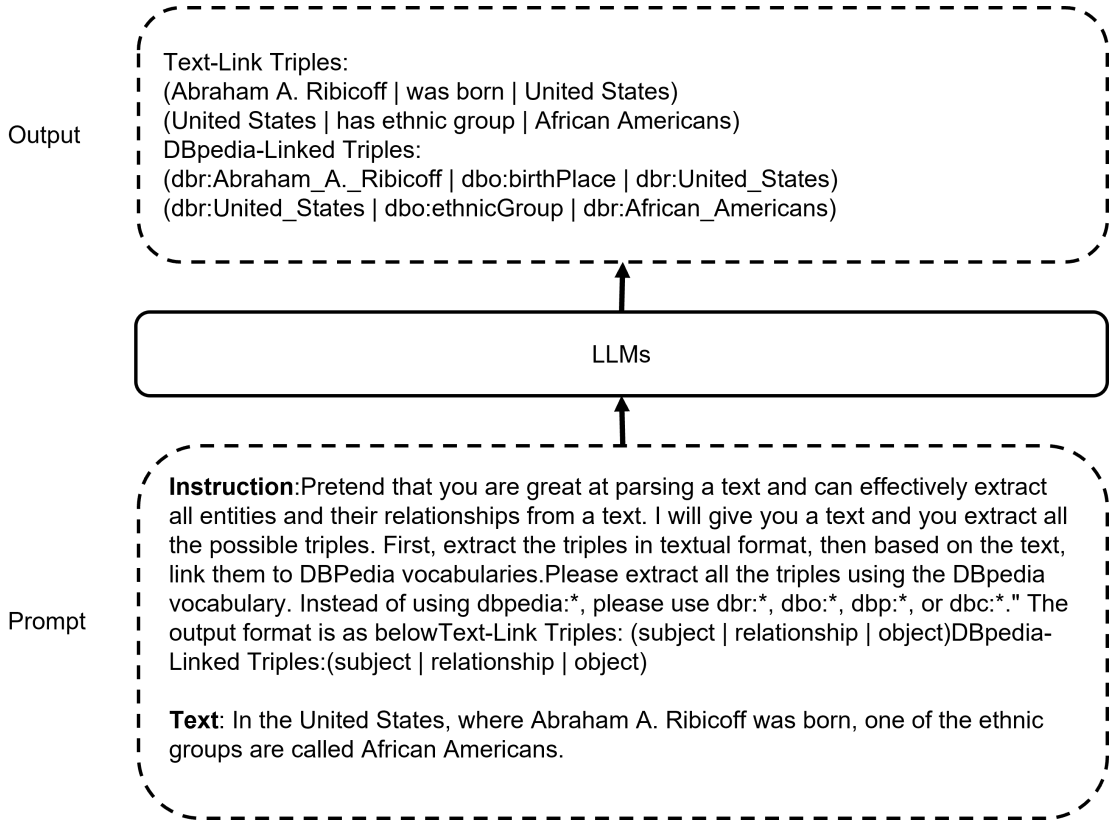


Figure 3.1: Example of triple extraction prompt workflow

Few-shot learning with GPT-3 [36] achieves state-of-the-art performance on standard relation extraction datasets, surpassing existing fully supervised models. Fine-tuning Flan-T5 on explanations generated by GPT-3 further enhances performance. Treating relation extraction as a text-generation task provides flexibility in expressing entities and relations. However, GPT-3 is opaque, not open source, and significantly costly.

3.3 Methods and Procedure

The paper focuses on extracting information from plain text. It is the task of building triples of the form $\langle subject, predicate, object \rangle$ based on the content of a sentence. Triple extraction is a domain-independent task. Two entities of a triple, i.e., subject and object, appear in the text, while a relation between these two entities is often deduced by 'understanding the meaning' of the sentence. All the components of a triple are extracted at the same time.

Llama2	You are an AI assistant who is an expert in knowledge graphs. You will be given an instruction and text. Generate a response to appropriately complete the instruction's request. {**instruction**}{**input**}{**output**}
LLongOrca	Below is an instruction that describes a task, paired with an input that provides further context. Write a output that appropriately completes the request. {**instruction**}{**input**}{**output**}
other models	### Instruction:{**instruction**} ### Input: {**input**} {**output**}

Table 3.1: Generic Prompt Template for Different Models.

Here is a more formal description of the task. Given a set of sentences $D := \{w_1, w_2, ..., w_n\}$, we want to obtain a set of facts built from and based on these sentences. Let this set be $Facts := \{fact_1, fact_2, ..., fact_n\}$, and each fact is denoted as $\langle s, p, o \rangle, s \in S, p \in P, o \in O$, where S, P, O are sets of subjects, predicates, and objects respectively.

These triples are the basic units of knowledge graphs, resulting from the development of the Semantic Web concept. The classes (types of entities) and properties (relationships and attributes) used to describe triples' components are defined using ontologies. One of the most well-known ontologies is the one used by DBpedia [5].

Within the DBpedia dataset, triples are generated and represented using the DBpedia ontology as the schema. This ontology consists of 320 classes organized into a subsumption hierarchy and 1650 distinct properties describing relations between them. The subsumption hierarchy is purposefully maintained relatively shallow, with a maximum depth of five to accommodate use cases where the ontology is traversed or visualized. Online browsing of the entire DBpedia ontology is available at [1].

[1] http://mappings.dbpedia.org/server/ontology/classes/

3.3.1 Datasets

WebNLG The WebNLG corpus [32] is made up of sets of triplets describing facts (entities and their relationships) and the matching facts expressed in natural language, in other words, text from which the triples are extracted. It includes 13,211 training data and 2,155 test data.

FewREL Few-Shot Relation Classification Dataset (FewRel)[37] composes 70,000 instances from Wikipedia and 100 relations. The dataset is divided into three subsets: training set (64 relations), validation set (16 relations), and test set (20 relations).

DocRED Document-Level Relation Extraction Dataset (DocRED) [38] is created from Wikipedia and Wikidata in relation extraction data. Annotated on 5,053 Wikipedia articles, DocRED comprises 132,375 entities and 56,354 relational facts. The collection offers large-scale distantly supervised data over 101,873 documents in addition to the human-annotated data.

KELM The English Wikidata KG and the corresponding natural text sentences make up the large-scale synthetic corpus known as KELM[39]. It has roughly 15 million artificially generated sentences produced by a refined T5 model. A list of triples of the format [subject, relation, object] is contained in each linearized KG graph in KELM. A subset of KELM, named KELM-sub, is used which contains 400,000/5,000 samples as train/test set.

SKE Baidu has released a Chinese dataset called SKE2019. The train set contains 194,747 sentences, whereas the validated set contains 21,639 sentences. SKE21 [40] has been released by manually labeling 1150 sentences from the test set with 2765 annotated triples. It contains 194,747 training data, 21,639 validation data, and 1,150 testing data. [2]

3.3.2 Large Language Models

LLMs like ChatGPT and GPT-4, pre-trained on a large-scale corpus, are composed of decoder modules based on the Transformer design, which incorporates a self-attention mech-

[2]http://ai.baidu.com/broad/download?dataset=sked

Data Format			
Data Augmentation (name)	Parts of prompt		Response
	instruction	**input**	**output**
Text2triples	Think of yourself as efficient in deconstructing a text and precisely identifying all the entities and their interrelations. I'll furnish you with a text and your job is to gather all potential triples, adhering to the pattern: (subject|relationship|object).	Sentence	Triples
Explanation	"Assume you're highly competent in scrutinizing a piece of text and successfully distilling all its entities along with their connections. I'll provide a text, and you are to extract every possible triplet, following the convention: (subject|relationship|object). Detail the entire process systematically."	Sentence	To extract triplets from the given text, we need to identify the subject, predicate, and object. Subject: "Aarhus Airport" Predicate: "cityServed" Object: "Aarhus, Denmark" The property "cityServed" is derived from the context of the sentence, where it implies that the airport serves the city of Aarhus. Therefore, here is the answer in the correct format: Aarhus_Airport | cityServed| "Aarhus, Denmark")
Triples2text	Picture yourself as an expert in scrutinizing a text, effectively extracting all entities and their relationships and then constructing text based on the given triples. Once I supply you with triples in the (subject|relationship|object) format, your duty is to reexamine these triples and create text that imparts their semantic interpretation.	Triples	Sentence
Reflection	Picture yourself as being highly skilled in text dissection, with the ability to efficiently identify all entities and their ties. When provided a text along with triples in the (subject|relationship|object) format, you are to check these triples in light of the text and correct any inaccuracies.	Sentence Triples	Corrected triples

Table 3.2: The Overall Data Augmentation Tricks

anism. However, it is difficult to conduct further research due to the close-source nature of models. Then, open-source decoder-only LLMs like Alpaca and Vicuna are released, which are fine-tuned based on LLaMA [27] and achieve competitive performance with ChatGPT and GPT-4.

ChatGPT-3.5 and GPT-4 Human-like conversations are the main purpose of ChatGPT, an advanced LLM created by OpenAI. To improve ChatGPT's alignment with human tastes and values, it uses RLHF [41] during the fine-tuning process. GPT-4, an advanced big language model created by OpenAI, is expanding on the achievements of its forerunners, such as GPT-3 and ChatGPT.

Vicuna-13B [42],Wizard [43], Orca [44], **LLaMA** [27], LlongOrca [45], **SOLAR** 10.7B [46]**Mixtral** Mixtral[3], Mistral mode[4], **Platypus** Platypus-30B [47] is the open-source model we choose from HuggingFace.

3.3.3 Prompt Engineering & Data Preparation

Training Dataset Name	**Used Data Format(s)**	**Size**
WebNLG (original)	*Text2triples*	N
WebNLG-combined	*Text2triples* + *Explanations* + *Triples2text*	3*N
WebNLG-combined-with-reflections	*Text2triples* + *Explanations* + *Triples2text* + *Reflection*	4*N
WebNLG-reflections-updated-instructions	*Text2triples* + *Explanations* + *Triples2text* + *Reflection* + new_instructions	4*N

Table 3.3: Variants of WebNLG Training Data

Prompt engineering is an in-context method for learning language models. In a nutshell, a prompt is a sequence of natural language inputs for a model, consisting of an instruction, context, and input text. The instruction guides the model to perform a specific task, while the context provides additional information; the input text is the text to be processed by the model. An example of the triple extraction prompt is shown in Figure 3.1.

In this work, we used different prompt formats for various models, ensuring that both

[3]https://mistral.ai/news/mixtral-of-experts/
[4]https://huggingface.co/ignos/Mistral-T5-7B-v1

fine-tuning and inference employed the same prompt format. The three types of prompts are detailed in Table 3.1. The components {**instruction**}, {**input**}, and {**output**} are replaced with information/data specific to the proposed Data Formats, Table 3.2.

The experiments have been conducted with the training datasets built with different versions of Data Formats. Such an approach allowed us to increase the size of training datasets by 3- and 4-fold. The process of building different datasets is illustrated at the top of Figure 3.2. Examples of data formats are included in Table 3.2. Each format has its style of the *instruction*, *input*, as well as *output*. The tasks associated with each Data Format differed from explaining the extraction process via reconstructing a sentence from triples to evaluating triples. The data formats were used to construct various Training Datasets, Table 3.3.

The first Training Dataset is called **WebNLG-combined dataset**. It contains 39,633 entries in three categories/subsets, each of 13,211 entries. The first subset includes *Test2triples*, i.e., sets of sentences together with the triples extracted from them. The second subset is the extension of the first one. We have added *Explanation* of the triple extraction process. The explanations were generated by prompting GPT-3.5 with the input text and the ground truth triples to elucidate the extraction process. The explanations comprise entity identification, property analysis, source derivation, entity relationships, and the resultant triples. The third is *Triple2text* subset. It sets the ground truth triples as the model input and the original text as the target output. The aim is to enhance reasoning capabilities and improve triple generation performance.

The second generated Training Data is named **WebNLG-combined-with-reflections** with 52,844 entries. We have extended the WebNLG-combined dataset with so-called *Reflection* data. These data were generated by a *Vicuna* model previously trained for the triple extraction task using *Test2triples* and *Explanation*. The model was fed with the text and triples generated from it, and the task was either amending the triples or confirming their correctness. The anticipated output was either a confirmation that a given input triple was

accurate or its correct version.

For both datasets mentioned above, the **instruction** was randomly selected from the previously generated set of twenty distinct instructions. These instructions were a mix of human-authored instructions and variations generated by GPT-4 to enhance diversity. All instructions underwent thorough evaluation before they were used.

The **WebNLG-reflections-updated-instructions** dataset was the WebNLG-combined-with-reflections dataset when a new set of instructions was used. This time, there are eleven instructions: ten newly constructed and one from the original set. Again, this new set of instructions is a mixture of human-written and rephrased by GPT-4.

3.3.4 Overall Experiments Setup

The workflow of experimental steps and some details about the components forming different Training Datasets are shown in Figure 3.2. Once the datasets were prepared, the models have been tuned and benchmarked using the testing dataset. The final step was an evaluation of the results (for details, see next section).

To prepare models for the process of triple extraction, we utilized HuggingFace libraries to perform supervised finetuning utilizing Parameter-Efficient Finetuning (PEFT) [48] and Low-Rank Adaptation (LoRA) [49] on the WebNLG dataset. We used two prewritten trainers, *finetune script* from *alpaca-Lora* and *autotrain-advanced* from HuggingFace. The *finetune script* was slightly modified to change evaluation steps and to ensure the graphics processing unit (GPU) cache was cleared after all evaluations and checkpoints were saved. All models were trained using two Nvidia 3090 24GB GPUs and a cutoff length 1024, with varying configurations of packages and datasets based on the trainer used.

For the *finetune script*, we set an approximately 85:15 split between training and validation data. The validation set size is 6,000 for *WebNLG-combined* and 8,000 for *WebNLG-combined-with-reflections*.

For *autotrain-advanced*, Supervised Fine-tuning (SFT) Trainer is used from the Trans-

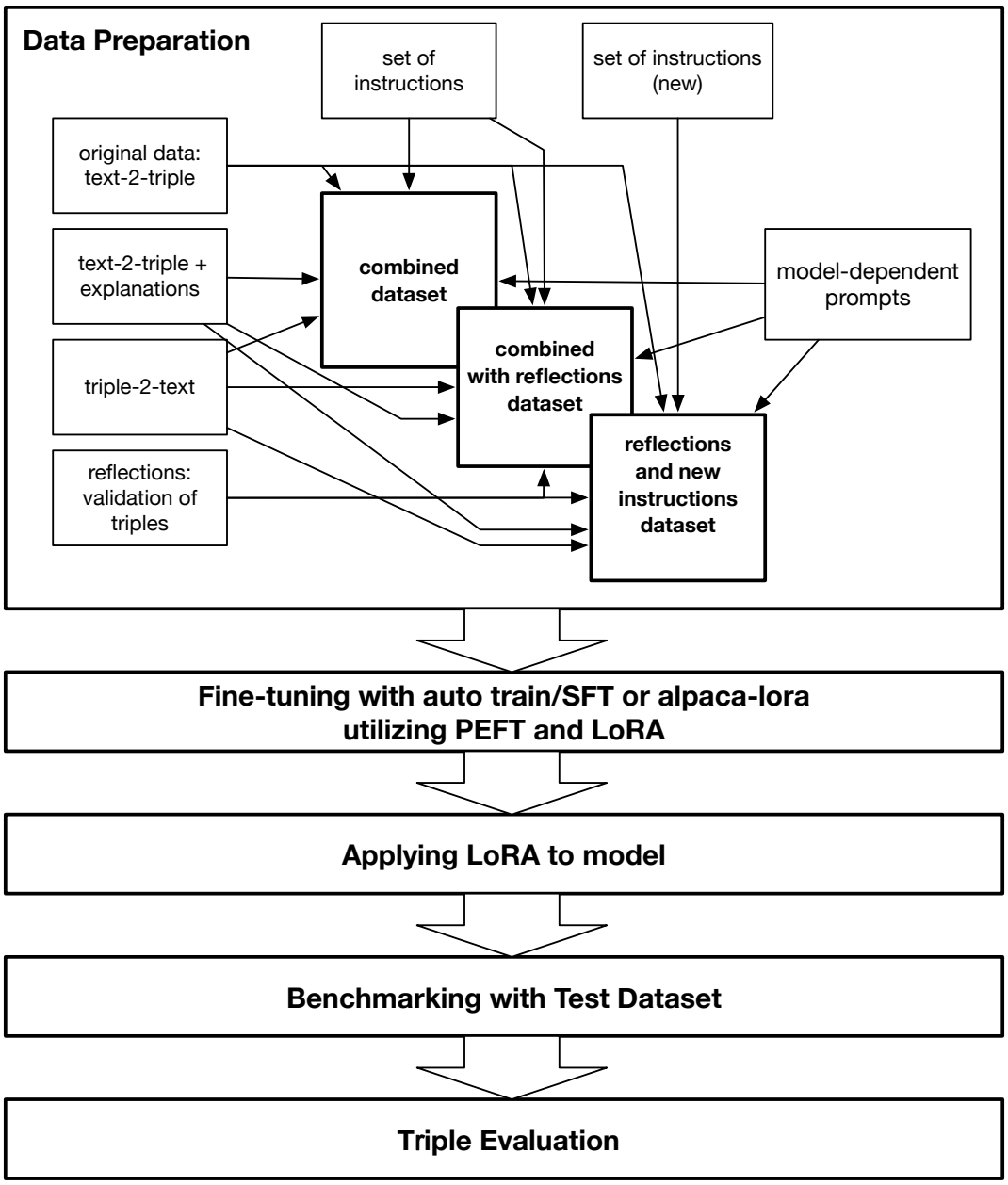


Figure 3.2: Experimental Workflow

former Reinforcement Learning (TRL) package that is included as an option for training in autotrain-advanced [50]. The *WebNLG-reflections-updated-instructions* dataset is used. It contained different instructions for each training task, including additional details about formatting triples and better explaining the model's role.

We trained a collection of eleven models chosen based on relative performance on the HuggingFace LLM leaderboard, and compare their performance between each other and GPT-4. After training, the LoRA weights are combined with the base model to obtain our fine-tuned model output. These exported weights are used to run inference on the model.

3.4 Evaluation Procedure and Results

3.4.1 Evaluation Procedure

	Type			Partial			Exact			Strict		
Model	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
GPT-4 50 samples	0.706	0.729	0.714	0.684	0.707	0.692	0.651	0.668	0.657	0.640	0.652	0.645
GPT-4-0314	0.693	0.711	0.700	0.668	0.688	0.675	0.634	0.649	0.640	0.626	0.634	0.629
ChatGPT-3.5-2023	0.592	0.610	0.599	0.570	0.588	0.577	0.533	0.548	0.539	0.521	0.532	0.525
GPT-4 Full	0.567	0.624	0.587	0.536	0.580	0.552	0.478	0.506	0.488	0.455	0.482	0.465
Vicuna-7b	0.715	0.729	0.721	0.702	0.714	0.706	0.683	0.693	0.687	0.680	0.689	0.683
WizardLM-7b	0.700	0.715	0.706	0.688	0.701	0.693	0.671	0.682	0.675	0.667	0.677	0.671
Orca-mini-7b	0.683	0.700	0.690	0.670	0.686	0.677	0.652	0.664	0.657	0.647	0.658	0.652
Orca-mini-2-7b	0.711	0.726	0.717	0.698	0.710	0.703	0.681	0.690	0.684	0.677	0.687	0.681
Orca-mini-3-7b	**0.746**	**0.762**	**0.753**	**0.732**	**0.746**	**0.738**	**0.715**	**0.726**	**0.719**	**0.712**	**0.723**	**0.717**
Llama-2-7b	0.705	0.714	0.708	0.689	0.698	0.693	0.669	0.677	0.673	0.666	0.673	0.669
Llama-2-chat-7b	0.685	0.700	0.691	0.670	0.684	0.675	0.650	0.660	0.654	0.645	0.654	0.649
LlongOrca-7b	0.710	0.722	0.715	0.697	0.707	0.701	0.680	0.689	0.684	0.677	0.685	0.680
SOLAR-Instruct-10b	0.729	0.741	0.734	0.716	0.727	0.720	0.699	0.708	0.703	0.697	0.705	0.700
Mistral-t5-7b	0.731	0.746	0.738	0.716	0.729	**0.721**	0.697	0.708	0.702	0.695	0.704	0.698
Mixtral-8x7b	0.730	0.739	0.734	0.716	0.725	0.720	0.699	0.706	0.702	0.696	0.702	0.698
Vicuna-33b	**0.750**	**0.762**	**0.755**	**0.738**	**0.749**	**0.742**	**0.723**	**0.732**	**0.727**	**0.720**	**0.729**	**0.724**
Platypus-30b	**0.747**	**0.762**	**0.753**	**0.732**	**0.746**	**0.738**	**0.715**	**0.726**	**0.720**	**0.713**	**0.724**	**0.718**

Table 3.4: WebNLG-reflections-updated-instructions Performance Results

The evaluation framework comprises two phases: Inference, generating the model's

output on the test set, and evaluation, comparing this output against ground truth triples. All models were benchmarked with a maximum token limit of 1,024, and the output was generated without streaming. For evaluation, the numerical results such as precision, recall, and F1, and saved as the output file. The test set includes the same instructions in our training data and includes 2,155 instances of directly extracting triples from text.

The scores are calculated using the evaluate package [51]. It calculates metrics based on four different criteria. First is *type evaluation (TE)* where only the tags must match to be considered correct. These tags are SUB, PRED, and OBJ for the subject, predicate, and object. *Partial evaluation (PE)* requires the triples to match partially or completely, irrespective of tag, to be considered partially or completely correct. *Exact evaluation (EE)* requires the triples to match exactly, irrespective of tag, to be considered correct. *Strict evaluation (SE)* requires both the triples and tag to match to be considered correct. Each evaluation type assigns a label of correct (COR), incorrect (InCOR), missed (MIS), or spurious (SPU), based on the triples and tags. Partial (PAR) is assigned only for the partial evaluation type. MIS and SPU are across all evaluation types, with MIS being assigned for each part of a reference triple when there is no matching candidate, and SPU assigned for each part of a candidate triple when there is no matching reference. The following formulas are calculations of precision (P), recall (R), and F1. The type and partial scores are calculated with the "partial" formulas and exact and strict scores are calculated with the "exact" formulas:

$$\begin{aligned} Possible &= COR + InCOR + PAR + MIS = TP + FN \\ Actual &= COR + InCOR + PAR + SPU = TP + FP \end{aligned} \tag{3.1}$$

$$\begin{aligned} P_{TE|PE} &= \frac{COR + 0.5 * PAR}{Actual} \\ R_{TE|PE} &= \frac{COR + 0.5 * PAR}{Possible} \end{aligned} \tag{3.2}$$

$$\begin{aligned} P_{EE|SE} &= \frac{COR}{Actual} = \frac{COR}{COR + InCOR + SPU} \\ R_{EE|SE} &= \frac{COR}{Possible} = \frac{COR}{COR + InCOR + MIS} \end{aligned} \tag{3.3}$$

3.4.2 Results

WebNLG Dataset. The obtained results for the fine-tuned models are included in Table 3.4. It can be observed that small 7b models *Orca-mini3-7b* and *Mistral-t5-7b* have the best performances even when compared with GPT-4. The *Orca-mini3-7b* model achieved the highest F1 scores for all evaluations, outperforming all 7b models in our comparative analysis.

Small variations have been observed between training methods and datasets. In general, models show slight improvement from WebNLG-combined to WebNLG-combined-with-reflections and then to WebNLG-reflections-updated-instructions. Additionally, modifying the instructions shows a decrease in training loss. GPT models had a bigger drop in performance going to the exact and strict metrics compared to our models, which resulted in our models performing relatively better on the exact and strict metrics.

Ablation Study. We performed ablation studies to evaluate the impact of different data augmentation strategies on the performance of these models. Figure 3.3 shows the effects of various data augmentation techniques on the models' performance. We show the performance results obtained for two models – *Orca* and *Vicuna* – and four Training Datasets: the original WebNLG dataset, the WebNLG-combined dataset, the WebNLG-combined-with-reflections dataset, and the WebNLG-reflections-updated-instructions dataset, Table 3.3. We report the precision, recall, and F1 values for the most demanding task of generating triples identical to those provided as the target. It is easily seen that the results obtained for the last Training Dataset are the best.

Other Datasets. Two models *Orca-mini-3* and *Llama-2-13b* have been finetuned on different datasets, Table 3.5. The best scores have been obtained for the **KELM** dataset. The *Llama-2-13b* finetuned on another dataset **DocRED** performed very poorly and was completely unable to learn proper formatting of triples.

The main issue with inference on other data is related to the type of triple properties and how many triples are extracted from a single sentence. For example, the analysis of the

DocRED dataset revealed that it is focused mainly on such relations as *country* and *location* while ignoring any other relations. In DocRed, a few triples are extracted from paragraph sentences. There is much looping in the models' output; models do not efficiently learn triple formats. Some outputs were of the form (subject | predicate | object). Further, there are only about 3,000 entries in annotated training data. For yet another dataset – **FewRel** – the issue seems to be related to the model not knowing when to generate triples following the DBpedia and when using Wikidata formats.

3.5 Discussion and Limitations

The obtained results and their analysis have led to a few observations that confirmed known facts about tuning large models and allowed to draw some new ones. We can categorize them into three parts: data size, model selection, and interaction with a model (prompt and data preparation).

Size and Quality of Datasets. It is a well-known fact that larger datasets lead to better results. Such an obvious statement is also true for the triple extraction process. It is seen in Table 3.5. The results obtained for KELM data – 400,000 samples in the training set – confirm that. The model was tuned with a simple prompt containing text-2-triple and instructions. Comparing that with our primary focused data, WebNLG, which includes only 13,211 training datasets, shows a significant advantage of large datasets.

Once we collected results for the other two datasets – DocRED and FewRel - we investigated the content of the training datasets. It has become apparent that the reference triples that were supposed to be constructed from sentences were of poor quality: limited to a few relations, incoherent structure, a limited number of triples (quite often just one) form small paragraphs.

Model Selection/Multilingual Triples. In our experiments, one of the datasets – SKE – is a set of Chinese sentences and extracted from them triples. The difference in results obtained from *orca-mini-3-7b* and *llama-2-13b* is very large. A quick investigation revealed

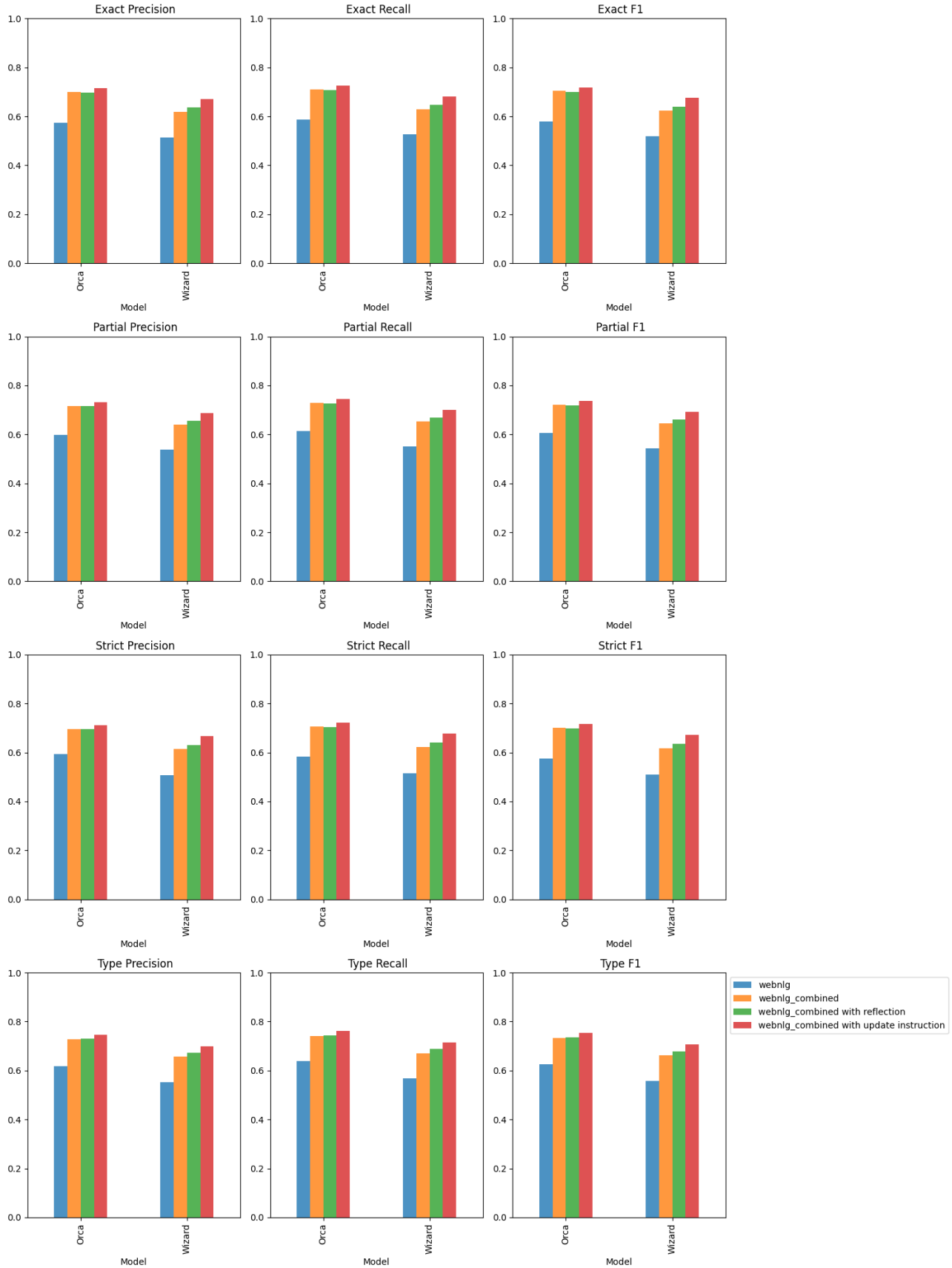


Figure 3.3: Results of Ablation Studies on Two Modes: Orca and Wizard

		Type			Partial			Exact			Strict		
Data	Model	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
SKE	orca-mini-3	0.828	0.828	0.828	0.829	0.829	0.829	0.829	0.829	0.829	0.827	0.827	0.827
	llama-2-13b	0.129	0.127	0.127	0.130	0.128	0.129	0.127	0.127	0.127	0.124	0.124	0.124
DocRED	orca-mini-3	0.057	0.054	0.052	0.050	0.050	0.048	0.031	0.031	0.030	0.024	0.025	0.024
	llama-2-13b	0.096	0.037	0.051	0.051	0.022	0.028	0.002	0.002	0.002	0.002	0.002	0.002
FewRel	orca-mini-3	0.314	0.402	0.342	0.354	0.425	0.376	0.312	0.362	0.327	0.240	0.286	0.254
	llama-2-13b	0.304	0.378	0.325	0.344	0.405	0.361	0.297	0.340	0.310	0.224	0.263	0.236
KELM	orca-mini-3-7b	0.867	0.899	0.879	0.848	0.873	0.857	0.823	0.841	0.830	0.820	0.837	0.826
	Llama-2-13b	0.861	0.865	0.852	0.825	0.836	0.825	0.779	0.796	0.785	0.769	0.786	0.776
raw_Webnlg	orca-mini-3-7b	0.618	0.638	0.626	0.598	0.615	0.605	0.574	0.588	0.579	0.593	0.583	0.575
	Llama-2-13b	0.62	0.637	0.626	0.602	0.618	0.608	0.581	0.593	0.586	0.577	0.588	0.581

Table 3.5: Performance on Other Datasets

that the dataset used to train the *orca-mini-3-7* model contained a large amount of Chinese text. Again, it confirms a commonsense fact that if a language model is not exposed to a text in a given language, its performance, related to this language, is not satisfactory.

Prompt and Data Preparation. The most interesting and important observation coming from our experiments is a high significance of the creative approach to constructing prompts and 'augmentation' of the training datasets.

As indicated earlier, the task of extracting triples from WebNLG data involves the usage of DBpedia vocabulary. In particular, properties/relations of the extracted triples have/should be in the DBpedia format. The WebNLG dataset has been analyzed to ensure the training data is of high quality. DBpedia ontology has been used to determine if the triples/relations were consistent.

The consistent structure of triples is essential so the model can effectively learn how to form triples properly. Exposure to different properties is also of high importance. The properties seen in the training and testing sets overlapped, with thirty-six properties unique to the test set. All properties were checked to ensure they were present in DBpedia.

A small amount of training data, just 13,211, has forced us to generate larger datasets from the original set via setting different tasks related to processing and extraction of triples. Section 3.3.3 details how various versions of Training Datasets were created. We

enhanced the data with explanations of triple generation processes generated by GPT-3.5 and previously tuned *Vacuna* model, generation of sentences based on sets of triples, and simple evaluation of extracted triples. These activities have improved our best model's performance, i.e., *orca-mini-3-7b*.

Limitations There are some limitations of fine-tuned models. They hallucinated on occasion, especially when they generated responses for more well-known topics, such as when we asked them to generate a response to the Jeff Bezos Wikipedia article. The models frequently hallucinated the birthplace of Bezos, providing false information about the location. Also, models had looping issues, where they would continually generate output until they reached the token limit.

3.6 Conclusions

The paper aims to investigate different scenarios of a triple extraction task. Various models and a few datasets have been used in the experiments. A prime contribution is the development of a procedure/methodology for augmenting the original dataset. The additions included several tasks indirectly related to the triple extraction process: explaining the extraction steps, reconstructing sentences from triples, and determining the correctness of extracted triples. It resulted in enlarged training datasets (3- or 4-fold). As an outcome, the performance of 7b tuned models is comparable to or even better than that of well-known models from the GPT family.

The applied procedures concentrated on generating triples containing elements compatible with a specific vocabulary, in our case, DBpedia. While our models suffer from occasional looping and hallucinations, they effectively extract triples following DBpedia ontology from sentences. The results demonstrate that achieving and exceeding GPT performance with fine-tuned models is possible without large datasets.

Chapter 4

Hierarchical Topic Modelling for Knowledge Graphs

In this chapter, we present a hierarchical topic modeling approach designed for knowledge graphs. This method is inspired by probabilistic topic modeling techniques, particularly Latent Dirichlet Allocation (LDA) and its hierarchical extension (hLDA). Our model aims to uncover latent structures within knowledge graphs by organizing entities and predicates into a tree of abstract topics.

The primary components of our approach are:

- Data Preprocessing: Knowledge graphs are collections of triples (subject, predicate, object). Our model treats predicates and <predicates, objects> as tags that describe subjects, similar to how words describe documents in topic models.

- Generative Model: The model generates a hierarchy of topics (nodes in a tree) where each node represents a distribution over tags and predicates. Subjects sample paths through this tree, providing a hierarchical clustering of subjects and a hierarchical organization of topics.

- Inference: We utilize a non-parametric prior (nested Chinese Restaurant Process) for tree generation, allowing the model to determine the tree structure based on data without requiring predefined parameters. Gibbs sampling, leveraging Multinomial-

Dirichlet conjugacy, is used for efficient posterior inference, making the model scalable to large datasets.

- Evaluation: We evaluate our model on three common datasets (FB15k-237, YAGO3-10, and DBpedia), and compare it against existing hierarchical clustering techniques. Our results show that our model can induce coherent topic hierarchies and perform well in clustering tasks.

Key Findings:

1. Our hierarchical topic model effectively organizes knowledge graph elements into meaningful clusters without requiring prior assumptions about the tree structure.
2. Quantitative evaluations on benchmark datasets demonstrate the model's competitive performance in clustering tasks.
3. Qualitative assessments highlight the coherence and interpretability of the induced topic hierarchies.

Limitions:

1. The subject is delineated by a singular path within the acquired tree, which inadequately represents subjects encompassing a diverse range of topics.
2. The breadth of the tree is attributed to the lack of control over redundant topics, leading to the inclusion of overlapping subjects such as artist, writer, and artist and writer.
3. The computational inference process consumes a significant amount of CPU resources and operates at a notably slow pace.

This work was published in the European Semantic Web Conference (ESWC) 2022, 29 May - 2 June; Hersonissos, Greece.
The primary author, Yujia Zhang, designed and conducted the experiments under the supervision of Prof. Marek Reformat.

Hierarchical Topic Modelling for Knowledge Graphs

Abstract: Recent years have demonstrated the rise of knowledge graphs as a powerful medium for storing data, showing their utility in academia and industry alike. This in turn has motivated substantial effort into modelling knowledge graphs in ways that reveal latent structures contained within them. In this paper, we propose a non-parametric hierarchical generative model for knowledge graphs that draws inspiration from probabilistic methods used in topic modelling. Our model discovers the latent probability distributions of a knowledge graph and organizes its elements in a tree of abstract topics. In doing so, it provides a hierarchical clustering of knowledge graph subjects as well as membership distributions of predicates and entities to topics. The main draw of such an approach is that it does not require any a priori assumptions about the structure of the tree other than its depth. In addition to presenting the generative model, we introduce an efficient Gibbs sampling scheme which leverages the Multinomial-Dirichlet conjugacy to integrate out latent variables, making the posterior inference process adaptable to large datasets. We quantitatively evaluate our model on three common datasets and show that it is comparable to existing hierarchical clustering techniques. Furthermore, we present a qualitative assessment of the induced hierarchy and topics.

4.1 Introduction

Knowledge bases have received considerable research attention in recent years, demonstrating their utility in areas ranging from question answering [52, 53] to knowledge generation [54, 55, 56] to recommender systems [57]. These knowledge bases are underpinned by graph structures called knowledge graphs which describe facts as a collection of triples that relate two entities via a predicate. Advances in artificial intelligence have spurred on the need to find representations of knowledge graphs which can be easily and accurately reasoned with by machines. One aspect of this is the increased research attention devoted

to generative models for knowledge graphs which learn the latent probability distributions of a graph. These models work by decomposing the knowledge graph to a set of probability distributions that, when sampled together, generate its relations. The learning process, therefore, amounts to inferring the posterior distribution conditioned on the data.

Probabilistic topic models are types of generative models that have received considerable attention in the field of natural language processing. The aim of these models is to build abstract word topics from a corpus of documents and their words. In this sense, topics may be viewed as clusters of words. Most topic models operate under the intuition that words which co-occur in the same documents are likely to have similar semantics and therefore belong to the same topics. Hierarchical topic models extend this principle and organize the induced topics into a topic hierarchy whereby each ancestor topic represents a conceptually coarser version of its descendant topics.

In this paper, we present a model for generating a topic hierarchy from knowledge graphs which extends on existing topic models. In our model, topics are collections of entities and predicates, and are organized hierarchically in the form of a rooted tree. In generating these topics, our model also implicitly hierarchically clusters subjects by sampling a corresponding tree path. Furthermore, we employ a non-parametric prior over the tree, allowing our model to be free of any a priori assumptions about its structure other than its depth. We present an efficient Gibbs sampling scheme for posterior inference of our model. The approach leverages the Multinomial-Dirichlet conjugacy to integrate out parameters for faster inference. Our evaluation demonstrates our model's ability to induce a coherent topic hierarchy as well as hierarchical subject clustering.

4.2 Proposed Model

In this section, we describe our model by positioning it in the context of existing probabilistic topic models from which it draws inspiration. Specifically, we first introduce readers to Latent Dirichlet Allocation (LDA) [30] and Hierarchical Latent Dirichlet Allocation

(hLDA) [58] before formalizing our model.

4.2.1 Problem Formulation

We define a knowledge graph as a collection of triples, $\mathcal{K}$, such that each triple relates a subject entity, s, to an object entity, o, via a predicate, p. Formally, $\mathcal{K} = \{\langle s, p, o\rangle \in \mathbf{S} \times \mathbf{P} \times \mathbf{O}\}$ where $\langle s, p, o\rangle$ is a triple, and $\mathbf{S}$, $\mathbf{P}$, and $\mathbf{O}$ are the sets of subjects, predicates, and objects in $\mathcal{K}$, respectively. We note that knowledge graphs are rarely bipartite in terms of $\mathbf{S}$ and $\mathbf{O}$. In other words, entities can take on the role of both subjects and objects in $\mathcal{K}$, thus $\mathbf{S} \cap \mathbf{O} \neq \emptyset$. Our goal is to find a representation of the knowledge graph in which entities and predicates are hierarchically organized such that entities representing coarse concepts subsume their fine grained counterparts. For instance, the concept Person is a coarser concept than Artist since it encompasses all persons, including artists and non-artists. A natural representation of this paradigm is a directed tree wherein coarse concepts occupy nodes closer to the root node. Nodes are then collections of entities and predicates which share similar semantics. Paths in the tree capture the progressive granularization of a concept.

4.2.2 Probabilistic Topic Models

Given a collection of documents and their words, $\mathbf{D}$, topic models generate abstract topics on the intuition that words belonging to the same topic are likely to occur in the same documents. Latent Dirichlet Allocation (LDA) [30] is a canonical example of the topic models used today. In this approach, each document, $d_i \in \mathbf{D}$, is a mixture of topics and each topic is a distribution of words. To generate a document, the number of document words, W_i, the document's topic mixture, θ_i, and each topic's word distributions, β_k, are sampled. For each document word, $w_{i,j}$, first a topic indicator $z_{i,j}$ is sampled according to θ_i then the word is generated from z_j's word distribution, β_{z_j}. This generative procedure is formally defined as follows:

- for each document; $d_i \in \mathbf{D}$
  - $W_i \sim$ Poisson(ξ)
  - $\theta_i \sim$ Dirichlet(α)
- for each topic; $k \in 1, 2, ..., K$
  - $\beta_k \sim$ Dirichlet(η)
- for each document; $d_i \in \mathbf{D}$
  - for each word in document; $w_{i,j} \in d_i$
    - $z_{i,j} \sim$ Multinomial(θ_i)
    - $w_{i,j} \sim$ Multinomial$(\beta_{z_{i,j}})$

Learning the distributions which generate the documents amounts to inferring the posterior distribution. Although this problem is intractable for exact inference, it can be approximated with algorithms such as Variational Bayes [30] or Collapsed Gibbs Sampling [59]. We refer readers to the original papers for the full inference procedure.

LDA has been extended to generate a hierarchy of topics in Hierarchical Latent Dirichlet Allocation (hLDA) [58]. The foundation of hLDA is the nested Chinese restaurant process (nCRP) which is an extension of the Chinese restaurant process (CRP) [60]. The CRP is a recursively defined stochastic process which gets its name from the analogy of seating patrons at a Chinese restaurant. In this restaurant, there are an infinite number of tables and each table can seat an infinite number of guests. When a guest enters, the probability of him being seated at a table is proportional to the number of patrons already seated at the table. Formally, when seating guest g_i at a restaurant that has M non-empty tables, the probability of seating the guest at table m is:

$$\mathbb{P}(g_i = m | g_{i-1}, ..., g_1) = \begin{cases} \frac{|n_m^i|}{i - 1 + \gamma} & m \leq M \\ \frac{\gamma}{i - 1 + \gamma} & m = M + 1 \\ 0 & M + 1 < m \end{cases} \tag{4.1}$$

where $|n_m^i|$ is the number of patrons sitting at table m when guest g_i arrives and γ is a hyperparameter which controls the probability that an incoming guest will be seated at an empty table.

The nCRP is used in hLDA as an infinitely deep and infinitely branching prior over a tree structure. In this process, a tree is generated by sampling a path, c_i, at each level in the tree via the CRP. Each node in a tree, $n_k \in \mathbf{N}$, has its own CRP and being seated at a table is analogous to taking a specific branch in the path down the tree. As before, the probability of taking a path is proportional to the amount of times the path has been taken before. When arriving at a node n_k with children $\mathbf{M}_k$ on the $(l-1)^{\text{th}}$ level in the tree, the probability of selecting an existing branch, $c_i[l] \in \mathbf{M}_l$ or creating a new branch, $c_i[l] = M_k^*$, is:

$$\mathbb{P}(c_i[l] = m | c_{i-1:\,1}, c_i[l-1:\,1]) = \begin{cases} \dfrac{|n_m^i|}{|n_k^i| + \gamma} & m \in \mathbf{M}_k \\ \dfrac{\gamma}{|n_k^i| + \gamma} & m = M_k^* \end{cases} \tag{4.2}$$

where $\mathbf{c}_i[l]$ is the node on the path of d_i at level l, $M_k^* = \min(\mathbb{Z}^+ \setminus \mathbf{M}_k)$ is the smallest positive integer not in $\mathbf{M}_k$, and $|n_k^i|$ is the number of entities that have gone through node n_k when entity i arrived, $|n_k^i| = |\{j \in \mathbb{Z}^+ \colon j < i \wedge \mathbf{c}_j[l] = n_k\}|$.

Putting everything together, hLDA uses the nCRP to generate a tree of topics. The tree is bounded to a maximum depth of L and each node in the tree is associated with a topic β_k. Each document d_i samples a path through L nodes in the tree, c_i, and a topic distribution over levels in the tree analogous to the topic mixture in LDA, θ_i. For each word $w_{i,j}$ in d_i, a topic $z_{i,j}$ is sampled from θ_i and a word is generated from that topic. The generative process is summarized as follows:

- for each node in the tree; $n_k \in \mathbf{N}$
  - $\beta_k \sim \text{Dirichlet}(\eta)$
- for each document; $d_i \in \mathbf{D}$
  - $c_i \sim \text{nCRP}(\gamma)$

- $\theta_i \sim \text{GEM}(\rho, \pi)$
- for each word in document; $w_{i,j} \in d_i$
  - $z_{i,j} \sim \text{Multinomial}(\theta_i)$
  - $w_{i,j} \sim \text{Multinomial}(\beta_{c_i[z_{i,j}]})$

where $\text{GEM}(\rho, \pi)$ stands for the stick-breaking process [61] and functions as the prior for topic levels. As with LDA, we refer the readers to the original papers for model inference.

4.2.3 Model Description

We present our model as an extension of hLDA which has been adapted to knowledge graphs. As such, we adopt the previously introduced concepts and notation, and focus on highlighting the differences.

The first difference is the departure from the domain of documents and words to that of subjects, predicates, and objects. We can think of a predicate-object pair as a *tag* which describes a subject in a way that is analogous to how a word describes a document. In this view, a tag, t, is defined as $\langle p, o \rangle$ and belongs to a subject such that $t_{i,j} \in \mathbf{T}_i$ denotes that tag $t_{i,j}$ belongs to subject s_i. This formulation is leveraged in our model by assigning a tag topic distribution, β^t, for each node in the tree. Furthermore, to capture the distributions of predicates in each cluster, we mix in a predicate specific topic, β^p. Predicates share their level indicators, $z_{i,j}$, with their corresponding tags. As such, the number of predicates belonging to a subject has to equal its tag count. We define the multiset of predicates which belong to subject s_i as $p_{i,j} \in \mathbf{P}_i$ such that $|\mathbf{P}_i| = |\mathbf{T}_i|$. Thus, each node is a collection of two topics whose elements span the domain of $\mathbf{T} \cup \mathbf{P}$.

Each subject s_i samples a path, c_i, through the tree using the nCRP as well as a level distribution, θ_i. A further departure from the original hLDA model is the replacement of the stick-breaking process as the prior of the level distribution with the Dirichlet distribution. This formulation is a return to the prior used in LDA and was chosen for two reasons. The first is that the Dirichlet distribution introduces only one hyperparameter in contrast

to the stick-breaking process' two. This makes our model easier to apply a priori since hyperparameter sensitivity and selection present challenges in non-parametric models. The second is that the inference scheme is simpler when using the Dirichlet prior. Finally, the theoretical benefits of the stick-breaking prior are not justified in a practical context since the infinite distribution would get bounded in our model by the tree depth, L.

As mentioned previously, level indicators, $z_{i,j}$, are shared among corresponding predicates and tags. Thus, we sample one level indicator for each tag analogously to hLDA. This indicator is used in conjunction with the subject path to determine the node whose topics will be sampled from. Unlike hLDA which only samples words, our model samples predicates and tags from the selected node's predicate and tag topic distributions, $\beta^p[c_i[z_{i,j}]]$ and $\beta^t[c_i[z_{i,j}]]$, respectively. We use the notation $\beta^p[c_i[z_{i,j}]]$ and $\beta^t[c_i[z_{i,j}]]$ to denote the predicate and tag topic distributions of the node at level $z_{i,j}$ on path c_i. The generative process is defined as follows:

- for each node in the tree; $n_k \in \mathbf{N}$
  - $\beta^p \sim$ Dirichlet(η_p)
  - $\beta^t \sim$ Dirichlet(η_t)
- for each subject; $s_i \in \mathbf{S}$
  - $c_i \sim$ nCRP(γ)
  - $\theta_i \sim$ Dirichlet(α)
  - for each tag in subject; $t_{i,j} \in \mathbf{T}_i$
    - $z_{i,j} \sim$ Multinomial(θ_i)
  - for each predicate in subject; $p_{i,j} \in \mathbf{P}_i$
    - $p_{i,j} \sim$ Multinomial$(\beta^p[c_i[z_{i,j}]])$
  - for each tag in subject; $t_{i,j} \in \mathbf{T}_i$

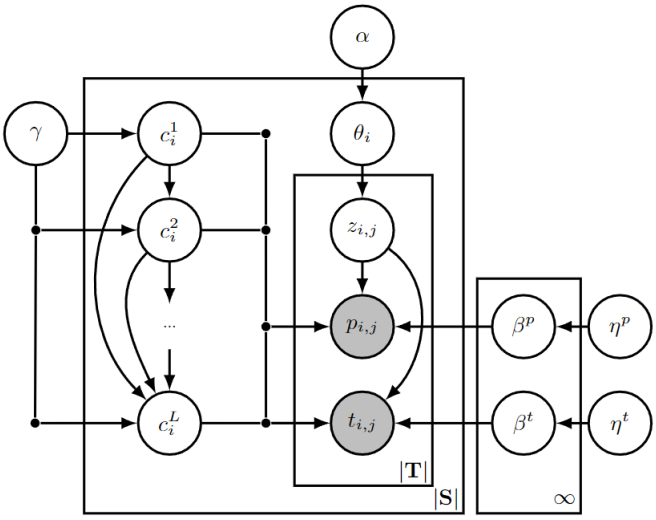


Figure 4.1: Plate diagram for our model.

* $t_{i,j} \sim \text{Multinomial}(\beta^t[c_i[z_{i,j}]])$

η_p and η_t are hyperparameters of our model which control the sparsity of the topics such that lower η values result in sparser topics which are more dissimilar from one another. Furthermore, the ratio between η_p and η_t controls the relative importance of predicates to tags when calculating the likelihood functions. γ is a hyperparameter of the nCRP and controls the probability of creating a new path in the tree such that higher γ values will generate trees with a higher average branching factor. Finally, α is the topic level hyperparameter. We provide a graphical representation of our model using plate notation in Figure 4.1.

4.2.4 Inference

Our model is intractable for exact inference, thus we approximate it using collapsed Gibbs sampling for posterior inference. The goal of the sampling scheme is to generate the subject paths, **c**, and level indicators, **z**, by inferring the latent parameters. For faster mixing, we integrate out the topic distributions, β^p and β^t, as well as the level distributions, θ, by leveraging the Multinomial-Dirichlet conjugacy. This reduces our inference scheme to

simply sampling paths and levels alternately until the parameters of the model are learned, at which point we can collect samples to estimate the true posterior.

Sampling Paths

The posterior distribution of c_i, the path for subject s_i, conditioned on all other variables is:

$$\mathbb{P}(c_i|\mathbf{c}_{-i}, \mathbf{z}_i, \mathbf{P}_i, \mathbf{T}_i, \gamma, \eta_p, \eta_t) \propto \mathbb{P}(c_i|\mathbf{c}_{-i}, \gamma)\mathbb{P}(\mathbf{P}_i|c_i, \mathbf{P}_{-i}, \mathbf{z}_i, \eta_p) \\ \mathbb{P}(\mathbf{T}_i|c_i, \mathbf{T}_{-i}, \mathbf{z}_i, \eta_t) \tag{4.3}$$

where $\mathbf{c}_{-i}$ denotes all paths in the tree excluding the path taken by subject s_i. Likewise, $\mathbf{P}_{-i}$ and $\mathbf{T}_{-i}$ denote the predicates and tags on the tree leaving out those belonging to to subject s_i. This expression is merely an application of Bayes' theorem which states the posterior is proportional to the likelihood times the prior. The first term, $\mathbb{P}(c_i|\mathbf{c}_{-i}, \gamma)$, is the nCRP prior and is calculated as outlined earlier in the paper. The second term, $\mathbb{P}(\mathbf{P}_i|c_i, \mathbf{P}_{-i}, \mathbf{z}_i, \eta_p)$, is the predicate likelihood given the choice of paths. In other words, it is the probability of observing the predicate data if subject s_i were to take path c_i. The calculation of this term is defined as follows:

$$\mathbb{P}(\mathbf{P}_i|c_i,\mathbf{P}_{-i}, \mathbf{z}_i, \eta_p) = \\ \prod_{l=1}^{L} \frac{\Gamma\Big(\sum_{p_{i,j}\in\mathbf{P}_{-i}} \#[\mathbf{z}_{-i} = l, \mathbf{c}_{-i,l} = c_{i,l}, \mathbf{P}_{-i} = p_{i,j}] + \eta_p|\mathbf{P}|\Big)}{\sum_{p_{i,j}\in\mathbf{P}_{-i}} \Gamma\Big(\#[\mathbf{z}_{-i} = l, \mathbf{c}_{-i,l} = c_{i,l}, \mathbf{P}_{-i} = p_{i,j}] + \eta_p\Big)} \\ \prod_{l=1}^{L} \frac{\prod_{p_{i,j}\in\mathbf{P}_i} \Gamma\Big(\#[\mathbf{z}_i = l, \mathbf{c}_{i,l} = c_{i,l}, \mathbf{P}_i = p_{i,j}] + \eta_p\Big)}{\Gamma\Big(\prod_{p_{i,j}\in\mathbf{P}_i} \#[\mathbf{z}_i = l, \mathbf{c}_{i,l} = c_{i,l}, \mathbf{P}_i = p_{i,j}] + \eta_p|\mathbf{P}|\Big)} \tag{4.4}$$

where $\Gamma(.)$ is the gamma function and $\#[.]$ indicates the number of elements that satisfy the given conditions. Finally, the third term, $\mathbb{P}(\mathbf{T}_i|c_i, \mathbf{T}_{-i}, \mathbf{z}_i, \eta_t)$, is the tag likelihood given

the choice of paths and is calculated analogously to the predicate likelihood:

$$\mathbb{P}(\mathbf{T}_i|c_i,\mathbf{T}_{-i},\mathbf{z}_i,\eta_t) = \prod_{l=1}^{L} \frac{\Gamma\Big(\sum_{t_{i,j}\in\mathbf{T}_{-i}} \#[\mathbf{z}_{-i}=l,\mathbf{c}_{-i,l}=c_{i,l},\mathbf{T}_{-i}=t_{i,j}]+\eta_t|\mathbf{T}|\Big)}{\prod_{t_{i,j}\in\mathbf{T}_{-i}}\Gamma\Big(\#[\mathbf{z}_{-i}=l,\mathbf{c}_{-i,l}=c_{i,l},\mathbf{T}_{-i}=t_{i,j}]+\eta_t\Big)} \prod_{l=1}^{L} \frac{\prod_{t_{i,j}\in\mathbf{T}_i}\Gamma\Big(\#[\mathbf{z}_i=l,\mathbf{c}_{i,l}=c_{i,l},\mathbf{T}_i=t_{i,j}]+\eta_t\Big)}{\Gamma\Big(\sum_{t_{i,j}\in\mathbf{T}_i}\#[\mathbf{z}_i=l,\mathbf{c}_{i,l}=c_{i,l},\mathbf{T}_i=t_{i,j}]+\eta_t|\mathbf{T}|\Big)} \tag{4.5}$$

The time complexity of sampling a single path, c_i, is $\mathcal{O}(|\mathbf{N}|(|\mathbf{S}|+|\mathbf{T}|))$, thus sampling all the paths in one iteration of the Gibbs sampler is $\mathcal{O}(|\mathbf{S}||\mathbf{N}|(|\mathbf{S}|+|\mathbf{T}|))$.

Sampling Levels

The posterior distribution of $z_{i,j}$, the level indicator for the j^{th} tag in subject s_i is as follows:

$$\mathbb{P}(z_{i,j}|\mathbf{z}_{i,-j},\mathbf{P}_{i,-j},\mathbf{T}_{i,-j},\mathbf{c},\eta_p,\eta_t,\alpha) \propto \mathbb{P}(z_{i,j}|\mathbf{z}_{i,-j},\alpha)\mathbb{P}(p_{i,j}|\mathbf{P}_{i,-j},\mathbf{c},\mathbf{z}_i,\eta_p) \mathbb{P}(t_{i,j}|\mathbf{T}_{i,-j},\mathbf{c},\mathbf{z}_i,\eta_t) \tag{4.6}$$

where $\mathbf{z}_{i,-j}$ are all the level indicators in subject s_i excluding $z_{i,j}$, the indicator for tag $t_{i,j}$. The prior for level indicators, $\mathbb{P}(z_{i,-j}|\mathbf{z}_{i,-j},\alpha)$, is obtained by integrating out the Multinomial distribution via the Multinomial-Dirichlet conjugacy and calculating the Dirichlet prior as follows:

$$\begin{aligned}\mathbb{P}(z_{i,j}|\mathbf{z}_{i,-j},\alpha) &= \mathbb{E}(z_{i,j}|\mathbf{z}_{i,-j},\alpha)\\ &= \mathbb{E}\Big(\mathbb{E}(z_{i,j}=l)|\theta_1,\theta_2,...,\theta_L,\mathbf{z}_{i,-j},\alpha\Big)\\ &\propto \#[\mathbf{z}_{i,-j}=l]+\alpha\end{aligned} \tag{4.7}$$

The predicate likelihood, $\mathbb{P}(p_{i,j}|\mathbf{P}_{i,-j},c_i,\mathbf{z}_i,\eta_p)$, is calculated by counting the total number of predicates at the node specified by $z_{i,j}$ on path c_i that are the same as $p_{i,j}$:

$$\begin{aligned}\mathbb{P}(p_{i,j}|\mathbf{P}_{i,-j},c_i,\mathbf{z}_i,\eta_p) &= \mathbb{E}(p_{i,j}|\mathbf{z}_i,c_i,\eta_p)\\ &\propto \#[\mathbf{z}_{-(i,j)}=z_{i,j},\mathbf{c}_{z_{i,j}}=c_{i,z_{i,j}},\mathbf{P}_{-(i,j)}=p_{i,j}]+\eta_p\end{aligned} \tag{4.8}$$

The tag likelihood, $\mathbb{P}(t_{i,j}|\mathbf{T}_{i,-j}, \mathbf{c}, \mathbf{z}_i, \eta_t)$, is calculated analogously:

$$\begin{aligned}\mathbb{P}(t_{i,j}|\mathbf{T}_{i,-j}, c_i, \mathbf{z}_i, \eta_t) &= \mathbb{E}(p_{i,j}|\mathbf{z}_i, c_i, \eta_t) \\ &\propto \#[\mathbf{z}_{-(i,j)} = z_{i,j}, \mathbf{c}_{z_{i,j}} = c_{i,z_{i,j}}, \mathbf{T}_{-(i,j)} = t_{i,j}] + \eta_t \end{aligned} \tag{4.9}$$

The time complexity of sampling a single topic, $z_{i,j}$, is $\mathcal{O}(L)$ and meaning that sampling all levels is $\mathcal{O}(|\mathbf{S}||\mathbf{T}||L)$.

Collapsed Gibbs Sampling

As mentioned previously, the collapsed Gibbs sampling process samples paths and levels alternately, as summarized in Algorithm 1 for our model, in Figure 4.1. This approach creates a Markov chain which iteratively approaches its stationary distribution. As such, it is necessary to burn-in a fixed number of samples before samples approximating the posterior distribution may be obtained. Although Gibbs sampling is guaranteed to converge in the infinite case, the speed with which it does so is highly variable and difficult to predict a priori. Monitoring the likelihood of the model is therefore important in determining whether sufficient training has taken place. Furthermore, due to the non-parametric nature of our model, the selection of hyperparameters is critically important. Recall, for instance, that the tree's structure and size changes every time it is sampled. Thus, high γ values may induce trees with branching factors too high to feasibly perform inference on.

Algorithm 1 Gibbs Sampling Procedure

Input: Knowledge graph, $\mathcal{K}$; nCRP hyperparmeter, γ; topic hyperparameters, η^p and η^t; level hyperparameter α; Number of iterations, $iters$
Output: Hierarchical topic model for $\mathcal{K}$ defined by $\mathbf{c}$ and $\mathbf{z}$

```
Obtain S, P, and T from K
for iter = {1, 2, ..., iters} do
    for i ∈ {1, 2, ..., |S|} do
        Sample c_i using Equation 4.3
        for j ∈ {1, 2, ..., |T|} do
            Sample z_{i,j} using Equation 4.6
        end for
    end for
end for
```

4.3 Evaluation

We split the evaluation of our model into two parts: quantitative and qualitative. In our quantitative evaluation, we train our model to obtain a hierarchical clustering of subject entities. This clustering is then evaluated by comparing against ground truth labels and calculating metrics of clustering performance. This gives insight into the quality of induced tree and allocation of subjects to leaf nodes. To assess the quality of the inferred topic clusters, we perform a qualitative evaluation by analyzing the membership distributions of predicates and tags to selected topics. What follows is a summary of our evaluation procedure and discussion of the results. The source code for our model along with the datasets used may be found on GitHub[1].

4.3.1 Datasets

We use three real-world datasets in our evaluation: FB15k-237, YAGO3-10, and DBpedia. The datasets were chosen based on their ubiquity in existing literature and to highlight the scalability of our sampling scheme on large datasets. What follows is a brief description of each dataset.

FB15k-237

The FB15k-237 dataset [62] was constructed from the FB15k dataset [55] by removing redundant and inverse triples. It contains data queried from a version of Freebase that existed around 2013. Specifically, it is comprised of 272115 triples, 14541 entities, and 237 predicates. For our hierarchical clustering analysis, we followed a similar approach to generating a ground truth subset of the data as [63]. Namely, we first mapped entities to the WordNet taxonomy [64] through the $sameAs$ predicate, which relates Freebase entities to YAGO entities. We then extracted triples containing subjects with labels on second level in the taxonomy from the sets provided in Table 4.1. This process yielded a dataset with 5301

[1] https://github.com/yujia0223/hkg

Table 4.1: Summary of Ground Truth Classes used to Derive Clustering Evaluation datasets.

	FB15k-237	**YAGO3-10**	**DBpedia**
Level 1	Person, Organization, Location, Event	Person, Organization, Body of Water	Person, Place
Level 2	Artist, Politician, Scientist, Officeholder, Writer, Musical Organization, Party, Enterprise, Nongovernmental Organization, County, Town, City, Mountain, Movie, Entertainment, Game, Contest	Artist, Politician, Scientist, Officeholder, Writer, Musical Organization, Party, Enterprise, Nongovernmental Organization, Stream, Lake, Ocean, Bay, Sea	Artist, Athlete, PopulatedPlace, NaturalPlace
Level 3	-	-	Actor, MusicalArtist, Painter, SoccerPlayer, GridironFootballPlayer, WinterSportPlayer, Swimmer, BodyOfWater, Mountain, Settlement, Island, Country
Level 4	-	-	AmericanFootballPlayer, IceHockeyPlayer, Lake, City, Town

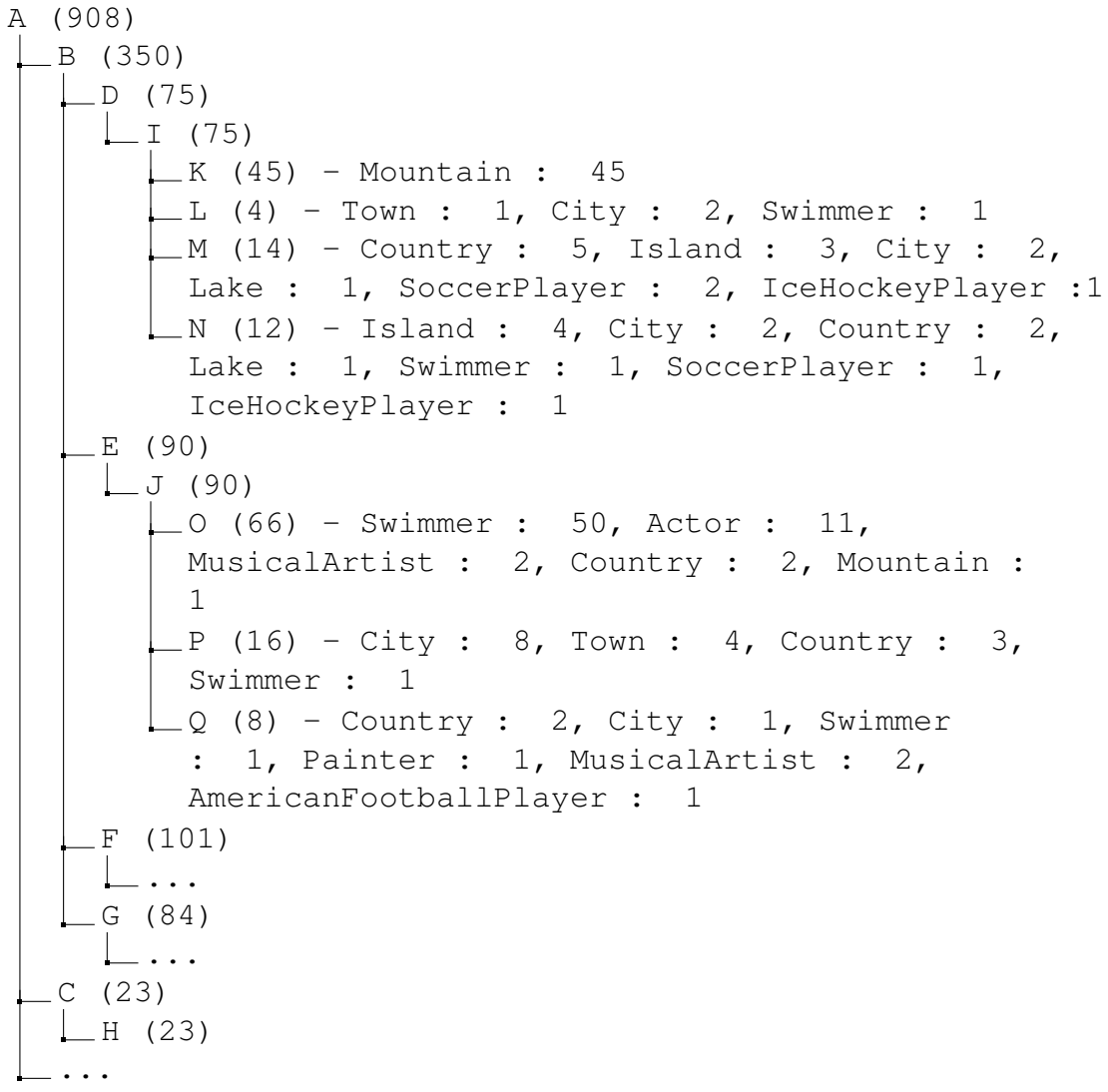


Figure 4.2: Excerpt of Our Induced Tree on the DBpedia Dataset. Numbers in brackets indicate the number of subjects which visited the cluster on its path.

subjects, 103550 triples, 10018 entities, and 190 predicates.

YAGO3-10

The YAGO3-10 dataset was derived from the YAGO3 database [4] which is a knowledge graph derived from Wikipedia and follows the hierarchical class structure of WordNet. As with FB15k-237, we mapped entities to the WordNet taxonomy before selecting the subset defined by classes in Table 4.1. This resulted in a dataset with 11954 subject, 84382 triples, 27572 entities, and 28 relations.

Table 4.2: Method Results (Mean ± Standard Deviation) on the FB15k-237, YAGO3-10, and DBpedia Datasets. Underscore denotes significance at alpha value of 0.05 compared against our model as per t-test.

	FB15k-237		**YAGO3-10**		**DBpedia**	
Method	ARI	NMI	ARI	NMI	ARI	NMI
RDF2VEC						
K-means	.308 ± .012	.567 ± .007	.070 ± .019	.199 ± .017	.223 ± .005	.416 ± .005
OPTICS	.087 ± .000	.283 ± .000	.009 ± .000	.172 ± .000	.001 ± .000	.311 ± .000
Agglom.	.455 ± .000	.601 ± .000	.038 ± .000	.174 ± .000	.236 ± .000	.414 ± .000
Spectral	.539 ± .000	.678 ± .000	.071 ± .000	.218 ± .000	.218 ± .000	.410 ± .000
TransE						
K-means	.405 ± .049	.632 ± .009	.263 ± .009	.367 ± .003	.247 ± .029	.389 ± .024
OPTICS	.031 ± .000	.253 ± .000	.049 ± .000	.150 ± .000	.001 ± .000	.198 ± .000
Agglom.	.491 ± .000	.599 ± .000	.226 ± .000	.337 ± .000	.198 ± .000	.383 ± .000
Spectral	**.658** ± .000	.684 ± .000	**.270** ± .000	.345 ± .000	.057 ± .000	.321 ± .000
DistMult						
K-means	.269 ± .011	.559 ± .013	.174 ± .012	.326 ± .015	.400 ± .008	.587 ± .010
OPTICS	.016 ± .000	.189 ± .000	.029 ± .000	.175 ± .000	.002 ± .000	.184 ± .000
Agglom.	.379 ± .000	.621 ± .000	.202 ± .000	**.382** ± .000	.389 ± .000	.594 ± .000
Spectral	.505 ± .000	.600 ± .000	.035 ± .000	.124 ± .000	.150 ± .000	.478 ± .000
ComplEx						
K-means	.271 ± .020	.562 ± .016	.137 ± .012	.342 ± .009	.462 ± .013	.630 ± .015
OPTICS	.019 ± .000	.202 ± .000	.017 ± .000	.152 ± .000	.002 ± .000	.235 ± .000
Agglom.	.385 ± .000	.630 ± .000	.181 ± .000	.299 ± .000	.442 ± .000	.628 ± .000
Spectral	.563 ± .000	.613 ± .000	.016 ± .000	.204 ± .000	.203 ± .000	.550 ± .000
ConvE						
K-means	.332 ± .031	.619 ± .013	.004 ± .003	.004 ± .001	**.474** ± .019	.612 ± .013
OPTICS	.040 ± .000	.254 ± .000	.012 ± .000	.088 ± .000	.002 ± .000	.238 ± .000
Agglom.	.384 ± .000	.630 ± .000	.003 ± .000	.005 ± .000	.458 ± .000	.614 ± .000
Spectral	.556 ± .000	**.703** ± .000	.002 ± .000	.006 ± .000	.439 ± .000	**.639** ± .000
ExCut	.343 ± .011	.651 ± .002	.130 ± .007	.322 ± .011	.380 ± .016	.595 ± .005
Our Method	.656 ± .005	.669 ± .021	.044 ± .006	.218 ± .002	.406 ± .042	.582 ± .022

DBpedia

The DBpedia dataset was generated by querying DBpedia [5] for random entities belonging to classes on levels 4 and 5 as specified in Table 4.1. Specifically, 75 entities were extracted for each of these classes. Triples where these entities take on the subject role were then queried for, filtering out triples which indicate class membership. This process resulted in 908 subjects, 57191 triples, 31202 entities, and 345 predicates. The impetus for this dataset was to evaluate our model on a hierarchy not rooted in the WordNet taxonomy. The hierarchical relations between DBpedia classes were obtained from the DBpedia ontology mapping which may be found on the DBpeida website[2]. All querying to generate the dataset and ground truth clusters was performed in November of 2021.

4.3.2 Quantitative Evaluation

To quantitatively evaluate our model, we examined the hierarchical clustering of subjects in our induced topic hierarchy. This type of evaluation jointly assesses the quality of the tree structure as well as the allocation of paths along it. Specifically, we ran our model five times on each of the aforementioned datasets using 100 burn-in samples. We then sampled from our learned distributions to obtain a topic hierarchy. We evaluated the quality of the clustering using the Adjusted Rand Index (ARI) [65] and Normalized Mutual Information (NMI) [66] as in previous works [63]. We compared our model against embedding based methods described in the related works section. Pretrained embeddings for these models were obtained from LibKGE[3] [67]. The mean and standard deviations of five runs are summarized in Table 4.2.

Our results indicate that our model is comparable with embedding based approaches. Indeed, the performance of all methods is highly variable with no method clearly outperforming the other. We note our model's underperformance on the YAGO3-10 dataset relative to other methods. We hypothesize that this is due to the high ratio of subjects to triples

[2]http://mappings.dbpedia.org/server/ontology/classes/
[3]https://github.com/uma-pi1/kge

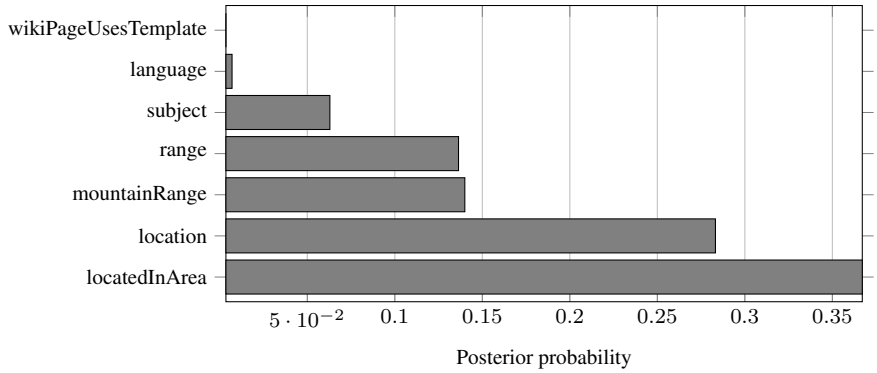


Figure 4.3: Predicates and Their Posterior Distribution for Cluster K on the DBpedia tree as displayed in Figure 4.2.

in this dataset. Such a characteristic results in a low amount of predicates and tags for each subject compared to other datasets. This in turn hinders our model's ability to approximate the true likelihood when calculating the posterior, resulting in lesser performance. Nevertheless, our model is still significantly better than many of the other methods as measured by a t-test. We conclude, therefore, that our model is capable of inducing coherent topic hierarchies on real world knowledge graphs.

4.3.3 Qualitative Evaluation

Cluster allocation is driven by the interaction of predicates and tags. Specifically, each cluster has predicate and tag membership distributions. This allows us to draw interesting observations in that we can describe a cluster by its predicate and tag distributions. This gives us insight into the composition of a cluster. Figure 4.2 provides an excerpt of our induced tree on the DBpedia dataset. On the other hand in Figure 4.3, we provide an example of cluster K's predicate distribution from the DBpedia dataset. We note that this predicate distribution is consistent with the subjects whose path ends at this cluster. Namely, the predicates are consistent with these subjects, i.e., mountains. Furthermore, we can also analyze the distribution of objects to which the predicates are connected to. We highlight this in Figure 4.4 which shows the object distribution for the predicate *locatedInArea* for cluster K. Based on the data that we used, the mountains in cluster K are most probably

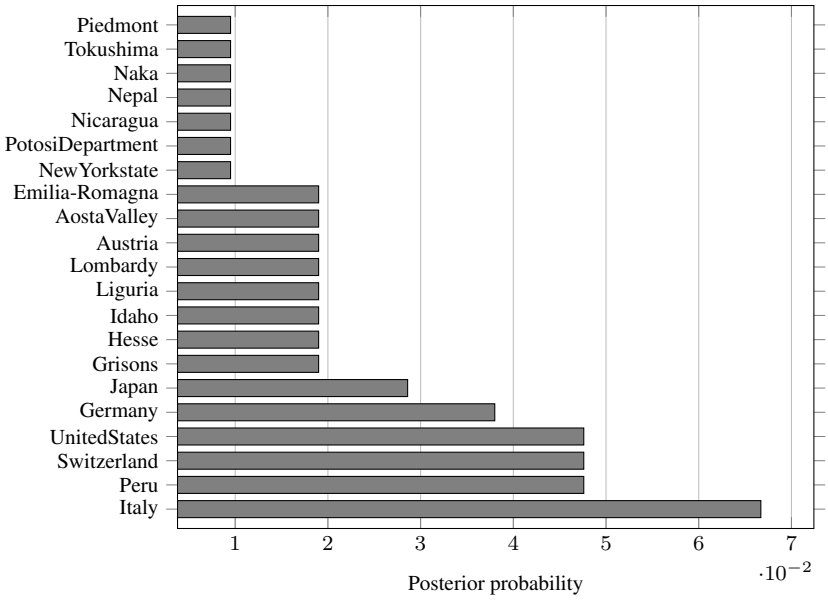


Figure 4.4: Objects' Posterior Distribution for Predicate locatedInArea

located in Italy, Peru, Switzerland, and United States.

4.4 Conclusions

In this paper we propose a model for discovering underlying hierarchical structures in knowledge graphs. For this purpose we adapt a hierarchical topic model used in natural language processing, namely hLDA, to the domain of knowledge graphs. Our model extends hLDA by introducing separate predicate and tag (predicate-object pair) topics, yielding a topic hierarchy consisting of predicate and tag distributions. Knowledge graph subjects take paths through this hierarchy which may be seen as an implicit hierarchical clustering of knowledge graph subjects. This formulation has the added benefit in that it is non-parametric, therefore does not require a priori assumptions about the tree structure other than its depth. To infer our model, we present an efficient Gibbs sampling scheme which leverages the Multinomial-Dirichlet conjugate to integrate out latent probability distributions allowing our model to scale to large datasets. We evaluate our model on three real world datasets and compare against benchmark methods. Our results demonstrate our model's ability to induce coherent topic hierarchies with high quality subject clusterings

and explainable topic predicate and tag memberships.

Chapter 5

Construction of Topic Hierarchy with Subtree Representation for Knowledge Graphs

In this chapter, we adopt a non-parametric probabilistic model, the nested hierarchical Dirichlet process, to the field of knowledge graphs. This model discovers latent subject-specific distributions along paths within the tree. Consequently, the global tree can be viewed as a collection of local subtrees for each subject, allowing us to represent subtrees for each subject and reveal cross-thematic topics.

The primary components of our approach are:

- Data Preprocessing: Knowledge graphs are collections of triples (subject, predicate, object). Our model treats predicates and objects as words that describe subjects, similar to how words describe documents in topic models.

- Generative Model: The nested Hierarchical Dirichlet Process (nHDP) is an extension of the Hierarchical Dirichlet Process (HDP) framework that enables subjects to access the entire tree and learn subject-specific distributions on thematically coherent subjects. Each subject is expected to have a primary path representing core themes, with branches for additional topics. The nHDP introduces two key modifications to the nested Chinese Restaurant Process (nCRP) formulation: (i) each word follows a unique path to a topic, and (ii) each subject possesses a distinct distribution on paths

within a shared tree.

- Inference: Stochastic variational inference is used to estimate the posterior inference of nHDP. This method optimizes local variational parameters for a particular group of individuals and then updates the overall variational parameters using the natural gradient. The approach involves approximating the posterior inference of nHDP through stochastic variational inference. It includes optimizing local variational parameters for a specific group of individuals and updating overall variational parameters using the natural gradient.

- Evaluation: The methodology was applied to FB15k-237, DBpedia, Wikidata, and WebRED datasets for evaluation. The models were assessed based on hierarchy topic quality, simple coverage, subject-based coverage, and vocabulary-based coverage. Evaluations included both quantitative and qualitative analyses of the results.

Key Findings:

1. Our nHDP_KG effectively organizes subject entities into a meaningful hierarchical tree with the inference of the model. Each subject could be represented by subtree shared with the global tree.

2. Quantitative Hierarchical Topic Quality evaluations on benchmark datasets demonstrate the nHDP_KG superior performance in topic coherence. HyperMinor consistently achieved high HTQ values across different datasets, especially when using the DistMult embedding. Traco and SawETM models showed different performance levels, with SawETM generally outperforming Traco. The hLDA model was found to be the least effective in this evaluation.

3. Quantitative on Coverage denotes the nHDP_KG and hLDA can learn the better hierarchical tree than the other three latest models, Traco, SawETM and HyperMiner.

4. Qualitative assessments highlight the coherence and interpretability of the induced subtree.

As mentioned earlier, this paper is still unpublished and intended for publication in the Information Processing and Management.
The primary author, Yujia Zhang, designed and conducted the experiments under the supervision of Prof. Marek Reformat.

Construction of Topic Hierarchy with Subtree Representation for Knowledge Graphs

Abstract: Hierarchy analysis of the Knowledge Graphs aims to discover the latent structure inherent in knowledge base data. Drawing inspiration from topic modeling, which identifies latent themes and content patterns in text corpora, our research seeks to adapt these analytical frameworks to the hierarchical exploration of knowledge graphs.

Specifically, we adopt a non-parametric probabilistic model, the nested Hierarchical Dirichlet Process, to the field of knowledge graphs. This model discovers latent subject-specific distributions along paths within the tree. Consequently, the global tree can be viewed as a collection of local subtrees for each subject, allowing us to represent subtrees for each subject and reveal cross-thematic topics.

We assess the efficacy of this model in analyzing the topics and word distributions that form the hierarchical structure of complex knowledge graphs.

We quantitatively evaluate our model using four common datasets: Freebase, Wikidata, DBpedia, and WebRED, demonstrating that it outperforms the latest neural hierarchical clustering techniques such as Traco, SawETM, and HyperMiner. Additionally, we provide a qualitative assessment of the induced subtree for a single subject.

5.1 Introduction

Knowledge graphs (KGs) are gaining more attention for their potential to integrate with large language models, addressing issues such as hallucination and token limitations. A KG [1] comprises entities and relations, where entities are regarded as nodes and relations as different types of edges. In semantic-oriented interpretation, knowledge graphs are composed of triples in the format $\langle$subject, predicate, object$\rangle$, which serve as the fundamental units of the graph. When representing an ontology within a specific domain $O = C, R$, concepts C correspond to nodes, while relations R between them form the edges. Relations,

in KGs, represent the connections between entities, which can be categorized into various types based on their nature and context. Hierarchical relationships, such as Is-A and Part-Of, capture taxonomic and compositional structures, respectively. Associative relationships describe general associations, while causal relationships define cause-effect links. Temporal and spatial relationships capture time and location dependencies, whereas functional, ownership, and membership relationships depict roles, control, and group inclusion. Additionally, dependency and social relationships represent dependencies and interpersonal or professional connections, respectively. These diverse relationship types enrich the knowledge graph, enabling more sophisticated queries and insights that can drive innovation and enhance user experiences across various applications.

Various factors motivate learning hierarchies from KGs. A key advantage of hierarchical configurations lies in their ability to emulate the innate human tendency to categorize and arrange data for enhanced comprehension and recall, which is particularly beneficial for tasks such as reasoning. The hierarchical structure facilitates the identification of relationships between various concepts and notions. Notably, a hierarchical KG consolidates data in a structured manner, delineating parent-child associations among entities. This framework classifies entities into higher-level groupings and subcategories, unveiling the underlying knowledge hierarchy. Hierarchical KGs streamline the categorization and organization of knowledge, enabling efficient navigation and reinforcement through semantic inference. KGs commonly exhibit a semantic hierarchy, evident in instances like $\langle$England, /location/location/contains, Pontefract/Lancaster$\rangle$ within Freebase [3], showcasing the hierarchical relationship between 'Pontefract/Lancaster' and 'England'. Moreover, another triples repository, DBpedia [5], offers an ontology that hierarchically organizes information about classes and concepts. Although some research endeavors have explored hierarchical structures [10][68], they often necessitate supplementary data or alternative methodologies to gather hierarchical insights. The quest for a method capable of autonomously and effectively replicating the semantic hierarchy remains a persisting challenge.

One crucial consideration in hierarchy analysis for KGs pertains to discerning significant relationships among nodes and identifying hierarchical configurations within the graph. This challenge is further aggravated by the need to accurately capture the hierarchical structure and semantic meanings of the nodes and edges in the KG [69][70]. This procedure frequently involves using algorithms like hierarchical clustering and graph embedding methods to reveal underlying patterns in the data. It is paramount to assess the effectiveness of these algorithms in capturing the intricate relationships and configurations existing in the KG, as this can profoundly influence the accuracy of the hierarchical analysis findings. Evaluating algorithmic efficacy in capturing complex interconnections and elaborate arrangements intrinsic to the KG is vital to ensuring the results' credibility and soundness. Furthermore, gauging the effectiveness of algorithms in capturing the interconnections and arrangements inherent in KGs is indispensable to guarantee the reliability and validity of the analysis outcomes.

In the previous work [71], a subject is described as a single path in the tree by the nested Chinese Restaurant Process, focusing on nonparametric clustering at the subject. Each subject progresses through a singular path in the tree, composing the topics following the path, highlighting a fundamental drawback in the previously suggested approach for subject modeling. However, a specific sequence of topics is expected to encompass the entire thematic content of the subject. Such an approach presents a combinatorial challenge as the nCRP aims to delineate a corpus's thematic essence with increasing levels of specificity. Consequently, similar topics may reappear at different points in the tree to reflect their relevance to the overarching subject theme. For instance, although the tree has already learned the separate topics of "artist" and "writer," when a topic emerges that combines these two, it generates a new composite topic rather than referencing the two pre-existing topics. Due to the exponential increase in nodes, mastering deeper levels of learning becomes demanding, leading to the truncation of nCRP trees, typically at the third tier. As a result, each subject is represented by a few topics that encapsulate its thematic content, potentially blending

multiple themes and resulting in a broader and more complex tree structure during inference. The development of a hierarchical topic model that permits a subject to use topics in various branches of the tree representing a hierarchy is what we are aiming at.

To overcome the mentioned issues, we are investigating novel and manageable methods for extracting hierarchical information from KGs. Our strategy is to adopt a non-parametric probabilistic model for the hierarchical clustering of KG data. It uncovers the latent subject-specific distributions on paths within the hierarchy or tree, a subtree for each subject. An entire tree is a collection of local subtrees representing individual subjects. The method provides the opportunity to identify cross-thematic topics while keeping individual topics in separate subtrees. Therefore, the proposed method clusters subject entities, corresponding predicates, and object entities, and provides insight into their distributions across subtrees. To accomplish that, we adapt the nested Hierarchical Dirichlet process (nHDP).

We quantitatively evaluate our model using four datasets: Freebase, Wikidata, DBpedia, and WebRED, demonstrating that it outperforms the latest neural network-based hierarchical clustering techniques such as Traco, SawETM, and HyperMiner. Additionally, we provide a qualitative assessment of the induced subtrees for subjects.

Here is a summary of our contributions:

- We adapt the nHDP to the knowledge graphs domain by replacing documents with subjects, words with predicates, and objects;
- We evaluate, both quantitative and qualitative, the hierarchical tree by conducting experiments on four real-world datasets such as Freebase, Wikidata, DBpedia, and WebRed;
- We demonstrate the impressive performance of the proposed nHDP_KG method surpassing other neural network-based hierarchical clustering techniques, including Traco, SawETM, and HyperMiner.

5.2 Related Work

5.2.1 Hierarchy of Knowledge Graphs

The hierarchy of knowledge graphs has been a topic of interest in various research studies. hTransM [72] proposes a hierarchy-constrained approach for link prediction in knowledge graphs, emphasizing the importance of hierarchical structures in enhancing prediction performance. HAKE [73] introduces Hierarchy-Aware Knowledge Graph Embedding (HAKE) to model semantic hierarchies in knowledge graphs. Path-based paper [10] creates a hierarchical structure of subject clusters, utilizing taxonomy induction. HamQA [74], a Hierarchy-aware multi-hop Question Answering framework on knowledge graphs, is used to align hierarchical information between question contexts and knowledge graphs. These studies collectively highlight the significance of hierarchy in knowledge graphs and propose various methods to leverage hierarchical structures for improved representation and prediction.

Two categories of hierarchical topic models exist. The first category comprises traditional models like hLDA [75] and its variations [76], which use Gibbs sampling or Variational Inference for parameter estimation. However, these models struggle with large datasets due to high computational costs. The second category consists of neural models such as Traco, SawETM, and HyperMiner. TraCo [77], a novel neural hierarchical topic model designed to address key challenges in topic modeling. TraCo leverages a new Transport Plan Dependency (TPD) approach to model the dependencies between hierarchical topics as optimal transport plans, ensuring sparse and balanced dependencies. This method enhances the affinity between parent and child topics while maintaining diversity among sibling topics. Additionally, TraCo incorporates a Context-aware Disentangled Decoder (CDD), which decodes documents using topics at each level individually and incorporates contextual semantic biases. This ensures that topics at different levels capture distinct semantic granularities, thereby improving the rationality of topic hierarchies. Through ex-

tensive experiments on benchmark datasets, TraCo demonstrates superior performance over state-of-the-art baselines in terms of affinity, rationality, and diversity of topic hierarchies. SawETM [78], a novel hierarchical topic model, addresses limitations of existing models by capturing dependencies and semantic similarities between topics across different layers. Unlike traditional models that assume topics are independent, SawETM uses the Sawtooth Connection technique to link topics across layers, enhancing coherence and depth in topic hierarchies. Additionally, it integrates a robust inference network within a deep hierarchical VAE framework, combining deterministic and stochastic paths to improve text data modeling. This design prevents common issues like posterior collapse and enables SawETM to discover rich, multi-layered topic representations. Experiments demonstrate that SawETM outperforms other models, providing deeper and more interpretable topics and better document representations. HyperMiner [79] introduces a novel method for topic modeling that addresses the shortcomings of Euclidean embedding spaces by utilizing hyperbolic space, renowned for its tree-like characteristics conducive to hierarchical data representation. By measuring distances between words and topics in this space, HyperMiner captures the underlying semantic hierarchies more effectively. General words, which frequently co-occur with others, are positioned near the center, while specific words are placed near the boundary, reflecting their unique contextual relationships. Additionally, hyperbolic space allows for the incorporation of prior structural knowledge, preserving hierarchical relations through distance constraints. Our contributions include leveraging hyperbolic space for enhanced semantic hierarchy mining and designing a graph-based learning scheme to guide the creation of meaningful topic taxonomies. Extensive experiments show that HyperMiner outperforms baseline methods in topic quality and document representation.

5.2.2 Knowledge Graphs Embedding

Knowledge graph embedding is a method to map knowledge graphs from discrete graph space to continuous vector space. It utilizes dense, low-dimensional continuous vectors to

represent triples [80]. Ensuring the vector space is both lower-dimensional and dense is crucial. A dense space enhances computational efficiency for tasks like similarity assessment and embedding model optimization. Conversely, sparse embeddings can complicate computations, resulting in inefficiencies and diminished generalization in subsequent applications. A well-distributed, dense representation preserves the graph's relational structure while enabling scalable and effective processing. It is powerful to allow knowledge graphs easily integrated with deep learning algorithms. TransE [55] is based on the idea that, when translated by valid predicates, subject embeddings should be positioned near object embeddings. This concept is formalized through an objective function that is optimized using stochastic gradient descent to derive the embeddings. DistMult [81] is a bilinear model that captures interactions between entities and relations in a knowledge graph. It represents relations as diagonal matrices and computes scores using the dot product of entity embeddings and relation matrices. ComplEx [82] extends DistMult by modeling relations as complex-valued vectors. It is capable of capturing both asymmetric and symmetric relations in the knowledge graph by utilizing complex-valued embeddings. RotatE [83] is a geometric model that represents relations as rotations in the complex vector space. It captures compositionality and symmetry within the knowledge graph by rotating entity embeddings based on relation embeddings. HolE [84] is a bilinear model that represents relations as circular correlations between the embeddings of entities. It captures the compositional nature of relations in the knowledge graph by computing circular correlations between entity embeddings.

5.3 Hierarchy Construction as Topic Modeling

We define a Knowledge Graph (KG), $\mathcal{G}$, as a collection of triples. Each fact triple is composed of a subject entity s that is linked to an object entity o via a predicate p. Formally, $\mathcal{G} = \{\langle s, p, o\rangle \in \mathcal{S} \times \mathcal{P} \times \mathcal{O}\}$ where $\langle s, p, o\rangle$ is a triple, and $\mathcal{S}$, $\mathcal{P}$, and $\mathcal{O}$ are the sets of subjects, predicates, and objects in $\mathcal{G}$, respectively. KGs are rarely bipartite regarding $\mathcal{S}$

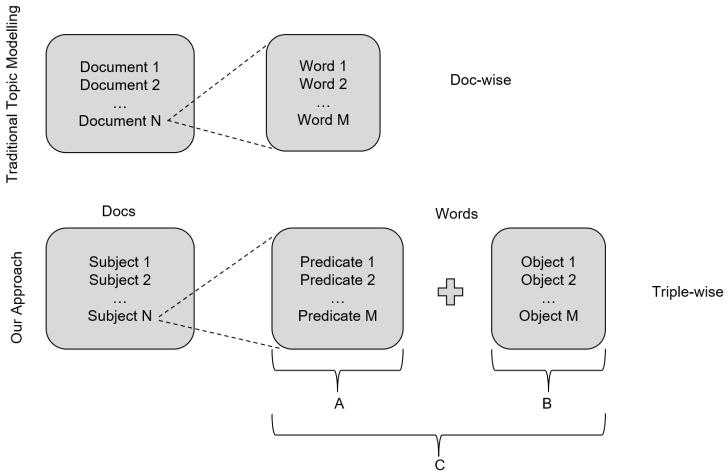


Figure 5.1: 'Conversion' of triple components into documents and related to them words: a given subject (document) is represented by both predicates and objects (words) of all triples containing the subject.

and $\mathcal{O}$. In other words, entities $\mathcal{E} = \mathcal{S} \cup \mathcal{O}$ can take on the role of both subjects and objects in $\mathcal{G}$, thus $\mathcal{S} \cap \mathcal{O} \neq \emptyset$.

The essential aspect of the proposed method is to treat a hierarchy construction task as topic modeling. To accomplish that, we convert graph triples into documents and words. As in Figure 5.1, a subject, a document in topic modeling terminology, is described by all its predicates $\langle p \rangle$ and objects $\langle o \rangle$ that are words in the topic modeling problem. In the presented work, we use the following convention: a subject s_i is described by its words $w_i \in \mathcal{W}_i$, here $\mathcal{W}_i$ is the set of words describing s_i. The set of all subjects is denoted as $\mathcal{S} \subseteq \mathcal{E}$, which means it is a subset of entities. Words, denoted as $w := \langle p \rangle, \langle o \rangle$ belong to the set of all words, called vocabulary $\mathcal{V}$ which means $\mathcal{W}_i \subseteq \mathcal{V}$.

We aim to develop a hierarchical representation of a knowledge graph (KG) in which the global tree structure captures general topics at the root level and specific topics at the leaf level. Nodes in this tree represent collections of entities with shared semantics. Unlike nCRP, where topics are restricted to a single path, our approach, called hereafter nHDP_-KG, allows each subject entity to access the entire tree. A subject-specific distribution over paths assigns higher probabilities to particular subtrees, reflecting the relevance of

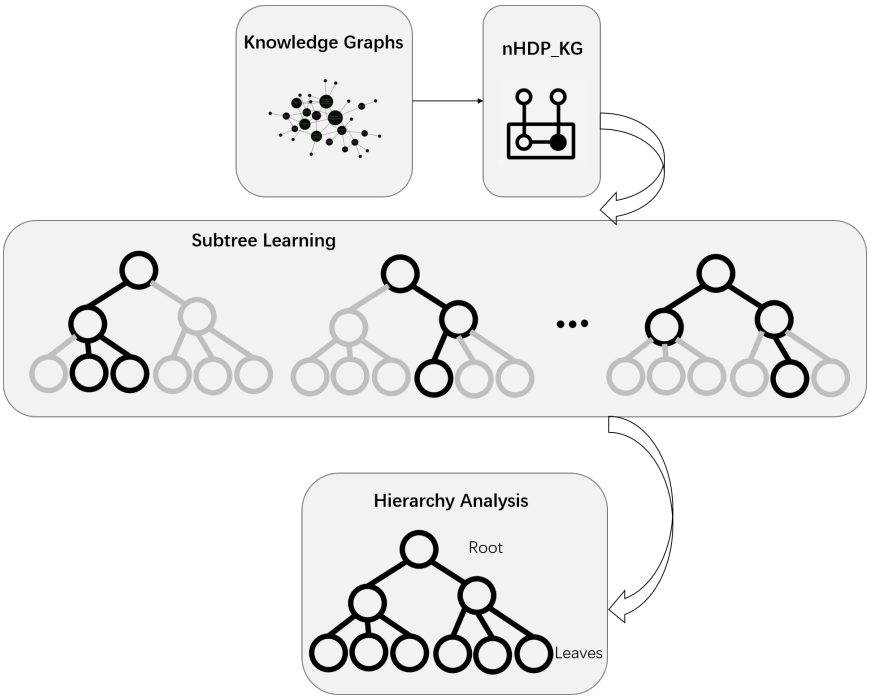


Figure 5.2: nHDP_KG Workflow

certain topics to the subject. It reveals subject-specific distributions on hierarchical paths, with subtrees for each subject. A complete tree comprises localized subtrees for individual subjects. This approach facilitates the identification of cross-thematic topics while preserving the distinctiveness of individual topics in separate subtrees. Consequently, the method clusters subject entities, associated words, offering insight into their distributions across subtrees. To achieve this, we employ the nested hierarchical Dirichlet process (nHDP). The overall workflow is depicted in Figure 5.2.

5.4 Model Description

The proposed model uses a nested Hierarchical Dirichlet Process (nHDP), which extends the Hierarchical Dirichlet Process (HDP), a Bayesian nonparametric model used in machine learning, to infer distributions over distributions. The nHDP expands upon the HDP by allowing a flexible tree-structured exchangeable random partition to model hierarchical clustering, which is beneficial for applications requiring hierarchically organized variable

depth clusters.

Let us describe nHDP via a sequence of short descriptions of necessary concepts.

5.4.1 Dirichlet Process

The Dirichlet Process (DP) [85, 86] is a stochastic process used in Bayesian nonparametric models to represent distributions over data. These mixture models partition a data set into categories based on statistical characteristics that all members within a cell share. The parameters of the mixture and a reasonable number of traits for describing the data can be learned using Dirichlet process priors. Mathematically, it is defined as:

$$W_n|\varphi_n \sim F_W(\varphi_n), \varphi_n|G \overset{\text{iid}}{\sim} G, G = \sum_{i=1}^{\infty} \alpha_i \delta_{\theta_i}. \tag{5.1}$$

where the data $W_1, ..., W_N$ are represented with a family of distributions F_W and the corresponding parameters $\varphi_1, ..., \varphi_N$. These parameters are drawn from the discrete, theoretically infinite distribution G, which the DP permits to exist. The data W is divided as a result of this discreteness in accordance with how the atoms θ_i are distributed among the chosen parameters φ_n. $G \sim DP(\alpha G_0)$ where $\alpha > 0$ is a scaling parameter, and G_0 is a continuous base probability measure.

5.4.2 Hierarchical Dirichlet Process

The Hierarchical Dirichlet Process (HDP) [87, 86] is a multi-level variation of the DP. It relies on the notion that the base distribution on the continuous space might be discrete, which is advantageous since a discrete distribution enables the placement of probability mass on a subset of atoms through multiple draws from the DP in advance. As a result, various data groups with different probability distributions can share the same atoms. It is necessary to create a distinct base, but the atoms are currently unknown. By obtaining the base from a DP prior, the HDP models these atoms. As a result, groupings $d = 1, ..., D$ go through a hierarchical procedure:

$$G_d|G \overset{\text{iid}}{\sim} DP(\beta G), G \sim DP(\alpha G_0) \tag{5.2}$$

Similar to the DP, inference requires explicit representations of the HDP. The representation is based on two levels of Sethuraman's stick-breaking construction [88]. Sample the discrete G as in the following construction

$$G = \sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1}(1 - V_j)\delta_{\theta_i}, V_i \overset{\text{iid}}{\sim} Beta(1, \alpha), \theta_i \overset{\text{iid}}{\sim} G_0 \tag{5.3}$$

here V_i can be understood as the percentage of a unit-length stick that is broken from the rest and G_0 is continuous base distribution. And then sample G_d according to the following equation:

$$G_d = \sum_{i=1}^{\infty} V_i^d \prod_{j=1}^{i-1}(1 - V_j^d)\delta_{\phi_i}, V_i^d \overset{\text{iid}}{\sim} Beta(1, \beta), \phi_i \overset{\text{iid}}{\sim} G \tag{5.4}$$

This form is the same as the previous equation with the crucial difference that G is discrete, which causes atoms ϕ_i to repeat. All random variables in this representation are i.i.d., which helps variational inference techniques.

5.4.3 Adapted Nested Hierarchical Dirichlet Process

The nHDP_KG, an adapted framework from nHDP [86], allows each subject to use the entire hierarchical structure while acquiring subject-specific distributions over semantically related topics. Each subject is represented by a primary trajectory corresponding to its core themes, along with branches that incorporate additional topics. The process of constructing a subject's topic distribution is divided into two stages: first, developing the subject's distribution over paths within the tree, and second, defining the word distribution conditioned on reaching specific nodes along these paths.

Global tree: distribution on paths

All subject entities share a global tree drawn according to the stick-breaking construction via nHDP.

$$G_{\boldsymbol{i}_l} = \sum_{j=1}^{\infty} V_{\boldsymbol{i}_l,j} \prod_{m=1}^{j-1}(1 - V_{\boldsymbol{i}_l,m})\delta_{\theta_{\boldsymbol{i}_l,j}}, V_{\boldsymbol{i}_l,j} \overset{\text{iid}}{\sim} Beta(1, \alpha), \theta_{\boldsymbol{i}_l,j} \overset{\text{iid}}{\sim} G_0 \tag{5.5}$$

This tree is just an endless collection of Dirichlet processes with a continuous base distribution G_0 and an inter-DP transition rule. This rule states that a path is followed from a root Dirichlet process $G_{\boldsymbol{i}_0}$ by drawing $\varphi_{l+1} \sim G_{\boldsymbol{i}_l}$ for $l = 0, 1, 2, ...$, where $\boldsymbol{i}_0$ is a constant root index and $\boldsymbol{i}_l = (i_1, ..., i_l)$ indexes the DP connected to the topic $\varphi_l = \theta_{\boldsymbol{i}_l}$. With the nested HDP, we use each Dirichlet process in a global tree as a base distribution for a local DP drawn independently for each entity rather than following pathways according to the global tree.

Then for each subject s, a local tree T_s will be constructed. For each $G_{\boldsymbol{i}_l} \in T$, the corresponding $G^s_{\boldsymbol{i}_l} \in T_s$ can be drawn as per the DP,

$$G^s_{\boldsymbol{i}_l} \sim DP(\beta G_{\boldsymbol{i}_l}) \tag{5.6}$$

as was discussed, the $G^s_{\boldsymbol{i}_l}$ In contrast $G_{\boldsymbol{i}_l}$, will have various probability weights on the same atoms. Therefore, the probability of a path in the tree T_s will vary for each subject, it will have the same nodes as T while each subject will have its unique distribution on the tree. A stick-breaking construction to represent this subject-specific DP:

$$G^s_{\boldsymbol{i}_l} = \sum_{j=1}^{\infty} V^s_{\boldsymbol{i}_l,j} \prod_{m=1}^{j-1} (1 - V^s_{\boldsymbol{i}_l,m}) \delta_{\theta^s_{\boldsymbol{i}_l,j}}, V^s_{\boldsymbol{i}_l,j} \overset{\text{iid}}{\sim} Beta(1, \beta), \theta^s_{\boldsymbol{i}_l,j} \overset{\text{iid}}{\sim} G_{\boldsymbol{i}_l} \tag{5.7}$$

The HDP samples distribution $G^s_{\boldsymbol{i}_l}$ with base measure $G_{\boldsymbol{i}_l}$. This results in a subject-specific distribution over paths in the globally shared tree T.

The key difference from the HTM is that now each subject has its own distribution over the shared hierarchical topic structure rather than all subjects sharing the same tree distribution.

Local tree: Generating a subject

A technique for choosing word-specific paths that are thematically compatible with the tree T_s for subject s, meaning they frequently reuse the same path while allowing for off-shoots.

The process goes as follows:

- For each node $\boldsymbol{i}_l$ in the tree T, draw a beta-distributed random variable $U_{s,\boldsymbol{i}_l} \sim Beta(\gamma_1, \gamma_2)$ that acts as a stochastic switch.

- To generate a word n in a subject s: Start at the root node and recursively traverse down the tree according to $G^s_{\boldsymbol{i}_l}$ until reaching some node $\boldsymbol{i}_l$. With probability $U_{s,\boldsymbol{i}_l}$ emit the topic $\theta_{\boldsymbol{i}_l}$ at this node. Otherwise, continue traversing down the tree according to $G^s_{\boldsymbol{i}_l}$.

This recursive process generates a path and selects a node $\boldsymbol{i}_l$ and topic θ_{i_l} for each word. The probability of a word being assigned topic $\theta_{\boldsymbol{i}_l}$ is:

$$p(\varphi_{s,n} = \theta_{\boldsymbol{i}_l} | T_s, U_s) = [\prod_{m=0}^{l-1} G^s_{\boldsymbol{i}_m}(\theta_{\boldsymbol{i}_{m+1}})][U_{s,\boldsymbol{i}_l} \prod_{m=1}^{l-1} (1 - U_{s,\boldsymbol{i}_m})] \tag{5.8}$$

Where the first term is the probability of path $\boldsymbol{i}_l$ according to G^s, and the second term is the probability of selecting topic $\theta_{\boldsymbol{i}_l}$ at that node.

This process allows each word to follow its own path in the tree according to the subject-specific distribution G^s, capturing unique topic combinations within a subject.

5.4.4 Stochastic Variational Inference

A method of stochastic variational inference is applied to approximate the posterior inference of the nested Hierarchical Dirichlet Process (nHDP). The process involves the optimization of local variational parameters for a specific group of individuals, followed by a progression along the natural gradient of the overall variational parameters. More inference details can be found in [86].

Greedy Subtree Selection

For each subject, a subtree is selected from the global tree T using a greedy algorithm that maximizes the variational objective function. Starting from the root, nodes are sequentially added based on their activation status, where an activated node is one whose parent is in the subtree but the node itself is not.

Stochastic Updates for Local Variables

Index Pointer. $z_{\boldsymbol{i},j}^{(s)}$ is index pointer to atom in global DP $G_{\boldsymbol{i}}$ for jth break in $G_{\boldsymbol{i}}^{(s)}$. The prior is $q(z_{\boldsymbol{i},j}^{(s)}) = \delta_{z_{i,j}^{(s)}}(k), \quad k = 1, 2, \ldots$. The updates for it are:

$$\begin{aligned} I_{s,t+1} &\leftarrow I_{s,t} \cup \{\boldsymbol{i}^*\}, \\ i^* &= \arg\max_{i' \in S_{s,t}} L_{\text{obj}}, \\ L_{\text{obj}} &= \sum_{n=1}^{N_s} \mathbb{E}_q \left[\ln p(W_{s,n} \mid c_{s,n}, \theta)\right] \\ +\mathbb{E}_q \left[\ln p(c_{s,n}, z^{(s)} \mid V, V_s, U_s)\right] &- \mathbb{E}_q \left[\ln q(c_{s,n})\right] \end{aligned} \tag{5.9}$$

Topic Indicator. $c_{s,n}$ is the topic indicator for word n in subject s, $q(c_{s,n}) = \text{Discrete}(c_{s,n}|\nu_{s,n})$ is the prior distribution. The variational distribution on the path for word $W_{s,n}$ is

$$\nu_{s,n}(\boldsymbol{i}) \propto \exp\left\{\mathbb{E}_q[\ln \theta_{\boldsymbol{i},W_{s,n}}] + \mathbb{E}_q[\ln \pi_{s,\boldsymbol{i}}]\right\}, \tag{5.10}$$

where the prior term $\pi_{s,\boldsymbol{i}}$ is the tree-structured prior of the nHDP,

$$\begin{aligned} \pi_{s,i} = &\left[\prod_{(\boldsymbol{i}',i) \subseteq \boldsymbol{i}} \prod_j \left(V_{\boldsymbol{i}',j}^{(s)} \prod_{m<j} (1 - V_{\boldsymbol{i}',m}^{(s)})\right)^{\mathbb{I}(z_{\boldsymbol{i}',j}^{(s)} = i)}\right] \\ &\times \left[U_{s,\boldsymbol{i}} \prod_{\boldsymbol{i}' \subset \boldsymbol{i}} (1 - U_{s,\boldsymbol{i}'})\right]. \end{aligned} \tag{5.11}$$

Stick Proportion. $V_{\boldsymbol{i},j}^{(s)}$ is the stick proportion for local DP for node $\boldsymbol{i}$, $q(V_{\boldsymbol{i},j}) = \text{Beta}(V_{\boldsymbol{i},j}^{(s)}|u_{\boldsymbol{i},j}^{(s)}, v_{\boldsymbol{i},j}^{(s)})$ is the prior distribution. The variational parameter updates for the subject-level stick-breaking proportions are

$$\begin{aligned} u_{\boldsymbol{i},j}^{(s)} &= 1 + \sum_{\boldsymbol{i}':(\boldsymbol{i},j) \subseteq \boldsymbol{i}'} \sum_{n=1}^{N_s} \nu_{s,n}(\boldsymbol{i}'), \\ v_{\boldsymbol{i},j}^{(s)} &= \beta + \sum_{\boldsymbol{i}':\boldsymbol{i} \subset \boldsymbol{i}'} \mathbb{I}\left(\bigcup_{m>j} \{z_{\boldsymbol{i},m}^{(s)} = \boldsymbol{i}'(l+1)\}\right) \sum_{n=1}^{N_s} \nu_{s,n}(\boldsymbol{i}'). \end{aligned} \tag{5.12}$$

In textual terms, the statistic concerning the first parameter denotes the anticipated quantity of words in subject s that either traverse or halt at node $(\boldsymbol{i}, j)$. The statistic related to

the second parameter signifies the anticipated quantity of words from subject s whose trajectories traverse the same ancestor $\boldsymbol{i}$, but subsequently transition to a node with an index exceeding j based on the indicators $z_{\boldsymbol{i},m}^{(s)}$ from the subject-level stick-breaking construction of $G_{\boldsymbol{i}}^{(s)}$.

Switch Probablity. $U_{s,\boldsymbol{i}}$ is the switch probability for node $\boldsymbol{i}$, $q(U_{s,\boldsymbol{i}}) = \text{Beta}(U_{s,\boldsymbol{i}}|a_{s,\boldsymbol{i}}, b_{s,\boldsymbol{i}})$ is the prior distribution. The variational parameter updates for the switching probabilities are similar to those of the subject-level stick-breaking process, but collect the statistics from $\nu_{s,n}$ in a slightly different way,

$$a_{s,\boldsymbol{i}} = \gamma_1 + \sum_{n=1}^{N_s} \nu_{s,n}(\boldsymbol{i}), \tag{5.13}$$

$$b_{s,\boldsymbol{i}} = \gamma_2 + \sum_{\boldsymbol{i}':\boldsymbol{i}\subset\boldsymbol{i}'} \sum_{n=1}^{N_s} \nu_{s,n}(\boldsymbol{i}'). \tag{5.14}$$

The statistic for the first parameter represents the anticipated word count related to the topic at node $\boldsymbol{i}$. On the other hand, the statistic for the second parameter indicates the expected number of words that transit through node $\boldsymbol{i}$ without ending there. The first parameter statistic denotes the projected word quantity associated with the topic at node $\boldsymbol{i}$. The second parameter statistic signifies the anticipated number of words that traverse through node $\boldsymbol{i}$ but do not stop there.

Stochastic Updates for Global Variables

Once the subtree is selected and the local subject-specific variational parameters are updated for each subject s in mini-batch m, we adjust the global q distribution parameters, including the topics $\theta_{\boldsymbol{i}}$ and the global stick-breaking proportions $V_{\boldsymbol{i}_l,j}$, using the natural gradient.

Topic Probalility. $\theta_{\boldsymbol{i}}$ is topic probability vector for node $\boldsymbol{i}$, $q(\theta_{\boldsymbol{i}}) = \text{Dirichlet}(\theta_{\boldsymbol{i}}|\lambda_{\boldsymbol{i},1}, \ldots, \lambda_{\boldsymbol{i},\mathcal{V}})$ $q(\theta_{\boldsymbol{i}})$ is the prior distribution. To update the Dirichlet q distributions on each topic $\theta_{\boldsymbol{i}}$ stochastically, start by constructing the vector $\lambda_{\boldsymbol{i}}'$ containing the necessary statistics based

on the information in sub-batch m.

$$\lambda'_{\boldsymbol{i},w} = \frac{S}{|C_s|} \sum_{s \in C_s} \sum_{n=1}^{N_s} \nu_{s,n}(\boldsymbol{i}) \mathbb{I}\{W_{s,n} = w\}, \tag{5.15}$$

For each w from 1 to $\mathcal{V}$, the vector represents the anticipated quantity of words with index w derived from topic $\theta_{\boldsymbol{i}}$ across subjects indexed by C_s. The modification for the corresponding q distribution is then calculated.

$$\lambda^{m+1}_{\boldsymbol{i},w} = \lambda_0 + (1 - \rho_m)\lambda^m_{\boldsymbol{i},w} + \rho_m \lambda'_{\boldsymbol{i},w}. \tag{5.16}$$

Stick Proportion. $V_{\boldsymbol{i},j}$ is stick proportion for the global DP for node $\boldsymbol{i}$. $q(V_{\boldsymbol{i},j}) = \text{Beta}(V_{\boldsymbol{i},j}|\tau^{(1)}_{\boldsymbol{i},j}, \tau^{(2)}_{\boldsymbol{i},j})$ is the prior distribution. The sufficient statistics for the q distribution on $V_{\boldsymbol{i}_l,j}$ from the subjects in mini-batch m are gathered along with $\theta_{\boldsymbol{i}}$. This process is done as a first step. This step is essential for the estimation of the q distribution on $V_{\boldsymbol{i}_l,j}$. The sufficient statistics collected from the subjects in mini-batch m are crucial for this estimation.

$$\tau'_{\boldsymbol{i}_l,j} = \frac{S}{|C_s|} \sum_{s \in C_s} \mathbb{I}\{\boldsymbol{i}_l \in \mathcal{I}_s\}, \tag{5.17}$$

$$\tau''_{\boldsymbol{i}_l,j} = \frac{S}{|C_s|} \sum_{s \in C_s} \sum_{j > i_l} \mathbb{I}\{(pa(\boldsymbol{i}_l), j) \in \mathcal{I}_s\}. \tag{5.18}$$

The initial value increases the count of subjects in mini-batch m containing atom $\theta_{(\boldsymbol{i},j)}$ in their subtree. The subsequent value increases the occurrence of an atom with a higher index value in the same Dirichlet Process used by a subject in sub-batch m. The global variational parameters are updated based on these values.

$$\tau^{(1)}_{\boldsymbol{i}_l,j}(m+1) = 1 + (1 - \rho_m)\tau^{(1)}_{\boldsymbol{i}_l,j}(m) + \rho_m \tau'_{\boldsymbol{i}_l,j}, \tag{5.19}$$

$$\tau^{(2)}_{\boldsymbol{i}_l,j}(m+1) = \alpha + (1 - \rho_m)\tau^{(2)}_{\boldsymbol{i}_l,j}(m) + \rho_m \tau''_{\boldsymbol{i}_l,j}. \tag{5.20}$$

5.5 Experiment Setup

The proposed methodology has been evaluated on FB15k-237, DBpedia, Wikidata, and WebRED datasets. Evaluation of these models was conducted based on hierarchy topic quality, simple coverage, subject-based coverage, and vocabulary-based coverage.

5.5.1 Dataset

In the conducted experiments four dataset have been used. A short characteristic of each set is presented below.

Dataset	**# Subjects**	**# Entities**	**# Relations**	**# Triplets**
FB15k-237	13781	14541	237	272115
FB15k-237subset	10000	22982	237	197497
Wikidata subset	10000	27608	374	44896
DBpedia	908	31222	345	57192
WebRED subset	10000	16595	428	45712

Table 5.1: Data Statistics

FB15k-237: The FB15k-237 dataset [62], derived from the FB15k dataset[55], was created by eliminating duplicate and reverse triples. It is based on a Freebase version from approximately 2013. It consists of 272,115 triples built using 14,541 different entities and 237 predicates. Due to the resource restriction for a fair comparison, we randomly extracted triples containing 10,000 subjects for our hierarchical clustering analysis. This process has resulted in a dataset with 10,000 subjects, 197,497 triples, 22,982 entities, and 237 predicates.

DBpedia: The DBpedia dataset [5] was created by randomly querying DBpedia for entities in various classes such as 'Politician', 'CelestialBody', 'MusicalWork', 'WrittenWork', 'Film', 'Scientist', 'Artwork', 'NaturalPlace', 'Building', 'Infrastructure', 'PopulatedPlace', 'Artist', 'Software', and 'Athlete'. A total of 75 entities were extracted for each class, resulting in 908 subjects, 57191 triples, 31202 entities, and 345 predicates. The goal of this dataset was to test a model on a hierarchy different from WordNet taxonomy. The hierarchy of DBpedia classes were obtained from the DBpedia ontology mapping available on the DBpedia website[1].

[1] http://mappings.dbpedia.org/server/ontology/classes/

Wikidata: The dataset Wikidata5m [89] is a large-scale KG containing millions of entities, each aligned with a corresponding Wikipedia page. This dataset combines information from Wikidata and Wikipedia, allowing for the evaluation of link prediction on entities that have not been seen before. The dataset is provided in three parts: the graph, a corpus, and aliases. The inductive data splits are used in our paper. Due to the resource restriction for a fair comparison, we randomly extracted triples containing 10,000 subjects for our hierarchical clustering analysis. This process yielded a dataset with 10,000 subjects, 44,896 triples, 27,608 entities, and 374 predicates.

WebRED: WebRED [90] is a dataset designed for relation extraction, sourced from various publicly available internet texts covering diverse domains and writing styles. The dataset includes approximately 200 million weakly supervised examples for supervised pre-training and 110,000 human-annotated examples for fine-tuning and model evaluation. To ensure a fair comparison under resource constraints, 10,000 subjects were randomly selected for hierarchical clustering analysis. The resulting dataset from this process consists of 10,000 subjects, 45,712 triples, 16,595 entities, and 428 predicates.

5.5.2 Evaluation Metrics

Hierarchy Topic Quality

A comprehensive method for constructing coherence measures in hierarchical tree structures to enhance the interpretability of hierarchical topics has been introduced in [91]. The topic coherence is calculated within branches and levels of the tree. It leads to two measures: Branch Topic Quality (BTQ) and Level Topic Quality (LTQ) – they are aggregated to obtain Hierarchical Topic Quality (HTQ), which serves as a metric for assessing and benchmarking topic models.

The mentioned above measures are an extension of the unifying framework presented in [92]. The original framework defines coherence measures using four sets: segmentation (S), confirmation measure (M), probability estimation (P), and aggregation (Σ), creating a

configuration space $C = S \times M \times P \times \Sigma$. The authors of [91] made a modification by adding a hierarchical word set (H_W). The new configuration space is $C^{(h)} = H_W \times S \times M \times P \times \Phi$. They also adapted the input of the segmentation set to account for hierarchical structures. In this method, hierarchical word sets (H_W) include words (W_B) from both parent and child nodes at a specific branch b (W_B^b) and words (W_L) from all nodes at a particular level l (W_L^l).

For our experiments, we use an open-source implementation of the configuration space C_V[2] for computing coherence of words representing a topic. The C_V measure's segmentation (S_{set}^{one}), probability estimation (P_{sw}), and confirmation measure (Φ) are utilized, using a sliding window of size 110 and combining indirect cosine measures with Normalized Pointwise Mutual Information (NPMI) [91]. The confirmation measures are aggregated with a diversity term to produce BTQ and LTQ as below. The arithmetic mean of BTQ and LTQ yields the hierarchical topic quality (HTQ) for the entire topic model. The equations are summarized as below:

$$BTQ = \frac{\sum_{b=1}^{B} \phi_i^{W_B} \cdot d_b}{B} \tag{5.21}$$

$$LTQ = \frac{\sum_{l=1}^{L} \phi_i^{W_L} \cdot d_l}{L} \tag{5.22}$$

$$HTQ = \frac{BTQ + LTQ}{2} \tag{5.23}$$

$$S_{one}^{set} = (W', W^*) | W' = w_i; w_i \in W; W^* = W \tag{5.24}$$

$$P_{sw}(S_i) = log \frac{P(W', W^*) + \epsilon}{P(W') * P(W^*)} \tag{5.25}$$

$$\phi(\vec{u}, \vec{w}) = \frac{\sum_{i=1}^{|W|} u_i \cdot w_i}{\|\vec{u}\|_2 \|\vec{w}\|_2} \tag{5.26}$$

[2]https://github.com/piskvorky/gensim/blob/develop/gensim/models/coherencemodel.py

Coverage

To evaluate constructed hierarchies more thoroughly, we propose a set of new coverage measures based on the work presented in [93]. The proposed metrics are simple coverage, coverage based on subjects, and coverage based on vocabulary.

A coverage evaluation process requires some preprocessing. If a given topic t_z is represented by an ordered set of words W^z, we replace in a subject (set of words describing it, Section 5.3) all occurrences of top words from W^z with the first word $w^z_{top} \in W^z$.

Simple Coverage. The hierarchy tree should organize topics from general to specific. Higher-level topics, closer to the root node, should encompass a broad range of subjects, while lower-level topics, closer to the leaf nodes, should exhibit narrower coverage. The coverage is calculated as follows:

$$Cov(L) = \frac{1}{k} \sum_{z} PMI_c(t_z) \tag{5.27}$$

$$PMI_c(t_z) = \frac{\#(w^z_{top})}{total\_num\_subjs} \tag{5.28}$$

where t_z is a topic z, $\#(w^z_{top})$ is the number of times the word w^z_{top} appears in a subject, and k is the number of topics at level L. The average coverage of all topics at the same level reflects the model's coverage capability. As the tree becomes deeper, with each successive level moving closer to the leaves, the coverage score should decrease accordingly.

Coverage based on subjects: Based on the idea of Simple Coverage we focus on evaluating coverage of all topics at a given level. The value of the coverage is calculating using the following equations

$$Cov(L) = \frac{1}{k} \sum_{z} PMI_s(t_z) \tag{5.29}$$

$$PMI_s(t_z) = \frac{1}{V} \sum_{j=1}^{V} log \frac{p(w^z_{top}, w_j)}{p(w^z_{top})p(w_j)} \tag{5.30}$$

$$p(w^z_{top}, w_j) = \frac{\#(w^z_{top}, w_j)}{total\_num\_subjs} \tag{5.31}$$

$$p(w^z_{top}) = \frac{\#(w^z_{top})}{total\_num\_subjs} \tag{5.32}$$

$$p(w_j) = \frac{\#(w_j)}{total\_num\_subjs} \tag{5.33}$$

where $p(w^z_{top}, w_j)$ is the frequency of co-occurrence of the first word w^z_{top} with a word w_j across all subjects; $p(w^z_{top})$ and $p(w_j)$ are the number of occurrences of the word w^z_{top} and w_j, respectively, and k is the number of topics at level L. Pointwise Mutual Information (PMI_s) calculates the similarity between word pairs across all subjects. The PMI score is computed for each topic in the tree. The average coherence of all topics at the same level is calculated to assess the model's coverage.

As a tree goes deeper, the coverage score is expected to decrease with lower levels, aligning with the assumption that more specific topics have narrower coverage.

Coverage based on vocabulary In contrast to the two previous approaches, from the perspective of words describing the specific topics, the number of top words in higher-level topics should be less than that in lower-level topics. Now the equations are:

$$Cov(L) = \frac{1}{k}\sum_z PMI_v(t_z) \tag{5.34}$$

$$PMI_v(t_z) = \frac{1}{V}\sum_{j=1}^{V} log\frac{p(w^z_{top}, w_j)}{p(w^z_{top})p(w_j)} \tag{5.35}$$

$$p(w^z_{top}, w_j) = \frac{\#(w^z_{top}, w_j)}{total\_num\_vocab} \tag{5.36}$$

$$p(w^z_{top}) = \frac{\#(w^z_{top})}{total\_num\_vocab} \tag{5.37}$$

$$p(w_j) = \frac{\#(w_j)}{total\_num\_vocab} \tag{5.38}$$

where this time $p(w^z_{top}, w_j)$ is the frequency of co-occurrence of the first word w^z_{top} and a word w_j in the whole vocabulary; $p(w^z_{top})$ and $p(w_j)$ is the number of times the words w^z_{top} and w_j occurs, and k is the number of topics at level L. Pointwise mutual information

(PMI_v) is employed to calculate the similarity of pairs in the whole vocabulary. The PMI score for each topic in the tree is computed.

The average coherence of all topics at the same level is used to reflect the coverage of the model. As the tree goes deeper, the level decrease, the coverage score should increase as the assumption.

5.5.3 Experiment environment

For our experiments, the nHDP_KG model was configured with a batch size of 20 and 1000 iterations. The hyperparameter beta0, which controls the Dirichlet base distribution, was set to 1. In practice, the number of topics is dataset-dependent and varies based on the desired depth of topic exploration.

5.6 Experiment Results

The results of experiments have been evaluated quantitatively and qualitatively.

5.6.1 Quantitative Evaluation

Hierarchical Topic Quality The comparison of the performance of the proposed method for constructing a hierarchy based on KGs with other approaches on the diverse datasets (FB15k-237, WikiData, DBpedia, and WebRED) is showed in Table 5.2. nHDP_KG and hLDA methods do not require embedding of triples while the latest three models were used with five different embeddings: TransE, DistMult, ComplEx, RotatE, HoIE. The used metrics are BTQ (Branch Topic Quality), LTQ (Level Topic Quality), and HTQ (Hierarchical Topic Quality).

The nHDP_KG consistently displayed superior performance across all datasets, with HTQ values ranging from 0.795 to 0.484, demonstrating the highest performance on FB15k-237 and the lowest on WebRED and surpassing all other models. In contrast, hLDA exhibited notably lower performance with HTQ values of 0.627 on FB15k-237 and 0.244 on

Models	Embeddings	FB15k-237			WikiData			DBpedia			WebRED		
		BTQ	LTQ	HTQ	BTQ	LTQ	HTQ	BTQ	LTQ	HTQ	BTQ	LTQ	HTQ
nHDP_KG		**0.785**	**0.805**	**0.795**	**0.671**	**0.647**	**0.659**	**0.502**	**0.445**	**0.473**	**0.489**	**0.479**	**0.484**
Hlda		0.662	0.592	0.627	0.299	0.228	0.263	0.413	0.424	0.418	0.269	0.220	0.244
Traco	TransE	0.385	0.385	0.385	0.015	0.023	0.019	0.443	0.449	0.446	0.190	0.190	0.190
	DistMult	0.337	0.341	0.339	0.075	0.150	0.112	0.396	0.420	0.408	0.267	0.277	0.272
	ComplEx	0.427	0.427	0.427	0.012	0.017	0.014	0.369	0.390	0.380	0.379	0.414	0.396
	RotatE	0.322	0.322	0.322	0.098	0.131	0.115	0.410	0.423	0.416	0.272	0.282	0.277
	HolE	0.370	0.374	0.372	0.100	0.158	0.129	0.344	0.333	0.339	0.283	0.287	0.285
SawETM	TransE	0.564	0.647	0.626	0.203	0.203	0.203	0.268	0.436	0.352	0.410	0.481	0.445
	DistMult	0.548	0.635	0.591	0.082	0.086	0.084	0.272	0.390	0.331	0.294	0.346	0.319
	ComplEx	0.505	0.621	0.563	0.116	0.116	0.116	0.301	0.459	0.380	0.272	0.320	0.296
	RotatE	0.524	0.621	0.573	0.137	0.143	0.140	0.167	0.299	0.233	0.318	0.380	0.349
	HolE	0.521	0.607	0.564	0.203	0.203	0.203	0.143	0.218	0.180	0.317	0.367	0.342
HyperMiner	TransE	0.628	0.705	0.666	0.097	0.100	0.098	0.128	0.384	0.256	0.367	0.374	0.370
	DistMult	0.650	0.734	0.692	0.028	0.034	0.031	0.128	0.384	0.256	0.335	0.335	0.335
	ComplEx	0.559	0.635	0.597	0.121	0.138	0.130	0.128	0.384	0.256	0.306	0.324	0.315
	RotatE	0.607	0.682	0.644	0.101	0.109	0.105	0.128	0.384	0.256	0.294	0.354	0.324
	HolE	0.580	0.655	0.618	0.082	0.094	0.088	0.128	0.384	0.256	0.402	0.434	0.418

Table 5.2: The Performance Comparison for Various Models over Datasets

WebRED, indicating its inferior effectiveness on these datasets.

The Traco model, employing various embeddings, exhibited varying levels of performance: TransE embedding performed relatively well on FB15k-237 (HTQ = 0.385) but poorly on WikiData (HTQ = 0.014) and WebRED (HTQ = 0.190). Similarly, DistMult, ComplEx, RotatE, and HoIE embeddings displayed comparable trends, showcasing optimal performance on FB15k-237 and a noticeable decline on WikiData.

SawETM demonstrated overall superior performance compared to Traco, except on DBpedia, particularly with TransE and DistMult embeddings. TransE achieved the highest HTQ of 0.626 on FB15k-237 among all embeddings for this model. Performance decreased on WikiData, with the highest HTQ values of 0.203 for TransE and HoIE.

HyperMinor generally exhibited the best performance among models utilizing embeddings on FB15k-237. DistMult embedding attained the highest HTQ on FB15k-237 (0.692) and sustained relatively strong performance across other datasets. TransE, ComplEx, RotatE, and HoIE embeddings also demonstrated robust outcomes, with HTQ values consistently surpassing those observed for Traco and SawETM across all datasets.

In conclusion, the nHDP_KG model outperformed other models across all datasets. HyperMinor consistently achieved high HTQ values across various datasets, particularly with the DistMult embedding. Traco and SawETM models displayed varying performance, with SawETM generally outperforming Traco. The hLDA model exhibited the least effectiveness in this evaluation.

Coverage As depicted in Figure 5.3-5.6, an analysis was conducted to observe the patterns in coverage measurements at different levels of hierarchy for various models across all datasets. The coverage scores are presented within a minimum and maximum values range, with the average coverage score indicated by dot values for all hierarchical levels. It is noteworthy that hLDA exhibits root topics due to its inherent technique characteristics.

The visual representations in Figure 5.3 illustrate the coverage trends for FB15k-237. Results from nHDP_KG, SaWETM, and HyperMiner show a clear decline in "Coverage_-

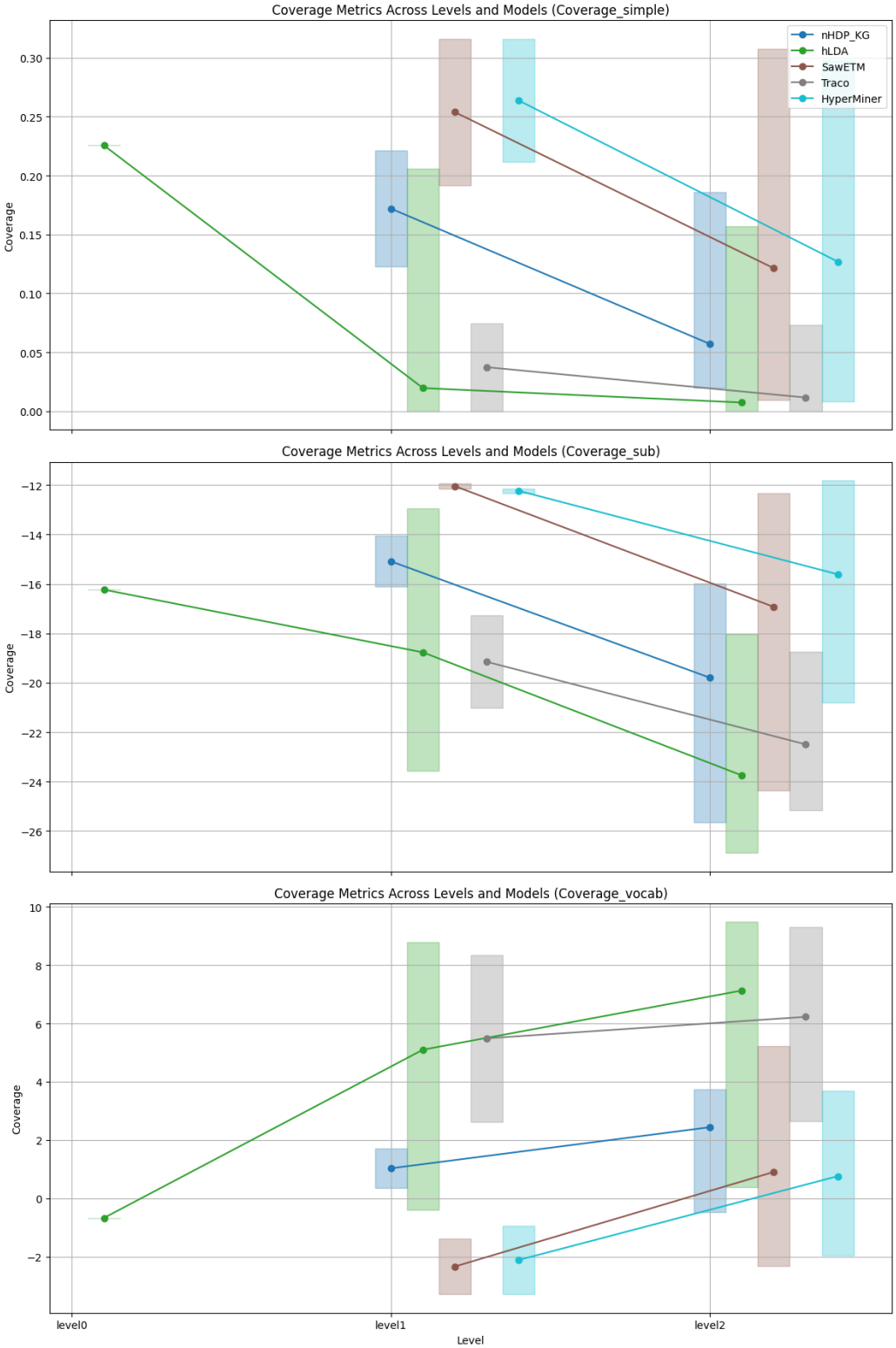


Figure 5.3: Coverage Trend of FB15k-237

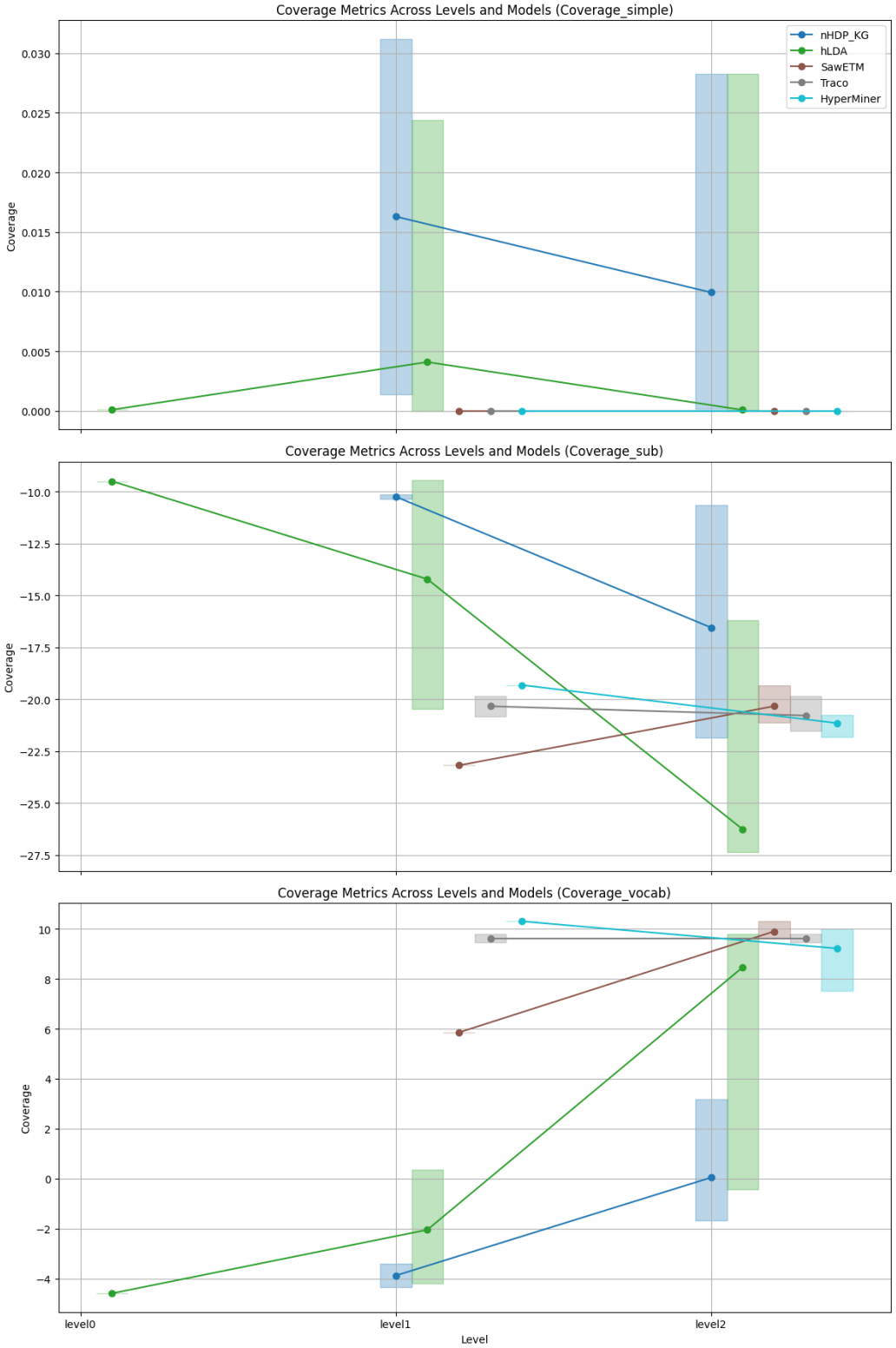


Figure 5.4: Coverage Trend of Wikidata

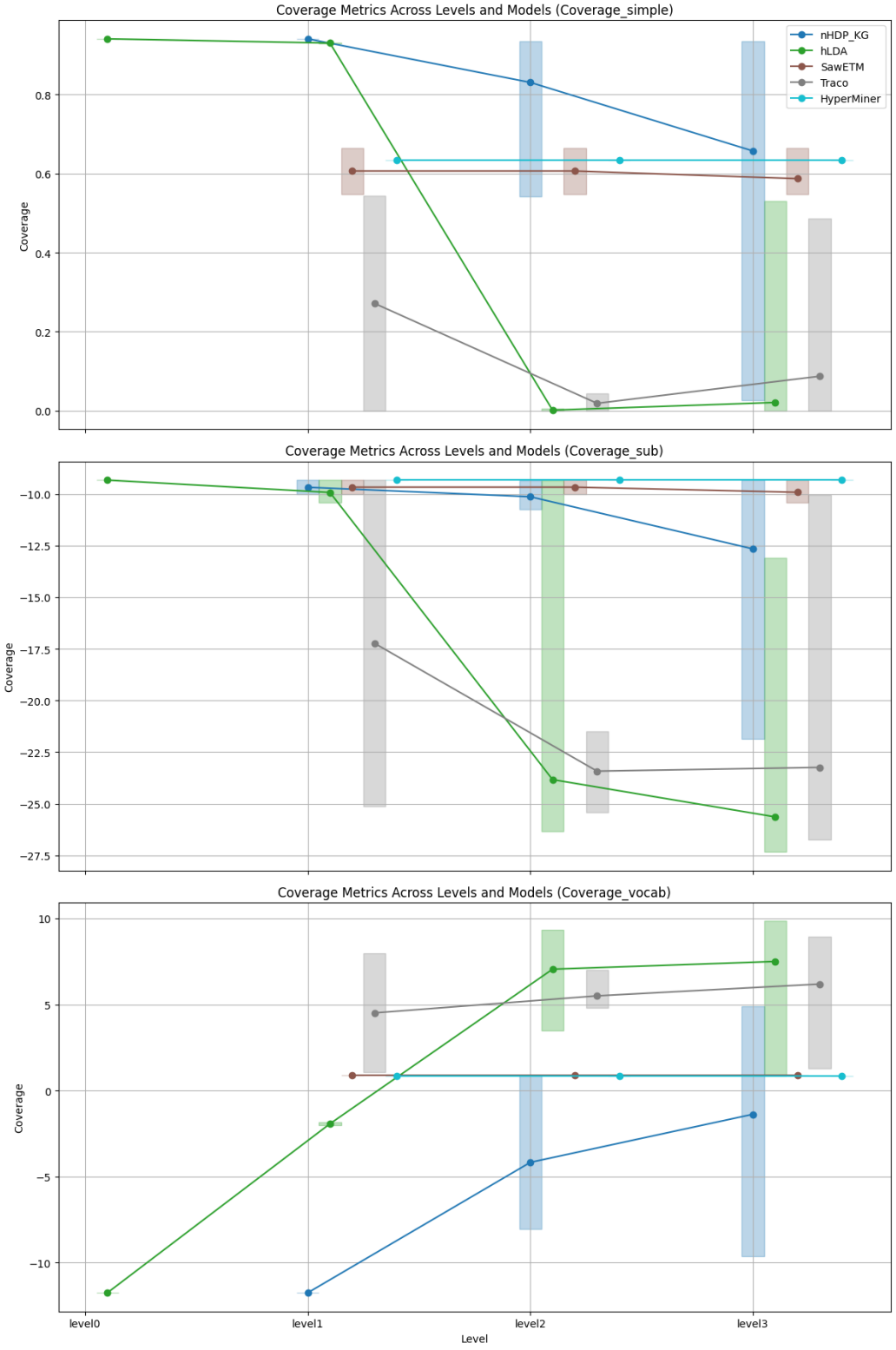


Figure 5.5: Coverage Trend of DBpedia

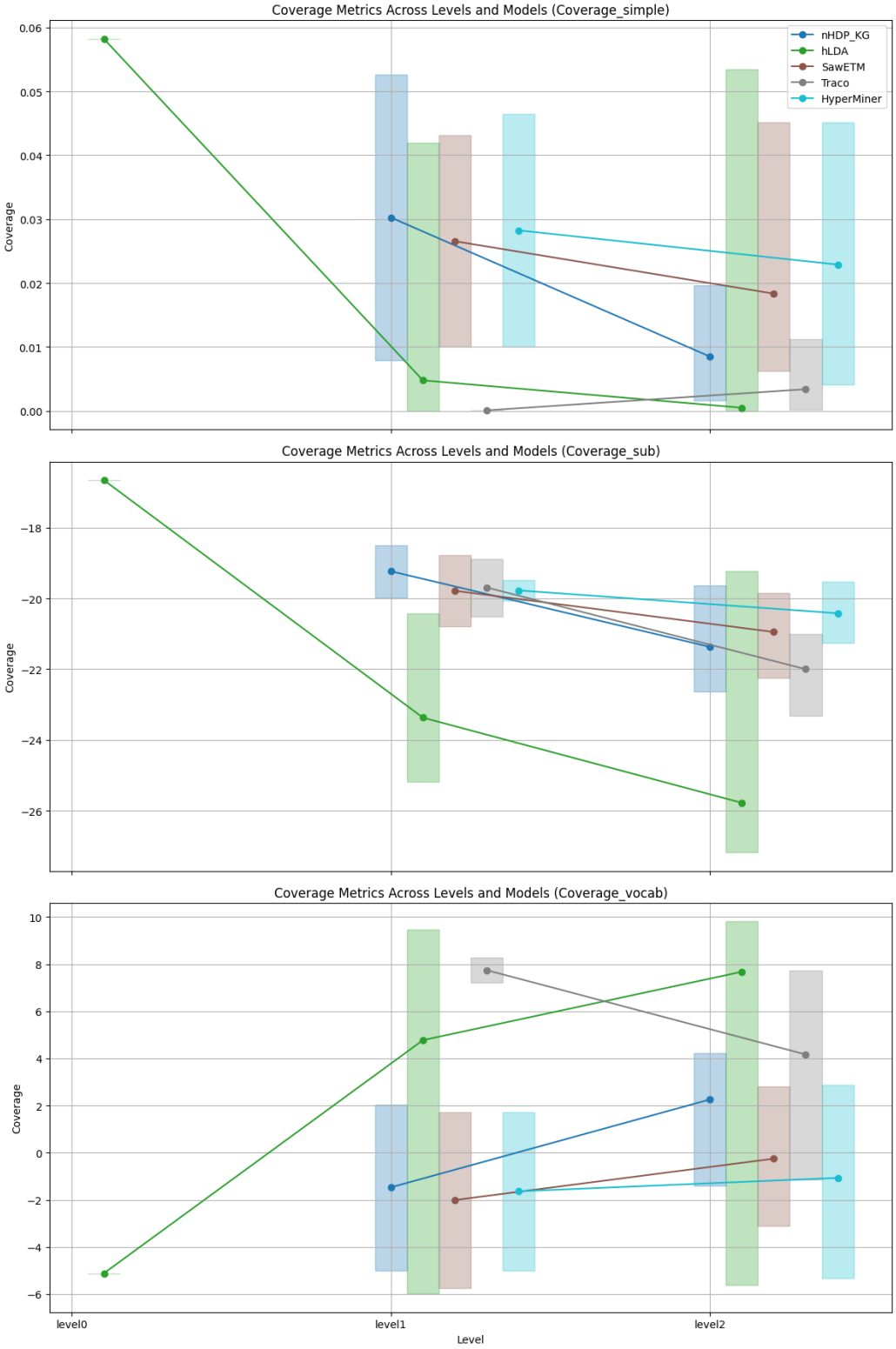


Figure 5.6: Coverage Trend of WebRED

simple" and "Coverage_sub" as coverage levels increase, while "Coverage_vocab" demonstrates an upward trend, consistent with our initial assumptions. In contrast, hLDA and Traco display relatively stable performance with only minor fluctuations across all metrics at levels 1 and 2.

In Figure 5.4, the coverage trends for Wikidata are depicted. The metric "Coverage_simple" shows a decline in both nHDP_KG and hLDA from level1 to level2, while SawETM, Traco, and HyperMiner display consistent levels without a decreasing trend. Regarding the "Coverage_sub" metric, nHDP_KG and hLDA exhibit a continuous decrease across levels, whereas Traco and HyperMiner demonstrate less pronounced but still evident reductions, with SawETM displaying an unexpected increase. Conversely, the "Coverage_vocab" metric indicates an upward trend for nHDP_KG and hLDA, suggesting enhanced vocabulary coverage at higher hierarchical levels. On the other hand, SaWETM and Traco show relatively stable trends with slight improvements, while HyperMiner exhibits variability but ultimately lower coverage at level2.

In Figure 5.5, the graph depicts the trends in coverage for Dbpedia. The metrics "Coverage_simple" and "Coverage_sub" show a clear decreasing trend as levels increase for nHDP_KG and hLDA, whereas SawETM and HyperMiner exhibit a relatively consistent coverage. Traco, on the other hand, does not display any noticeable coverage trend in this context. Conversely, nHDP_KG and hLDA demonstrate an increasing trend in the "Coverage_vocab" metric, suggesting enhanced vocabulary coverage at higher levels. Additionally, Traco shows a slight increase across all levels, while SawETM and HyperMiner maintain a more stable performance.

Figure 5.6 illustrates the coverage trends for WebRED. The nHDP_KG model shows a sharp decline in the "Coverage_simple" metric as levels increase, while other models demonstrate more stable or slightly decreasing trends. However, Traco does not efficiently capture the hierarchy. In the "Coverage_sub" metric, nHDP_KG exhibits a consistent decrease across levels, while the other models display moderate declines. On the other

hand, in the "Coverage_vocab" metric, nHDP_KG shows an increasing trend, indicating enhanced vocabulary coverage at higher levels. In contrast, other models show more stability or slight increases in this metric, with Traco showing a decline.

In summary, nHDP_KG and hLDA successfully generate hierarchical trees that align with our coverage assumptions, demonstrating their ability to capture the intended topic structure. In contrast, other models appear to be weaker in this regard.

5.6.2 Qualitative Evaluation

Table 5.3: Top 5 Words Distribution of Topics for Gene Wilder

Entities		Subtree for entity: (sub_id: 2582) /m/027l0b (Gene Wilder, Wikipedia) With multi label Entertainer, communicator
Subtree	(1,)	
	(1, 1)	/people/person/profession /award/award_nominee/award_nominations./award/award_nomination/award /film/actor/film./film/performance/film /award/award_nominee/award_nominations./award/award_nomination/nominated_for /people/person/nationality
	(1, 1, 1)	/award/award_nominee/award_nominations./award/award_nomination/award /people/person/profession /award/award_nominee/award_nominations./award/award_nomination/award_nominee /music/artist/track_contributions./music/track_contribution/role /award/award_winner/awards_won./award/award_honor/award_winner
	(1, 1, 3)	/award/award_nominee/award_nominations./award/award_nomination/award_nominee /award/award_winner/awards_won./award/award_honor/award_winner /film/actor/film./film/performance/film /award/award_nominee/award_nominations./award/award_nomination/award /award/award_nominee/award_nominations./award/award_nomination/nominated_for
	(1, 2)	/music/record_label/artist /m/09nqf(United States dollar) /people/ethnicity/people /olympics/olympic_participating_country/medals_won./olympics/olympic_medal_honor/olympics /media_common/netflix_genre/titles
	(1, 2, 1)	/location/location/contains /location/location/adjoin_s./location/adjoining_relationship/adjoins /location/location/time_zones /m/0jbk9(United States Department of Housing and Urban Development) /location/hud_foreclosure_area/estimated_number_of_mortgages./measurement_unit/dated_integer/source

Table 5.4: Top 5 Words Distribution for Topics for Gene Wilder (continued)

Entities		Subtree for entity: (sub_id: 2582) /m/027l0b (Gene Wilder, Wikipedia) With multi label Entertainer, communicator
	(1, 2,.4)	/award/award_category/winners./award/award_honor/award_winner /award/award_category/winners./award/award_honor/ceremony /award/award_category/nominees./award/award_nomination/nominated_for /government/government_office_category/officeholders./government/government_position_held/jurisdiction_of_office /business/job_title/people_with_this_title./business/employment_tenure/company
Original_Triples		Triples for the subject **/m/027l0b – Gene Wilder** '/film/actor/film./film/performance/film', '/m/085bd1', '/common/topic/webpage./common/webpage/category', '/m/08mbj5d', '/people/person/place_of_birth', '/m/0dyl9', '/film/actor/film./film/performance/film', '/m/0hvvf', '/award/award_nominee/award_nominations./award/award_nomination/award', '/m/09qvc0', '/award/award_nominee/award_nominations./award/award_nomination/award', '/m/0gqy2', '/award/award_nominee/award_nominations./award/award_nomination/nominated_for', '/m/0291ck', '/award/award_winner/awards_won./award/award_honor/award_winner', '/m/052hl', '/award/award_nominee/award_nominations./award/award_nomination/nominated_for', '/m/017kz7', '/people/person/profession', '/m/0dxtg', '/people/person/religion', '/m/03_gx', '/people/person/spouse_s./people/marriage/type_of_union', '/m/04ztj', '/people/person/profession', '/m/02jknp', '/people/person/profession', '/m/018gz8', '/people/person/religion', '/m/0kpl', '/people/person/profession', '/m/02hrh1q', '/people/person/profession', '/m/0kyk', '/film/actor/film./film/performance/film', '/m/017kz7', '/film/actor/film./film/performance/film', '/m/0291ck', '/people/person/profession', '/m/0xzm', '/film/actor/film./film/performance/film', '/m/018f8', '/award/award_nominee/award_nominations./award/award_nomination/nominated_for', '/m/01q_y0', '/people/person/profession', '/m/0cbd2']

The acquired hierarchy tree is presented herein, derived from predicates and objects employed to characterize subject entities. The final goal is to analyze the distribution pattern of these terminologies within the context of subject-based hierarchical clustering.

Here, we present the results for the top 5 word distributions for topics for *Gene Wilder*. Table 5.3 presents a meticulous hierarchical representation of Gene Wilder's career and achievements, identified by subject ID 2582. The subtrees describe Wilder's roles in films, nominations, awards, musical contributions, and other aspects of his career.

The hierarchical notation, e.g., (1), (1, 1), (1, 1, 1), demonstrates how each category is further subdivided to present a comprehensive view of the obtained tree. There is, for example, a node (1, 1, 1) that explains the specifics of award nominations and wins related to Wilder's film performances. The structured approach facilitates a deeper understanding of Wilder's career, highlighting the breadth and depth of his contributions. Each level represents a different level of detail throughout the subtrees, from broad categories such as profession or award to finer details such as specific award nominations or film performances. For instance, a triple indicating a film performance ⟨'/film/actor/film/film/performance/film', '/m/0bsb1d'⟩ would be categorized under the film performance subtree, while an award nomination triple ⟨'/award/award_nominee/award_nominations/award/award_nomination/nominated_for', '/m/09qwc0'⟩ falls under the awards subtree. This hierarchical structuring of triples into subtrees facilitates a comprehensive and nuanced analysis of Gene Wilder's career, illustrating the interconnectedness of his professional achievements in a detailed manner.

Table 5.5 provides a hierarchical representation of the entity *Centre College*, identified by subject ID 4815, encompassing its multifaceted attributes and relationships. The entity is associated with labels such as *organization* and *institution.* The hierarchical structure, indicated by tuple notations (e.g., (1), (1, 1), (1, 2)), organizes various attributes into subtrees for detailed analysis. For example, the subtree under (1, 1) highlights categories related to professions, awards, and film performances, while (1, 2) includes financial and demographic attributes such as currency and ethnicity. The subtree (1, 2, 1) expands on geographical and administrative details, such as locations and time zones. Compared with 'Original_Triples,' it provides specific details on educational aspects, financial data, and institutional characteristics. This hierarchical organization allows for a comprehensive and nuanced analysis of *Centre College*, illustrating the interconnectedness of its various attributes systematically.

The distribution of subtree sizes over all subjects is presented in Figure 5.7. Most of the

Table 5.5: Top 5 words distribution for topics for Centre College

Entities		Subtree for entity: (sub_id: 4815) /m/04344j (Centre College, https://www.wikidata.org/wiki/Q1804942) With multi label organization, instituion
Subtree	(1,)	
	(1, 1)	/people/person/profession /award/award_nominee/award_nominations./award/award_nomination/award /film/actor/film./film/performance/film /award/award_nominee/award_nominations./award/award_nomination/nominated_for /people/person/nationality
	(1, 2)	/music/record_label/artist /m/09nqf (United States dollar) /people/ethnicity/people /olympics/olympic_participating_country/medals_won./olympics/olympic_medal_honor/olympics /media_common/netflix_genre/titles
	(1,2,1)	/location/location/contains /location/location/adjoin_s./location/adjoining_relationship/adjoins /location/location/time_zones /m/0jbk9 (United States Department of Housing and Urban Development) /location/hud_foreclosure_area/estimated_number_of_mortgages./measurement_unit/dated_integer/source
	(1,2,3)	/education/educational_institution/students_graduates./education/education/major_field_of_study /music/performance_role/track_performances./music/track_contribution/role /education/educational_institution/students_graduates./education/education/student /award/award_ceremony/awards_presented./award/award_honor/award_winner /music/performance_role/regular_performances./music/group_membership/role
Original_Triples		[Triples for the subject **/m/04344j – Centre College** '/education/educational_institution/school_type', '/m/04qbv', '/common/topic/webpage./common/webpage/category', '/m/08mbj5d', '/education/university/domestic_tuition./measurement_unit/dated_money_value/currency', '/m/09nqf', '/organization/endowed_organization/endowment./measurement_unit/dated_money_value/currency', '/m/09nqf', '/education/university/fraternities_and_sororities', '/m/035tlh', '/education/educational_institution/students_graduates./education/education/major_field_of_study', '/m/062z7', '/education/educational_institution/school_type', '/m/01rs41', '/education/university/local_tuition./measurement_unit/dated_money_value/currency', '/m/09nqf', '/education/educational_institution/colors', '/m/011l849', '/education/university/fraternities_and_sororities', '/m/04m8fy', '/education/educational_institution_campus/educational_institution', '/m/04344j', '/education/university/fraternities_and_sororities', '/m/0325pb']

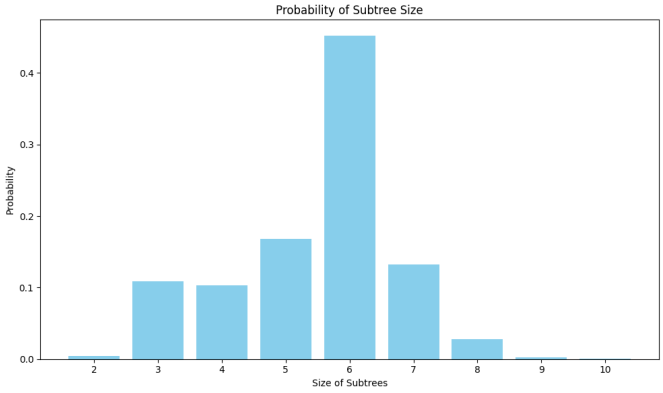


Figure 5.7: Overall subtree size distribution

subjects have 6 node subtrees.

5.7 Conclusion

In conclusion, our study effectively applies the nested hierarchical Dirichlet process (nHDP) to analyze knowledge graphs, revealing subject-specific distributions and representing global knowledge graphs as local subtrees. Through quantitative evaluation of multiple models on various datasets, our $nHDP\_KG$ model outperforms existing neural-network-based hierarchical clustering techniques, indicating its potential to advance the organization of large-scale knowledge bases. Additionally, qualitative assessment showcases the model's ability to generate meaningful subtrees, providing insights into the structure and relationships within knowledge graphs and demonstrating the robustness and versatility of our approach.

Chapter 6

Conclusions, Recommendations, & Future Work

The growing interest in knowledge graphs and their use in Retrieval-Augmented Generation (RAG) systems has led to the need for automated methods for building and analyzing graphs. This research focuses on creating graphs with specific vocabulary using advanced techniques. A substantial portion of the work is dedicated to analyzing and processing the graphs. The developed approaches transform the flat structure of a knowledge graph into a hierarchical organization, revealing data patterns at various levels of abstraction.

This thesis reports the findings of three projects discussed in the previous chapters. Here we provide a thorough overview of the contributions, conclusions, and potential directions for future research.

6.1 Contributions

The dissertation addressed the process of building knowledge graphs and further, to a higher degree, the methods for hierarchical analysis of graphs. We investigated various large language models for constructing knowledge graphs and proposed new data augmentation to improve the triple extraction process. We performed the analysis of knowledge graphs by adapting conventional hierarchical topic models and introducing a few innovative approaches for learning hierarchy. We evaluated their effectiveness on real-world data.

- **Chapter 3** explores the fine-tuning of large language models (LLMs) for triple extraction using data augmentation techniques and comparing their performance to the GPT family of LLMs.

  We implemented a pipeline composed of processes for data augmentation and preparation and model training using diverse trainers from HuggingFace. We applied it to the WebNLG dataset and benchmarked against ChatGPT and GPT-4. Our analysis extended to real-world datasets such as SKE, DocRed, FewRel, and KELM. We demonstrated that fine-tuned models with seven billion parameters outperformed GPT-4, particularly on WebNLG.

  Key findings highlight the effectiveness of data augmentation, the superior performance of smaller fine-tuned models, and the critical role of high-quality training data. Limitations include occasional hallucinations and looping issues. This work underscores that effective data augmentation and fine-tuning can enable smaller LLMs to match or exceed the performance of larger models like GPT-4 in specific tasks.

- In **Chapter 4**, we introduced a hierarchical topic modeling approach tailored for knowledge graphs, inspired by probabilistic models like Latent Dirichlet Allocation (LDA) and its hierarchical extension (hLDA). Our method aimed to uncover latent structures within knowledge graphs by organizing entities and predicates into a tree of abstract topics.

  Key components of the proposed technique include 1) data preprocessing, where <predicates> and <predicates, objects> are treated as tags for subjects; 2) a generative model that forms a hierarchical topic tree; and 3) inference using a non-parametric prior and Gibbs sampling for efficient posterior inference.

  We evaluated the model on datasets FB15k-237, YAGO3-10, and DBpedia, demonstrating its effectiveness in clustering tasks and the coherence of the induced topic hierarchies.

Key findings show that the model organizes knowledge graph elements into meaningful clusters without prior tree structure assumptions, performs competitively in quantitative evaluations, and produces interpretable topic hierarchies. However, limitations include the model's singular path representation for subjects with diverse topics, redundancy in the tree leading to overlapping subjects, and high computational resource consumption during inference.

- **Chapter 5**: We adopted a non-parametric probabilistic model, the nested hierarchical Dirichlet process (DP), to process knowledge graphs, aiming to discover latent subject-specific distributions along paths within a hierarchical tree. The proposed approach allowed us to represent global trees as collections of local subtrees for each subject, revealing cross-thematic topics. Key elements of the approach are data preprocessing, treating predicates and objects as words describing subjects, a generative model enabling subjects to learn distributions on coherent topics, and inference using stochastic variational methods to estimate the posterior inference of GDP.

  Evaluations were conducted on datasets FB15k-237, DBpedia, Wikidata, and WebRED, assessing hierarchy topic quality, simple coverage, subject-based coverage, and vocabulary-based coverage.

  Key findings indicate that nHDP_KG effectively organizes subjects into meaningful hierarchical trees, demonstrating superior topic coherence and coverage performance compared to other models such as Traco, SawETM, and hLDA. Qualitative assessments further highlight the coherence and interpretability of the induced subtrees, showcasing the model's ability to generate insightful hierarchical structures within knowledge graphs.

Ultimately, it is anticipated that the researchers in knowledge representation will adopt and build upon the presented research contributions. It is worth noting that the learning hierarchies from knowledge graphs still need to be researched. No single method cur-

rently exists that can handle all the tasks mentioned, and many hierarchies produced by current methods, whether discussed in this dissertation or not, have room for improvement. Consequently, we have not achieved an optimal automated learning of hierarchies from knowledge graphs. We hope that future researchers advance this often-overlooked field. All this leads us to the final point of discussion in this dissertation, which is the potential avenues for future research.

6.2 Future Work

The implications of the findings presented in this thesis are significant for both research and practical applications for constructing and analyzing knowledge graphs. The adaptability and performance exhibited in constructing knowledge graphs and analyzing hierarchies across different datasets emphasize the potential enhancements for knowledge updates and information retrieval. Moreover, there is a desire to further investigate the integration of knowledge graphs with large language models to retrieve more precise and relevant information. The causal knowledge graph construction and downstream tasks could benefit from utilizing hierarchical structures learned from models utilizing conditional knowledge graphs and knowledge graph reasoning.

- Integrating large language models with knowledge graphs for enhanced retrieval-augmented generation presents a compelling research opportunity with significant implications. This strategy can significantly enhance information retrieval by utilizing the organized data in knowledge graphs, resulting in more precise and contextually relevant outputs. Moreover, this fusion improves natural language comprehension, enabling models to produce more cohesive and informative responses by integrating external knowledge sources. The broad applicability of this research across different domains such as healthcare, finance, and e-commerce indicates promising advancements in information retrieval and generation tasks.

- Constructing causal knowledge graphs and hierarchies of events/facts they represent is very important in diverse fields because it can reveal the fundamental causes of intricate issues, thus enabling precise interventions. These graphs play a crucial role in healthcare, finance, and manufacturing decision-making by examining causal connections to guide decision-making. Moreover, specialized causal knowledge graphs in specific domains improve reasoning and inference mechanisms by capturing complex causal links, enhancing comprehension of intricate systems, and boosting analytical abilities. The hierarchy information we learned from hierarchical topic modeling could help construct the conditional KG with the latent hierarchical patterns.

- The hierarchy structure learned from KGs is essential for enhancing knowledge graph reasoning by organizing entities and their relationships in a structured manner. This hierarchical organization allows for efficient reasoning processes through faster traversal and inference, facilitates semantic understanding by capturing hierarchical dependencies, and supports contextual knowledge representation. Additionally, hierarchical reasoning enables analysis at different levels of abstraction, uncovering complex relationships that a flat graph might miss. The hierarchy structure also promotes scalability and generalization, allowing the KG to handle larger and more complex datasets while generalizing knowledge across various abstraction levels. Consequently, this structured representation significantly enhances the KG's ability to capture and reason over intricate relationships and dependencies, leading to more accurate and insightful reasoning outcomes.

In summary, this thesis's contributions are twofold. It adds to the expanding literature on large language models by proposing a promising framework for knowledge graph construction using prompt engineering. It also shows that non-parametric techniques with probabilistic topic modeling are a promising approach to analyzing and extracting information at different levels of abstraction from graphs. Hopefully, this work will pave the

way for more effective, accessible, and personalized interventions in the knowledge graph domain.

Bibliography

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2021.

[2] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008, pp. 1247–1250.

[4] F. Mahdisoltani, J. Biega, and F. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *7th Biennial Conference on Innovative Data Systems Research*, CIDR Conference, 2014.

[5] J. Lehmann *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic web*, vol. 6, no. 2, pp. 167–195, 2015.

[6] X. Zou, "A survey on application of knowledge graph," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1487, 2020, p. 012 016.

[7] Q. Guo *et al.*, "A survey on knowledge graph-based recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3549–3568, 2020.

[8] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, ser. WSDM '19, Association for Computing Machinery, 2019, 105–113, ISBN: 9781450359405. DOI: 10.1145/3289600.3290956. [Online]. Available: https://doi.org/10.1145/3289600.3290956.

[9] J. Yang *et al.*, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," *ArXiv Preprint arXiv:2304.13712*, 2023.

[10] M. Pietrasik and M. Reformat, "Path based hierarchical clustering on knowledge graphs," *ArXiv Preprint arXiv:2109.13178*, 2021.

[11] E. J. Miller, "An introduction to the resource description framework," *Journal of Library Administration*, vol. 34, no. 3-4, pp. 245–255, 2001.

[12] D. Beckett and B. McBride, "Rdf/xml syntax specification (revised)," *W3C Recommendation*, vol. 10, no. 2.3, 2004.

[13] N. Trokanas, F. Cecelja, and T. Raafat, "Towards a re-usable ontology for waste processing," in *Computer Aided Chemical Engineering*, vol. 33, Elsevier, 2014, pp. 841–846.

[14] V Sugumaran, "Semantic technologies for enhancing knowledge management systems," in *Successes and Failures of Knowledge Management*, Elsevier, 2016, pp. 203–213.

[15] N. Trokanas, T. Raafat, F. Cecelja, A. Kokossis, and A. Yang, "Semantic formalism for waste and processing technology classifications using ontology models," in *Computer Aided Chemical Engineering*, vol. 30, Elsevier, 2012, pp. 167–171.

[16] A. Patel and N. C. Debnath, "A comprehensive overview of ontology: Fundamental and research directions," *Current Materials Science: Formerly: Recent Patents on Materials Science*, vol. 17, no. 1, pp. 2–20, 2024.

[17] L. Gillam, M. Tariq, and K. Ahmad, "Terminology and the construction of ontology," *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, vol. 11, no. 1, pp. 55–81, 2005.

[18] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," *Electronics*, vol. 9, no. 5, p. 750, 2020.

[19] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2015.

[20] A. García-Durán, A. Bordes, and N. Usunier, "Effective blending of two and three-way interactions for modeling multi-relational data," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, Springer, 2014, pp. 434–449.

[21] C. Fourrier, N. Habib, A. Lozovskaya, K. Szafer, and T. Wolf, *Open llm leaderboard v2*, https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

[22] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[23] T. Xie *et al.*, "Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models," *ArXiv Preprint arXiv:2201.05966*, 2022.

[24] S. Minaee *et al.*, "Large language models: A survey," *ArXiv Preprint arXiv:2402.06196*, 2024.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv Preprint arXiv:1810.04805*, 2018.

[26] J. Achiam *et al.*, "Gpt-4 technical report," *ArXiv Preprint arXiv:2303.08774*, 2023.

[27] H. Touvron *et al.*, "Llama: Open and efficient foundation language models," *ArXiv*, vol. abs/2302.13971, 2023.

[28] A. Chowdhery *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[29] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[30] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[31] M. Hofer, D. Obraczka, A. Saeedi, H. Köpcke, and E. Rahm, "Construction of knowledge graphs: State and challenges," *ArXiv Preprint arXiv:2302.11509*, 2023.

[32] T. Castro Ferreira *et al.*, "The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020)," in *3rd International Workshop on Natural Language Generation from the Semantic Web*, T. Castro Ferreira *et al.*, Eds., Dublin, Ireland (Virtual): Association for Computational Linguistics, Dec. 2020, pp. 55–76. [Online]. Available: https://aclanthology.org/2020.webnlg-1.7.

[33] Y. Liu, T. Zhang, Z. Liang, H. Ji, and D. L. McGuinness, *Seq2rdf: An end-to-end application for deriving triples from natural language text*, 2018. ArXiv: 1807.01763 (cs.CL).

[34] X. Wei *et al.*, *Zero-shot information extraction via chatting with chatgpt*, 2023. ArXiv: 2302.10205 (cs.CL).

[35] K. Sun, Y. E. Xu, H. Zha, Y. Liu, and X. L. Dong, *Head-to-tail: How knowledgeable are large language models (llm)? a.k.a. will llms replace knowledge graphs?* 2023. ArXiv: 2308.10168 (cs.CL).

[36] S. Wadhwa, S. Amir, and B. C. Wallace, *Revisiting relation extraction in the era of large language models*, 2023. ArXiv: 2305.05003 (cs.CL).

[37] X. Han *et al.*, "Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation," *ArXiv Preprint arXiv:1810.10147*, 2018.

[38] Y. Yao *et al.*, "Docred: A large-scale document-level relation extraction dataset," *ArXiv Preprint arXiv:1906.06127*, 2019.

[39] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou, "Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training," *ArXiv Preprint arXiv:2010.12688*, 2020.

[40] C. Xie, J. Liang, J. Liu, C. Huang, W. Huang, and Y. Xiao, "Revisiting the negative data of distantly supervised relation extraction," *ArXiv Preprint arXiv:2105.10158*, 2021.

[41] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[42] W.-L. Chiang *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, vol. 2, no. 3, p. 6, 2023.

[43] C. Xu *et al.*, "Wizardlm: Empowering large language models to follow complex instructions," *ArXiv Preprint arXiv:2304.12244*, 2023.

[44] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, and A. Awadallah, *Orca: Progressive learning from complex explanation traces of gpt-4*, 2023. ArXiv: 2306.02707 (cs.CL).

[45] W. Lian *et al.*, *Llongorca7b: Llama2-7b model instruct-tuned for long context on filtered openorcav1 gpt-4 dataset*, https://https://huggingface.co/Open-Orca/LlongOrca-7B-16k, 2023.

[46] D. Kim *et al.*, "Solar 10.7 b: Scaling large language models with simple yet effective depth up-scaling," *ArXiv Preprint arXiv:2312.15166*, 2023.

[47] A. N. Lee, C. J. Hunter, and N. Ruiz, "Platypus: Quick, cheap, and powerful refinement of llms," *ArXiv Preprint arXiv:2308.07317*, 2023.

[48] H. Liu *et al.*, "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1950–1965, 2022.

[49] E. J. Hu *et al.*, "Lora: Low-rank adaptation of large language models," *CoRR*, 2021.

[50] L. von Werra *et al.*, *Trl: Transformer reinforcement learning*, https://github.com/huggingface/trl, 2020.

[51] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo, "SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013)," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, S. Manandhar and D. Yuret, Eds., Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 341–350. [Online]. Available: https://aclanthology.org/S13-2056.

[52] A. Bordes, N. Usunier, S. Chopra, and J. Weston, "Large-scale simple question answering with memory networks," *ArXiv Preprint arXiv:1506.02075*, 2015.

[53] R. Das *et al.*, "Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning," *ArXiv Preprint arXiv:1711.05851*, 2017.

[54] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*, Springer, 2018, pp. 593–607.

[55] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[56] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Thirty-second AAAI Conference on Artificial Intelligence*, 2018.

[57] V. Bellini, A. Schiavone, T. Di Noia, A. Ragone, and E. Di Sciascio, "Knowledge-aware autoencoders for explainable recommender systems," in *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*, 2018.

[58] D. M. Blei, T. L. Griffiths, and M. I. Jordan, "The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies," *Journal of the ACM (JACM)*, vol. 57, no. 2, p. 7, 2010.

[59] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[60] D. J. Aldous, "Exchangeability and related topics," in *École d'Été de Probabilités de Saint-Flour XIII—1983*, Springer, 1985, pp. 1–198.

[61] J. Pitman, *Combinatorial stochastic processes: Ecole d'eté de probabilités de saint-flour xxxii-2002*. Springer, 2006.

[62] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 2015, pp. 57–66.

[63] N. Jain, J.-C. Kalo, W.-T. Balke, and R. Krestel, "Do embeddings actually capture knowledge graph semantics?" In *European Semantic Web Conference*, Springer, 2021, pp. 143–159.

[64] G. A. Miller, *WordNet: An electronic lexical database*. MIT Press, 1998.

[65] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[66] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[67] S. Broscheit, D. Ruffinelli, A. Kochsiek, P. Betz, and R. Gemulla, "Libkge-a knowledge graph embedding library for reproducible research," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 165–174.

[68] M. Pietrasik and M. Reformat, "A simple method for inducing class taxonomies in knowledge graphs," in *European Semantic Web Conference*, Springer, 2020, pp. 53–68.

[69] W. Croft, "On two mathematical representations for "semantic maps"," *Zeitschrift für Sprachwissenschaft*, vol. 41, no. 1, pp. 67–87, 2022.

[70] J. Jalving, S. Shin, and V. M. Zavala, "A graph-based modeling abstraction for optimization: Concepts and implementation in plasmo. jl," *Mathematical Programming Computation*, vol. 14, no. 4, pp. 699–747, 2022.

[71] Y. Zhang, M. Pietrasik, W. Xu, and M. Reformat, "Hierarchical topic modelling for knowledge graphs," in *European Semantic Web Conference*, Springer, 2022, pp. 270–286.

[72] M. Li, Y. Wang, D. Zhang, Y. Jia, and X. Cheng, "Link prediction in knowledge graphs: A hierarchy-constrained approach," *IEEE Transactions on Big Data*, vol. 8, no. 3, pp. 630–643, 2018.

[73] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 3065–3072.

[74] J. Dong, Q. Zhang, X. Huang, K. Duan, Q. Tan, and Z. Jiang, "Hierarchy-aware multi-hop question answering over knowledge graphs," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2519–2527.

[75] T. Griffiths, M. Jordan, J. Tenenbaum, and D. Blei, "Hierarchical topic models and the nested chinese restaurant process," *Advances in Neural Information Processing Systems*, vol. 16, 2003.

[76] J. H. Kim, D. Kim, S. Kim, and A. Oh, "Modeling topic hierarchies with the recursive chinese restaurant process," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 783–792.

[77] X. Wu *et al.*, "On the affinity, rationality, and diversity of hierarchical topic modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. [Online]. Available: https://arxiv.org/pdf/2401.14113.pdf.

[78] Z. Duan *et al.*, "Sawtooth factorial topic embeddings guided gamma belief network," in *International Conference on Machine Learning*, PMLR, 2021, pp. 2903–2913.

[79] Y. Xu, D. Wang, B. Chen, R. Lu, Z. Duan, M. Zhou, *et al.*, "Hyperminer: Topic taxonomy mining with hyperbolic embedding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31 557–31 570, 2022.

[80] R. Fu, J. Guo, B. Qin, W. Che, H. Wang, and T. Liu, "Learning semantic hierarchies: A continuous vector space approach," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 461–471, 2015.

[81] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *ArXiv Preprint arXiv:1412.6575*, 2014.

[82] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International Conference on Machine Learning*, PMLR, 2016, pp. 2071–2080.

[83] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *ArXiv Preprint arXiv:1902.10197*, 2019.

[84] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 30, 2016.

[85] T. S. Ferguson, "A bayesian analysis of some nonparametric problems," *The Annals of Statistics*, pp. 209–230, 1973.

[86] J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan, "Nested hierarchical dirichlet processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 256–270, 2014.

[87] M. J. B. Yee Whye Teh Michael I Jordan and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006. DOI: 10.1198/016214506000000302. [Online]. Available: https://doi.org/10.1198/016214506000000302.

[88] J. Sethuraman, "A constructive definition of dirichlet priors," *Statistica sinica*, pp. 639–650, 1994.

[89] X. Wang *et al.*, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176–194, 2021.

[90] R. Ormandi, M. Saleh, E. Winter, and V. Rao, *Webred: Effective pretraining and finetuning for relation extraction on the web*, 2021. ArXiv: 2102.09681 (cs.CL). [Online]. Available: https://arxiv.org/abs/2102.09681.

[91] M. K. N. C. J. Marius and K. S. Burkhardt, "Hierarchical topic evaluation: Statistical vs. neural models," *Bayesian Deep Learning Workshop, NeurIPS*, 2021.

[92] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the eighth ACM International Conference on Web Search and Data Mining*, 2015, pp. 399–408.

[93] A. M. Almars, I. A. Ibrahim, X. Zhao, and S. Al-Maskari, "Evaluation methods of hierarchical models," in *Advanced Data Mining and Applications: 14th International Conference, ADMA 2018, Nanjing, China, November 16–18, 2018, Proceedings 14*, Springer, 2018, pp. 455–464.