Extended: Neuro-Inspired Large-Scale Knowledge Graph Construction for Wikipedia

Mustafa Abdullah Hakkoz
Researcher,
Istanbul Technical University

Serkan Macit
Researcher,
Istanbul Technical University

Gürkan Soykan
Researcher,
Wageningen University

Abstract

In this research, we propose a novel, neuro-inspired, unsupervised framework for constructing large-scale knowledge graphs (KGs) from Wikipedia. Unlike current expensive LLM-based approaches, our method builds a scalable, dynamic KG using entropy-based clustering, tree-like reasoning structures, and a specialized multi-granular embedding model. The foundation of our system is a custom-trained text embedding architecture that incorporates hierarchical Matryoshka representations, entropy-optimized contrastive learning, and Kantian Transcendental Category grounding. These embeddings provide the structured input for an unsupervised KG builder that simulates neural processes such as synaptic connection formation and pruning. The resulting KG will be interpretable, computationally efficient, and capable of evolving with Wikipedia's ever-changing content. We believe this research will significantly contribute to knowledge organization within Wikimedia and enable new tools for reasoning, querying, and open data enrichment.

Introduction

The knowledge graphs (KG) are organized ways to show information. Such kind of graph-based Representations make it possible to store and get back data with complex relationships and hierarchies. Knowledge graphs are excellent for providing context, making entities clear, and helping with reasoning tasks, especially for AI-based applications and chatbots.

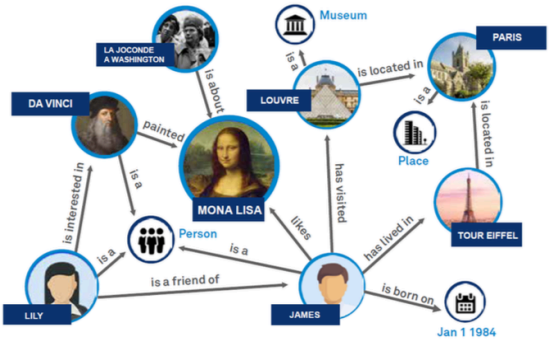

Fig. 1: *Knowledge Graph representation of entities of several types, like individuals, places, and dates.*

The problem this project addresses is the lack of cost-effective, scalable methods for constructing and maintaining **large knowledge graphs** from Wikipedia. The majority of KG constructors currently in use are LLM-based, and despite their strength, they are unaffordable when used on millions of articles. Moreover, they often lack transparency in how knowledge is structured and inferred, limiting their use in contexts where auditability and explainability are essential, such as Wikimedia's ecosystem.

This gap is critical. Wikipedia holds vast unstructured knowledge, but Wikidata only represents a fraction of this information. Bridging this gap through scalable KG construction can

- Enable intelligent assistants and bots to access structured knowledge.
- Support advanced semantic search across Wikipedia.
- Automatically enrich Wikidata with verified facts.
- Provide a foundation for open-domain reasoning engines.

Our research question is: How can we build a computationally efficient, interpretable, and dynamically evolving knowledge graph of Wikipedia using unsupervised, neuro-inspired learning and customized embedding strategies?

Date: We propose a full 24-month timeline (starting at July 1, 2025, and ending at June 30, 2027) to comprehensively address the research goals and ensure high-quality, scalable outcomes. The project is divided into two major phases:

Year 1 (July 2025 – June 2026): *Specialized Embedding Model Development.*
During this phase, we will focus on designing and training a BERT-based text embedding model tailored specifically for Wikipedia and knowledge graph construction. The process includes integrating granular knowledge representations, transcendental category context, entropy-based fine-tuning, and contrastive learning. Extensive evaluation and tuning will be carried out using established semantic benchmarks.

Year 2 (July 2026 – June 2027): *Neuro-Inspired Knowledge Graph Construction.*
In the second year, the trained embeddings will be used as input to a fully unsupervised, tree-based and entropy-driven graph construction engine. We will implement and optimize our neuro-inspired reasoning framework, build and evaluate the large-scale KG, develop visualization and querying tools, and prepare the public API and open-source toolkit for community adoption.

This extended timeline ensures each research component receives focused attention and that the resulting system is stable, reproducible, and impactful for the Wikimedia ecosystem.

Related work

Constructing **large-scale knowledge graphs** from unstructured textual sources, such as Wikipedia, is a critical step toward enabling advanced knowledge retrieval, semantic search, multi-hop question answering, and structured data enrichment. While this problem has been studied across various paradigms, each method involves trade-offs between **accuracy**, **cost**, **interpretability**, and **scalability**.

LLM-Based Knowledge Graph Builders. Large Language Models (LLMs) are often used in the current state-of-the-art in KG construction to directly extract entity-relation triples from text. Tools like Neo4j's LLM Graph Builder [1] are capable of generating accurate, richly contextualized relationships. However, these systems are prohibitively expensive at the scale of the full Wikipedia corpus. Each article requires multiple model passes to extract structured data, leading to

- High inference cost per article,
- Infeasibility of full-corpus processing,
- Opaque decision-making and low explainability.

Thus, while LLMs are ideal for prototyping or small-scale applications, they remain unsuitable

for economically maintaining an up-to-date knowledge graph for Wikipedia.

Lazy and Hybrid Pipelines. To address cost concerns, hybrid methods such as LazyGraphRAG (Microsoft Research) [2] have emerged. These systems strategically minimize LLM usage by deploying lighter-weight operations for most content and reserving LLM calls for ambiguous or high-priority content. LazyGraphRAG achieves considerable cost savings while maintaining respectable extraction quality. However, it still inherits some weaknesses from LLM-centric designs, particularly in terms of reliance on pre-trained models and ongoing API costs. Furthermore, its hybrid architecture increases pipeline complexity and introduces new integration risks at scale.

TF-IDF-Based Knowledge Graph Construction. Classic statistical approaches such as TF-IDF-based KG construction [3] remain attractive due to their simplicity, computational efficiency, and domain-agnosticism. These methods identify key terms and co-occurrence-based relationships to bootstrap a graph from large corpora. However, they suffer from semantic shallowness, failing to distinguish between homonyms or subtle conceptual overlaps, which limits their usefulness for deep knowledge representation.

Agent-Oriented Temporal KG Models. Recent research has introduced agent-centric knowledge graph systems such as Graphiti [4], which supports temporally aware and incrementally updated knowledge structures. Graphiti excels at modeling state-based reasoning, dynamic memory updates, and context-aware search, capabilities useful for AI agents and enterprise applications. However, despite its efficiency in graph updating and retrieval, Graphiti still integrates LLMs during ingestion stages, making it costly for large-scale, continuous updates like those required by Wikipedia.

Our proposed research fills a critical gap in this landscape:

- It does not depend on LLMs for extraction, avoiding high costs and black-box reasoning.
- It goes beyond statistical keyword matching by training **domain-specific embeddings** that capture deep semantics and hierarchical structures.
- It innovates on top of brain-inspired models by developing a custom, **entropy-driven KG builder** optimized for large-scale dynamic knowledge representation.
- It is **fully unsupervised**, dynamically updatable, and computationally efficient.

This project introduces a novel architecture for constructing large-scale, unsupervised knowledge graphs that are both scalable and semantically meaningful. By addressing the limitations of existing methods in cost, interpretability, and adaptability, it aims to set a new standard for open-domain knowledge representation. While this approach offers significant advantages in efficiency and transparency, it also presents challenges, such as ensuring semantic richness without LLMs and managing structural complexity at scale. The following section outlines our methodology for meeting these challenges.

Methods

To address the challenges of constructing a dynamic, interpretable, and semantically rich large-scale knowledge graph (KG) from Wikipedia, we propose a two-step research architecture:

1. Designing an entropy-based, tree-structured graph builder that incrementally organizes Wikipedia knowledge with neural-like behavior.

2. Developing enhanced text embeddings capable of supporting unsupervised structure formation.

This architecture draws inspiration from brain-mimetic learning systems, entropy optimization, and epistemological grounding to construct a system that is both computationally efficient and cognitively interpretable.

1. Large-Scale Knowledge Graph Construction

At the core of our KG construction is a custom unsupervised clustering and node evolution system. Rather than using LLMs for relation extraction, we adapt decision-tree-inspired methodologies, tailored for text embeddings, to incrementally build and evolve the graph.

1.1 Architectural Inspirations and Design

We draw main ideas and core concepts selectively from the following models:

Isolation Forests [5]. Efficient for unsupervised anomaly detection, isolation-based heuristics are useful in identifying "concept boundaries" in vector space, which we adapt for identifying when new nodes should be created or existing clusters split.

Unsupervised Decision Trees. XAI Clustering [6] and Hierarchical Tree Clustering [7] construct decision trees without labeled supervision by using feature distributions and splitting criteria derived from intra-cluster variance or interpretability objectives. We adapt their core idea of interpretable rule-based cluster formation, applying it to high-dimensional text embeddings to form hierarchical structures that reflect conceptual granularity (e.g., topic → subtopic → article section).

Fuzzy Decision Trees [8]. We incorporate fuzzy logic principles in early prototypes to handle overlapping or ambiguous conceptual clusters.

Online Learning for Decision Trees. Online tree learning [9] allows the structure to evolve incrementally as new data arrives. We apply this methodology to simulate the “live” evolution of Wikipedia: as articles are added or updated, our system dynamically refines, splits, or merges nodes in the KG. The method supports scalability and avoids expensive retraining.

Neural Decision Trees [10]. These offer hierarchical clustering with differentiable split criteria. We do not adopt their full architecture but borrow the concept of adapting boundaries through online learning, useful in managing high-dimensional semantic data.

Brain-inspired Cortical Coding (.BIC) [11]. This framework guides the maturation or pruning of representational nodes, analogous to synaptic plasticity. .BIC inspires our entropy-driven node management, where nodes grow, merge, or forget based on information utility over time.

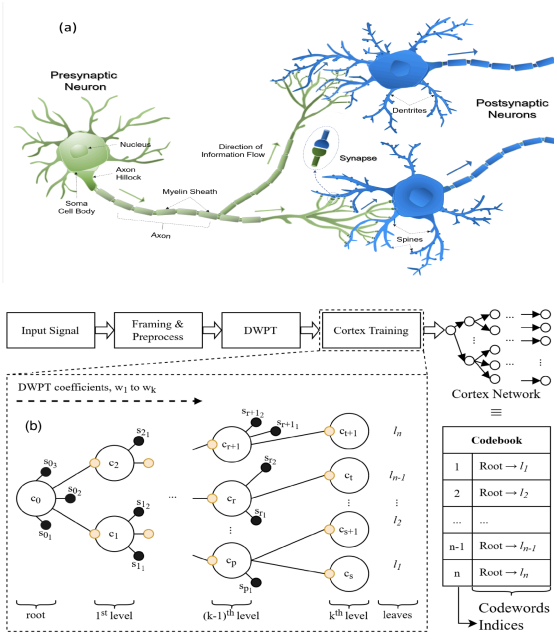

Fig. 2: *Biological neurons (a) and their equivalents (b) in the Brain-Inspired Coding system.*

1.2 Node Structure and Growth Behavior

Each node in the KG represents a latent concept composed of semantically similar Wikipedia text segments. Nodes evolve through:

Maturation. Frequently referenced or semantically central nodes mature, forming stable hubs.

Propagation. Matured nodes can branch to form child nodes when semantic variance within grows beyond a set entropy threshold.

Pruning. Nodes with low activity or minimal connectivity are removed—simulating "forgetting" and reducing noise.
Node connections are formed based on similarity, co-occurrence, and entropy minimization. The graph topology is maintained in a 3D relational format, echoing biological synaptic wiring, enabling both semantic (type-based) and spatial (contextual) queries.

1.3 Output Format and Tooling

The resulting knowledge graph will be structured for interoperability and scalability, with the following characteristics:
- **Typed Nodes:** Each node represents a distinct entity or concept, categorized into types such as Person, Place, Event, Concept, etc.
- **Labeled, Weighted Edges:** We label relationships between nodes (e.g., is_related_to, subclass_of, caused_by) and assign weights based on semantic strength or frequency.
- **Export Formats:** The full graph will be exportable in standard formats including Neo4j, RDF, and JSON-LD, supporting integration with Wikidata and Linked Open Data tools.
- **Temporal Metadata:** Nodes and edges will include time-aware annotations (e.g., last_updated, source_revision_id) to track provenance, support revision monitoring, and enable knowledge aging or decay mechanisms.

2. Enhanced Text Embeddings for Graph Support

To ensure that our graph builder works efficiently with high-dimensional semantic input, we first build a custom BERT-based embedding model tailored to the Wikipedia domain and graph clustering requirements. ModernBERT [12] is the most recent model, suggested for fine-tuning cases.

2.1 Motivation

Many unsupervised tree-based methods struggle with high-dimensional input vectors (e.g., 768–1536 dimensions). Rather than overcomplicating the clustering logic, we optimize the embedding vectors to be
- Multi-granular,
- Entropy-regularized,
- Domain-specific,
- Conceptually structured.

2.2 Innovations and Integrations

Matryoshka Representation Learning [13].
We use MRL to encode hierarchical semantic levels within a single embedding. This technique allows the graph builder to "zoom in" or "out" depending on node granularity, which supports flexible tree growth and merging.

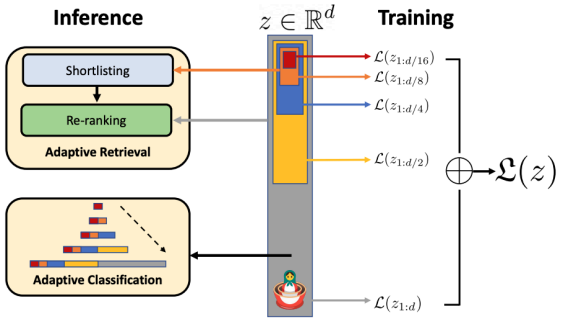

Fig. 3: *Different hierarchical components of Matryoshka Embeddings can be used in different levels of knowledge graphs.*

Domain-Specific Fine-Tuning [14]. Using Google Vertex AI's parameter-efficient embedding tuning, we fine-tune embeddings on Wikipedia

sections with weak supervision (e.g., article categories or linked Wikidata entities) to improve clustering fidelity.

Kantian Transcendental Categories [15]. Inspired by UPAR [16], we prompt an LLM to annotate or guide fine-tuning with Kant's four categories (quantity, quality, relation, modality), giving our embeddings an a priori conceptual scaffolding that generalizes well across domains. This idea may work synergistically with MRL embeddings and provide contextual metadata to their granular levels.

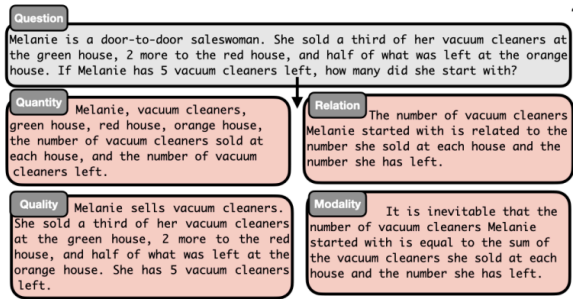

Fig. 4: *Extracting Kantian context for texts using LLMs.*

Entropy-Enriched Embeddings. MinEnt [17] applies an auxiliary entropy minimization loss during training to force more decisive representations. This technique may support more stable clustering and meaningful splits in the KG construction step.

Contrastive Learning Enhancements. Contrastive learning trains embeddings to pull semantically similar texts closer and push dissimilar ones apart, improving their clustering and retrieval quality. We can borrow ideas from GTE [18] for multi-stage unsupervised/supervised training, LLM-Synthetic Contrast [19] for generating pseudo-positive/negative pairs for Wikipedia segments without manual labels, and E5 [20] for mining Wikipedia hyperlinks and section co-occurrence as weak supervision to construct training pairs for large-scale contrastive pre-training.

2.3 Embedding Output Features

The final embedding model is designed to produce **multi-resolution Matryoshka vectors,** enabling flexible representation across different levels of semantic granularity. It is **fine-tuned specifically on Wikipedia content** to capture domain-specific semantics and contextual nuances unique to encyclopedic text. Additionally, the model integrates **Kantian transcendental structure**, allowing it to support reasoning-aware categorization inspired by foundational concepts in epistemology. To enhance clarity and robustness, the embeddings are both **entropy-regularized** and **contrastive-hardened,** ensuring semantic distinctiveness and reduced noise. The output can be extracted at varying levels (**entity-level**, **sentence-level**, **paragraph-level**, and **article-level),** providing adaptable input for downstream graph construction and analysis.

3. Data and Experimental Design

3.1 Data Collection

Our primary data source will be a curated dump of the English Wikipedia, comprising approximately 5 million articles. Each article will be segmented into structured components such as sections, infoboxes, and references to capture both narrative and factual content. For development and evaluation purposes, we will first work with a benchmark subset of around 50,000 high-activity articles. This subset will allow us to prototype the embedding and graph construction pipelines efficiently before scaling to the full corpus.

3.2 Experimental Setup

Our experiments will follow the two-stage process outlined in the Methods section: (1) embedding model training and (2) unsupervised knowledge graph construction. The primary focus will be algorithmic development and generative data processing, carried out using high-performance hardware on-premise. We deliberately avoid using cloud infrastructure due to the large volume of data involved, which could lead to substantial and unsustainable costs over time.

Phase	Tools	Tasks
Embedding Training	PyTorch + Hugging Face + Vertex AI	Fine-tune BERT with custom objectives
Graph Construction	Custom Python modules + Neo4j + NumPy	Online clustering, node evolution
Evaluation	FAISS, MTEB, BEIR, graph stats	Benchmark retrieval and clustering quality

Table 1: *Some of the proposed experiments for our methodology. We may add more experiments during the project.*

The experimental infrastructure includes two **H200 GPUs** (or equivalent) to support embedding fine-tuning and **LLM-based pseudo-label generation**, as well as a high-memory compute node (≥ 256 GB RAM) dedicated to processing and maintaining the evolving knowledge graph. We will also utilize **scalable object storage** to manage article dumps and persist large graph outputs.

The software stack includes **Python**, **PyTorch**, **Hugging Face Transformers**, **Neo4j** for graph storage and querying, and **FAISS** for similarity search. We will optionally utilize **Google Cloud Vertex AI** for embedding tuning during early-stage development. All components of the pipeline will be containerized using **Docker** to ensure reproducibility and portability across different systems.

Expected output

Our project will deliver several tangible and impactful outputs for both the Wikimedia ecosystem and the broader research community:

1. Public Embedding Model.
Audience: Wikimedia developers, researchers
Benefit: A reusable, domain-specific text embedding model optimized for semantic structure, tailored to Wikipedia content and knowledge graph use cases.

2. Wikipedia-Scale Knowledge Graph.
Audience: Wikidata maintainers, research communities, automated agents
Benefit: A scalable and interpretable graph of millions of facts, entities, and concepts extracted from Wikipedia in a cost-efficient and explainable manner.

3. Open-Source KG Builder.
Audience: AI/ML and NLP researchers
Benefit: A low-cost alternative to LLM-based knowledge graph extraction pipelines, based on unsupervised decision trees and entropy-driven node evolution.

4. Demo Dashboard.
Audience: Wikimedia editors
Benefit: An interactive tool to visualize, query, and explore the knowledge graph, supporting editorial tasks such as fact-checking, linking, and content expansion.

5. Research Publications.
Audience: Communities in NLP, Knowledge Graph Construction (KGC), and Human–Computer Interaction (HCI)
Benefit: Contributions to the theory and practice of domain-specific embeddings, contrastive training, and large-scale unsupervised graph learning.

6. PhD Dissertation Output.
Audience: Academic reviewers, AI research community
Benefit: This project constitutes the core of a computer science PhD thesis by Mustafa Abdullah Hakkoz, contributing to the academic understanding of neuro-inspired unsupervised knowledge modeling.

Additionally, we will produce documentation, reproducible code, and cost-comparison benchmarks showing up to 10x–100x savings over traditional LLM-based KG builders, validating the practical advantages of our approach.

Risks

Several challenges may arise throughout the execution of this project, given its reliance on advanced embedding techniques and unsupervised graph modeling at scale.

One key risk is that the **embedding model may underperform**, either due to suboptimal hyperparameters or insufficient semantic separation in the training data. To mitigate this, we will apply iterative tuning, expand the training dataset with additional Wikipedia content, and fall back to well-established open-source models if needed.

Another potential issue is that **tree-based graph builders may struggle with high-dimensional embedding vectors**, especially when dealing with subtle semantic differences. To address this, we will employ multi-resolution (Matryoshka) embeddings, consider dimensionality reduction techniques, and reinforce node separability using hybrid contrastive learning strategies.

The evolving nature of Wikipedia introduces the risk of **domain drift**, where newly added or updated content diverges from patterns learned during model training. We plan to integrate temporal context vectors and periodic re-embedding of updated content to help the graph adapt and stay current with editorial changes.

Lastly, **scaling the graph builder to the entire Wikipedia corpus** poses significant computational demands. We will mitigate these issues by developing the system incrementally, starting with smaller benchmark subsets, processing data in batches, and validating the architecture's efficiency before scaling up to full-scale runs.

Community impact plan

This project offers a transformative contribution to the Wikimedia ecosystem by introducing a scalable, interpretable, and cost-effective approach to constructing a Wikipedia-scale knowledge graph (KG). Unlike traditional LLM-based systems, which are computationally expensive and opaque, our neuro-inspired, unsupervised framework enables the creation of verifiable and dynamic knowledge structures aligned with Wikimedia's core principles of transparency, openness, and adaptability.

Each node and relationship in the KG is generated through explainable rules based on entropy-driven clustering and semantic coherence, providing clear justification for connections between concepts. This design supports Wikimedia's goal of maintaining **auditable and trustworthy knowledge.**

The outputs of this project map directly to many Wikimedia use cases. To maximize real-world adoption and community engagement, we are implementing a focused **community impact strategy**:

1. The **Wikipedia-scale KG** can enhance **Wikidata** through structured suggestions for entity relationships, subclass hierarchies, and temporal data.

2. We will **integrate the KG outputs into Wikidata ingestion workflows**, allowing both bots and human editors to semi-automatically enhance structured data coverage.

3. The **public domain-specific embedding model** enables Wikimedia developers and tool builders to perform semantic similarity search, entity linking, and fact retrieval more effectively.

4. The **open-source KG builder** offers a reusable, low-cost alternative for creating and updating knowledge graphs across multiple Wikimedia projects.

5. A v**isual dashboard** allows editors to explore entity neighborhoods, uncover related topics, identify semantic gaps, validate page relationships, and support more cohesive and richly interlinked article development.

6. We will host **online workshops** tailored for Wikimedia volunteers, editors, and developers to introduce KG-assisted editing, graph querying, and embedding-based recommendations.

7. The project will also generate a computer science PhD dissertation along with two research papers, advancing academic research on knowledge representation, embeddings, and unsupervised learning at web scale.

8. All code, models, and documentation will be released under **permissive open-source licenses** (MIT and CC BY-SA), allowing for transparent reuse, auditing, and long-term maintenance.

By bridging state-of-the-art unsupervised learning with practical Wikimedia needs, this project not only contributes technically but also **empowers the global Wikimedia community** with new tools for structuring, curating, and expanding the world's largest collaborative knowledge base.

Evaluation

We will evaluate the success of this project based on four core criteria: embedding quality, graph coverage and structure, computational efficiency, and Wikimedia utility. Embedding performance will be measured using standard benchmarks such as **MTEB** and **BEIR**, while ablation studies will help isolate the impact of specific design choices. The quality of the knowledge graph will be assessed through its **coverage of Wikipedia concepts**, coherence of clustered entities, and its ability to evolve with new content. We will compare our system's **cost and runtime** against LLM-based baselines to validate its efficiency and scalability. Finally, community impact will be gauged by **feedback from editors** using the graph interface, adoption of our open-source tools, and integration into Wikimedia workflows. Success will be defined by the delivery of a public embedding model, a large-scale interpretable KG, and clear uptake by the Wikimedia community.

Budget

Use the Wikimedia Foundation's budget [21] Our total project budget is $150,000 USD, detailed in

the Wikimedia Foundation budget template. The budget includes
- **Personnel Support:** $36,000 for 3 researchers over 24 months.
- **Dissemination Activities:** $600 for presenting at Wikimania and $5,000 for open-access publishing.
- **Compute Resources:** $108,400 for 2× NVIDIA H200 GPUs, a GPU server, and a high-memory compute node with 24TB storage.

Why do we purchase hardware instead of renting or using cloud services? Building Wikipedia-scale knowledge graphs and training custom embeddings require intensive GPU compute and large storage capacity. Renting equivalent cloud infrastructure (e.g., H100/H200 GPUs with 20+ TB storage) can cost tens of thousands of dollars monthly and limits long-term access. In contrast, purchasing hardware offers unlimited usage, full control over performance and data, and sustainability for future Wikimedia-aligned research. It also ensures reproducibility and avoids recurring costs.

Why is hardware pricier in Turkey? While GPU prices in the U.S. are already high (e.g., ~$32,000 per H200), the cost in Turkey increases significantly due to a **20% VAT** and **8% customs tax** on imported electronics. Combined with currency fluctuations and market markups, hardware costs can be **30% above U.S. retail prices**. Despite this, local purchase remains the most practical and transparent option for sustained, secure research infrastructure due to extended guarantee services.

References

[1] Neo4j Labs. Neo4j LLM Knowledge Graph Builder - Extract Nodes and Relationships from Unstructured Text. Neo4j, https://neo4j.com/labs/genai-ecosystem/llm-graph-builder/. Accessed 16 Apr. 2025.

[2] Edge, Darren, Ha Trinh, and Jonathan Larson. “LazyGraphRAG: Setting a New Standard for Quality and Cost.” Microsoft Research Blog, 25 Nov. 2024, https://www.microsoft.com/en-us/research/blog/lazygraphrag-setting-a-new-standard-for-quality-and-cost/. Accessed 16 Apr. 2025.

[3] Wang, Yu, et al. Knowledge Graph Prompting for Multi-Document Question Answering. 2023. arXiv, https://arxiv.org/abs/2308.11730.

[4] Rasmussen, Preston, et al. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. 2025. arXiv, https://arxiv.org/abs/2501.13956.

[5] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation Forest.” Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM), IEEE, 2008, pp. 413–422. doi:10.1109/ICDM.2008.17.

[6] Loyola-González, Octavio, et al. “An Explainable Artificial Intelligence Model for Clustering Numerical Databases.” IEEE Access, vol. 8, 2020, pp. 52370–52384. IEEE, https://doi.org/10.1109/ACCESS.2020.2980581.

[7] Basak, Jayanta, and Raghu Krishnapuram. “Interpretable Hierarchical Clustering by Constructing an Unsupervised Decision Tree.” IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 1, Jan. 2005, pp. 121–132. IEEE, https://doi.org/10.1109/TKDE.2005.11.

[8] Jiao, Lianmeng, et al. “Interpretable Fuzzy Clustering Using Unsupervised Fuzzy Decision Trees.” *Information Sciences*, vol. 611, 2022, pp. 540–563. Elsevier, https://doi.org/10.1016/j.ins.2022.08.077.

[9] Held, Marcus, and Joachim M. Buhmann. “Unsupervised On-Line Learning of Decision Trees for Hierarchical Data Analysis.” Advances in Neural Information Processing Systems, Dec. 1997, pp. 514–520. https://papers.nips.cc/paper/1479-unsupervised-on-line-learning-of-decision-trees-for-hierarchical-data-analysis.pdf.

[10] Balestriero, Randall. Neural Decision Trees. 2017. arXiv, https://arxiv.org/abs/1702.07360.
[11] Yucel, Meric, Serdar Bagis, Ahmet Sertbas, Mehmet Sarikaya, and Burak Berk Ustundag. "Brain Inspired Cortical Coding Method for Fast Clustering and Codebook Generation." Entropy, vol. 24, no. 11, 2022, article 1678. MDPI, https://doi.org/10.3390/e24111678.
[12] Warner, Benjamin, et al. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. 2024. arXiv, https://arxiv.org/abs/2412.13663.
[13] Kusupati, Aditya, et al. Matryoshka Representation Learning. 2024. arXiv, https://arxiv.org/abs/2205.13147.
[14] Google Cloud. "Tune Text Embeddings." Generative AI on Vertex AI, 16 Apr. 2025, https://cloud.google.com/vertex-ai/generative-ai/docs/models/tune-embeddings.
[15] "Category (Kant)." Wikipedia, Wikimedia Foundation, 16 Apr. 2025, https://en.wikipedia.org/wiki/Category_(Kant). Accessed 16 Apr. 2025.
[16] Geng, Hejia, Boxun Xu, and Peng Li. UPAR: A Kantian-Inspired Prompting Framework for Enhancing Large Language Model Capabilities. 2023. arXiv, https://arxiv.org/abs/2310.01441.
[17] Li, Shuo, et al. "MinEnt: Minimum Entropy for Self-Supervised Representation Learning." Pattern Recognition, vol. 138, 2023, article 109364. Elsevier, https://doi.org/10.1016/j.patcog.2023.109364.
[18] Li, Zehan, et al. Towards General Text Embeddings with Multi-Stage Contrastive Learning. 2023. arXiv, https://arxiv.org/abs/2308.03281.
[19] Wang, Liang, et al. Improving Text Embeddings with Large Language Models. 2024. arXiv, https://arxiv.org/abs/2401.00368.
[20] Wang, Liang, et al. Text Embeddings by Weakly-Supervised Contrastive Pre-Training. 2024. arXiv, https://arxiv.org/abs/2212.03533.
[21] Research Fund Budget (Whitehand AI Research Group). Google Sheets, https://docs.google.com/spreadsheets/d/1AX43qtxNho_4_O2otr73iRSnOUw9ovp7zzZp4zQ_6p0/edit?usp=sharing. Accessed 16 Apr. 2025.