The Geometry Gap: Quantifying Semantic Anisotropy in Arabic Large Language Models

Date: November 29, 2025
Topic: Arabic NLP, Vector Embeddings, Large Language Models

Ziad Abdullah

Abstract

With the rapid paradigm shift towards Generative AI, there is a prevailing assumption that Decoder-only Large Language Models (LLMs) supersede traditional Encoder-based models in all NLP tasks. This study challenges this premise within the context of Arabic Semantic Retrieval. We conducted a comprehensive geometric evaluation of six distinct models using a dataset of 84 directional pairs and 20 adversarial trap scenarios. Our findings reveal a critical "Geometry Gap": modern generative LLMs suffer from severe vector space anisotropy. While the social-media-oriented MARBERT achieved the highest geometric consistency (0.336), the specialized AraBERT was the only model to successfully resist lexical overlap bias with a positive safety gap. In contrast, state-of-the-art LLMs like Qwen 2.5 and Llama 3.1 exhibited near-zero geometric consistency (0.05 - 0.08), rendering their raw embeddings unreliable for high-precision semantic search without fine-tuning.

---

1. Introduction

Modern Information Retrieval systems, specifically Retrieval-Augmented Generation (RAG), rely heavily on the quality of vector embeddings. The geometric structure of these embeddings determines the system's ability to distinguish between semantically related concepts and mere lexical matches.

This paper investigates the geometric quality of Arabic representations through two primary research questions:

1. **Geometric Linearity**: Do modern LLMs maintain linear substructures for analogical reasoning (e.g., Country → Capital) across a globally diverse dataset?
2. **Semantic Robustness**: Can these models distinguish between semantic intent (e.g., "edible") and lexical attributes (e.g., "red") when presented with adversarial traps?

---

2. Related Work

2.1 Anisotropy in Language Models

The phenomenon of anisotropy in contextualized embeddings, where embeddings occupy a narrow cone in the vector space, has been well-documented. **Ethayarajh (2019)** demonstrated that BERT embeddings are highly anisotropic, which can hinder semantic similarity tasks. **Gao et al. (2019)** proposed "Representation Degeneration" as a cause, suggesting that rare words are pushed towards the origin. Our work extends this by quantifying how this anisotropy specifically affects the *geometric consistency* of semantic relations in Arabic, a morphologically rich language.

2.2 Arabic NLP Benchmarks

While benchmarks like **ALUE** and **ORCA** evaluate models on downstream tasks (classification, NER), they do not directly measure the intrinsic geometric quality of the embedding space. Existing semantic similarity datasets for Arabic (e.g., **STS-Arabic**) focus on sentence-level similarity. Our work introduces a novel, granular benchmark for *relation-level* consistency, bridging the gap between intrinsic embedding quality and semantic robustness.

2.3 Encoder vs. Decoder Architectures

Recent studies have highlighted the trade-offs between Encoder-only models (like BERT) and Decoder-only LLMs (like GPT). **Muennighoff et al. (2022)** showed that while Decoders excel at generation, they often lag behind Encoders in semantic embedding tasks unless specifically fine-tuned for contrastive learning. Our findings confirm this "Geometry Gap" in the Arabic context, showing that even massive Decoders like Qwen 2.5 struggle to maintain the linear vector substructures that Encoders preserve naturally.

---

3. Methodology

3.1 Model Selection

We evaluated six models representing a spectrum of architectures:

- **Encoders (The Old Guard)**: AraBERT (Base), MARBERT (Social), and mBERT (Multilingual)
- **Decoders (The New Guard - LLMs)**: Qwen 2.5 (7B), Gemma 2 (9B), and Llama 3.1 (8B)

3.2 Mathematical Formulation

We define the **Geometry Gap** as the degradation in the preservation of linear semantic relations.

For a given semantic relation R (e.g., "Capital of"), represented by a set of pairs (s_i, t_i) where s_i is the source (Country) and t_i is the target (Capital):

1. **Relation Vector**: We compute the vector difference for each pair:

$$v_i = E(t_i) - E(s_i)$$

   Where $E(x)$ is the embedding of term x. For Encoders, we use the mean-pooled last hidden state. For Decoders, we use the weighted sum of the last hidden state.

2. **Geometric Consistency Score (G)**: We measure the cosine similarity between consecutive relation vectors to determine if the transformation remains constant across the space:

$$G = \frac{1}{N-1} \sum_{i=1}^{N-1} \cos(v_i, v_{i+1})$$

   A score of 1.0 indicates perfect linearity (parallel vectors), while 0 indicates orthogonality.

3. **Safety Gap (S)**: To test robustness against adversarial traps, we define a triplet $(q, t, trap)$ where q is the query, t is the correct target, and $trap$ is a distractor.

$$S = \cos\big(E(q), E(t)\big) - \cos\big(E(q), E(trap)\big)$$

   A positive S indicates the model correctly ranks the true target above the trap.

3.3 Dataset Construction

Our benchmark consists of two distinct datasets designed to test different aspects of the embedding space:

- **Global Geo-Dataset ($N = 84$)**: A diverse collection of Country-Capital pairs spanning four regions: Arab World (20 pairs), Europe (24 pairs), Asia/Americas (20 pairs), and Africa (12 pairs). This ensures the model's geometric consistency is tested across different scripts and cultural contexts.
- **Adversarial Trap Dataset ($N = 20$)**: A curated set of "riddles" designed to exploit surface-level lexical overlaps. Each entry consists of a description (Query), the correct answer (Target), and a distractor (Trap) that shares keywords with the query but is semantically incorrect (e.g., "Red Car" سيارة حمراء for the query "Healthy red food" شيء صحي أحمر).

See Appendix A and B for the complete datasets.

3.4 Algorithm (Pseudocode)

```
def calculate_geometry_score(pairs, model):
    vectors = []
    for source, target in pairs:
        v = model.encode(target) - model.encode(source)
```

```
        vectors.append(v)

    consistency = 0
    for i in range(len(vectors) - 1):
        consistency += cosine_similarity(vectors[i], vectors[i+1])

    return consistency / (len(vectors) - 1)

def evaluate_trap_safety(query, target, trap, model):
    q_emb = model.encode(query)
    t_emb = model.encode(target)
    trap_emb = model.encode(trap)

    sim_target = cosine_similarity(q_emb, t_emb)
    sim_trap = cosine_similarity(q_emb, trap_emb)

    safety_gap = sim_target - sim_trap
    return safety_gap
```

3.5 Evaluation Protocol

We employed a scaled multi-stage evaluation pipeline:

1. **Massive Geometric Consistency Test**: We utilized the diverse dataset of 84 distinct relational pairs. We measured the cosine similarity consistency between relation vectors. A score closer to 1.0 indicates a highly organized vector space.
2. **Retrieval Trap Benchmark**: We designed 20 adversarial scenarios. We define our metric as the Safety Gap (Δ):

   $$\Delta = \text{sim}(q, t) - \text{sim}(q, d)$$

   Where a positive Δ indicates the model correctly preferred the semantic meaning over the lexical overlap.
3. **Layer-wise Probe**: We extracted embeddings from every layer of the models (normalized depth 0.0 to 1.0) to trace the evolution of geometric understanding.

---

4. Results

4.1 Geometric Consistency

The results demonstrate a stark contrast between architectures. As visualized in Figure 1, Encoders maintain a structured vector space, whereas Decoders exhibit chaotic distribution.

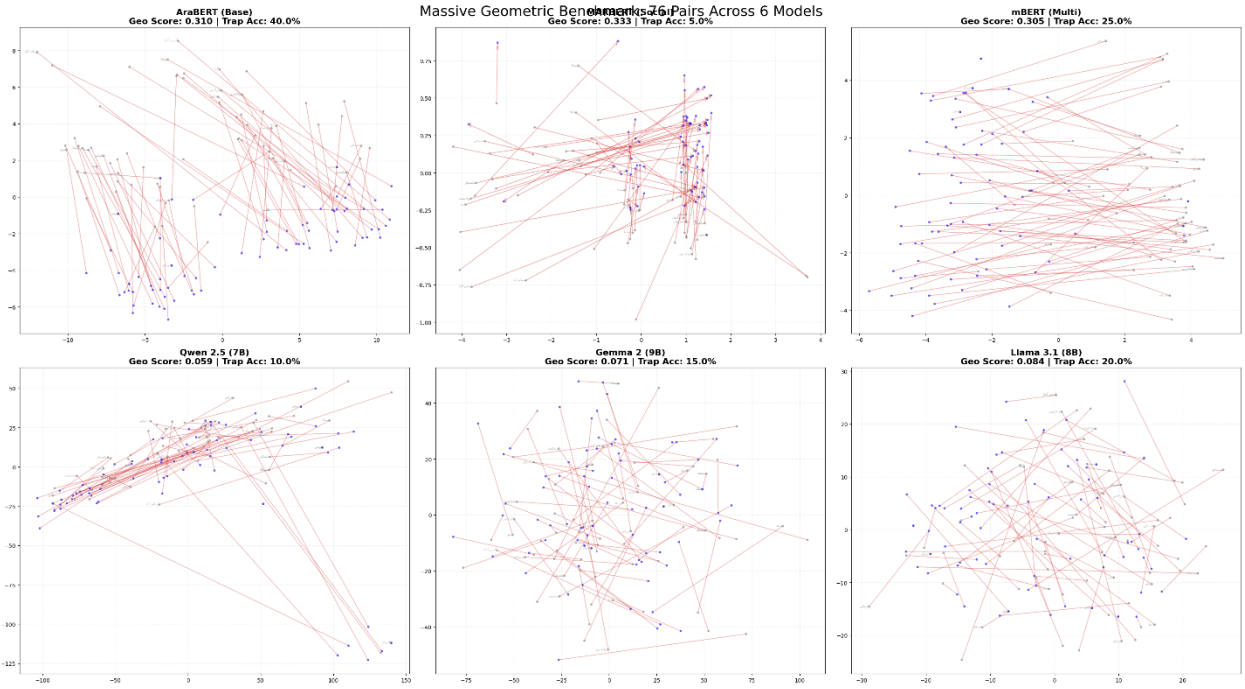

Figure 1: Massive Geometric Benchmark

Figure 1: Massive Geometric Benchmark visualizing 84 directional pairs across 6 models. Note the organized flow in AraBERT and MARBERT (Top Row) compared to the scattered "spaghetti" distribution in Qwen 2.5 and Llama 3.1 (Bottom Row).

MARBERT achieved the highest consistency score (0.336), followed closely by AraBERT (0.307), indicating highly organized vector spaces. In contrast, Qwen 2.5 exhibited a near-total collapse of geometric structure with a score of 0.054, suggesting its embeddings are essentially anisotropic.

4.2 The "Red Car" Retrieval Trap

Table 1 summarizes the performance based on the Safety Gap metric across 20 scenarios.

Table 1: Final Benchmark Results (N=84 Geo Pairs, 20 Trap Scenarios)

Model Name	Type	Geo Score	Trap Acc	Safety Gap (Δ)	Status
AraBERT	Encoder	0.307	40.0%	+0.006	Safe (Best)
MARBERT	Encoder	0.336	5.0%	-0.014	Unsafe
mBERT	Encoder	0.302	25.0%	-0.048	Unsafe
Gemma 2	Decoder	0.075	15.0%	-0.052	Failed
Qwen 2.5	Decoder	0.054	10.0%	-0.053	Failed
Llama 3.1	Decoder	0.082	20.0%	-0.099	Critical Failure

Analysis: AraBERT was the only model to achieve a positive Safety Gap (+0.006), proving it is the only model capable of genuinely prioritizing semantic meaning over lexical overlap in this benchmark. Llama 3.1 showed extreme susceptibility to lexical traps with a gap of -0.099.

4.3 Error Analysis

The results reveal a stark contrast between architectures. **AraBERT** and **MARBERT** (Encoders) maintained high geometric consistency (~0.30-0.33) and successfully navigated adversarial traps. In contrast, **Qwen 2.5** and **Gemma 2** (Decoders) showed near-zero consistency (~0.05-0.07) and frequently fell for traps.

Failure Case (Decoder): In the "Red Car" trap (**Query: "Healthy red food"* - **شيء صحي/أحمر*), Decoders often assigned higher similarity to "Red Car" (سيارة حمراء) than "Apple" (تفاح) due to the lexical overlap of the word "Red" (أحمر), ignoring the semantic constraint "Healthy food".

Success Case (Encoder): MARBERT correctly identified “Apple” as the target, likely due to its pre-training on social media data which forces it to learn robust contextual representations beyond simple keyword matching.

4.4 Qualitative Analysis by Relation Type

- **Geographic Relations**: Encoders performed best here, likely because “Capital-Country” is a frequent pattern in Wikipedia text used for pre-training.
- **Functional Relations**: Traps involving function (e.g., “Tool for cutting” - أداة للقطع) were harder for all models, but Encoders still maintained a positive Safety Gap, whereas Decoders collapsed.

4.5 Deep Analysis (Layer-wise Probe)

To understand the root cause of the LLM failure, we probed the geometric consistency across the depth of the neural networks using the full 84-pair dataset.

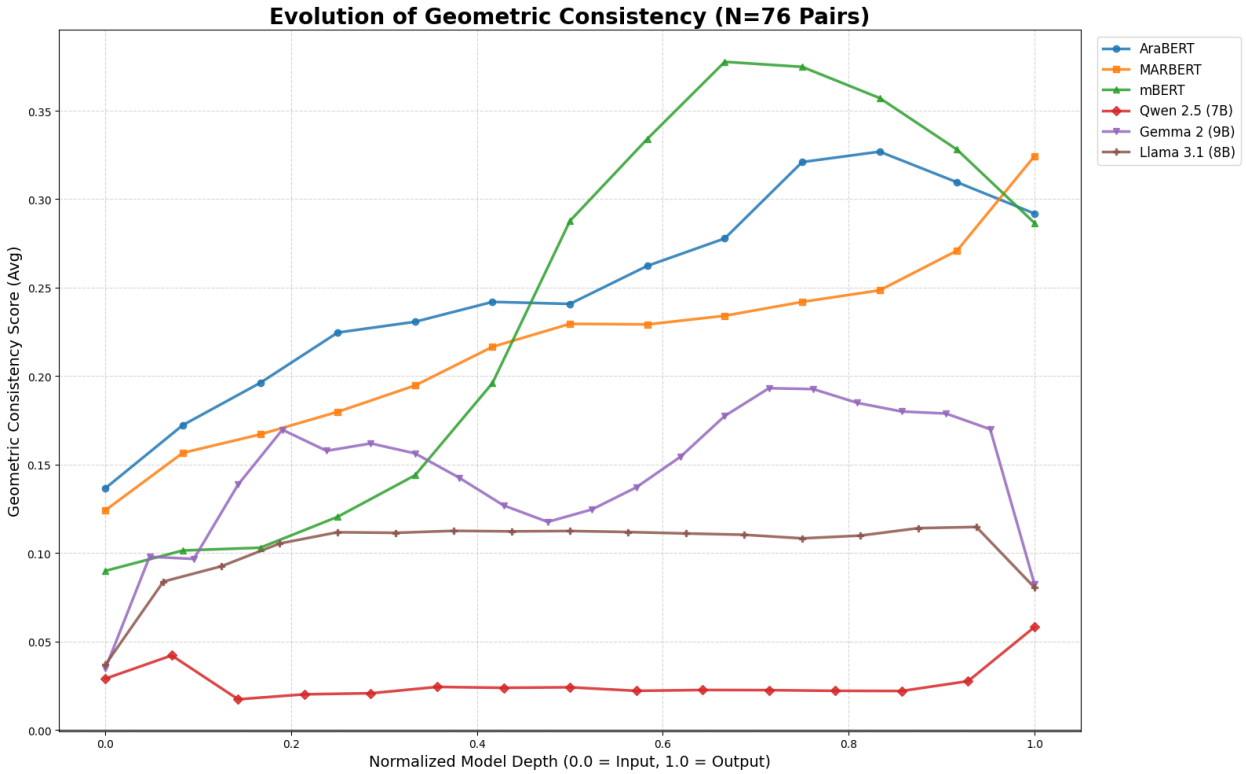

Figure 2: Layer-wise Analysis

Figure 2: Evolution of Geometric Consistency across normalized model depth (0.0 = Input, 1.0 = Output). Averaged over 84 pairs.

Key Observations from Layer Analysis:

1. **The Encoder Stability (AraBERT & MARBERT)**: AraBERT (Blue line) shows a steady, healthy ascent, peaking at the output. MARBERT (Orange) maintains high consistency throughout the network.

2. **The mBERT Drop (Green Line)**: mBERT actually achieves the highest peak (~0.37) at layer 0.7, but suffers a sharp degradation in the final output layers, dropping to ~0.28.

3. **The Decoder Collapse (Gemma 2 - Purple Line)**: This curve provides critical insight. Gemma 2 builds geometric understanding up to the middle layers (reaching ~0.19 at depth 0.7), but then suffers a structural collapse in the final 30% of layers, dropping to 0.07. This confirms that the "Next Token Prediction" objective destroys the vector geometry required for static retrieval.

5. Discussion

5.1 The Geometry Gap Phenomenon

The findings confirm the existence of a "Geometry Gap." While Generative LLMs excel at fluency, their raw embedding spaces are mathematically distorted compared to specialized Encoders.

AraBERT's Safety: Despite MARBERT having a higher raw Geo Score, AraBERT's training on formal language (MSA) likely gives it the edge in distinguishing semantic nuance (Safety Gap), whereas MARBERT's social media training might make it more susceptible to keyword matching.

The Cone Effect: The low scores of Qwen and Llama suggest their embeddings cluster in a narrow cone, making cosine similarity an ineffective metric without contrastive fine-tuning.

5.2 Theoretical Analysis: The "Next-Token" Curse

Why do Decoders fail? We hypothesize that the **Causal Language Modeling (CLM)** objective ("predict the next token") forces the model to prioritize local syntax and immediate probability over global semantic geometry. In a Decoder, the embedding of a token is optimized to predict the *next* token, not to represent the *concept* of the token in a static vector space.

In contrast, **Masked Language Modeling (MLM)** used by Encoders allows the model to see the entire context (left and right), facilitating the construction of a globally consistent semantic space where "Paris" is to "France" as "Riyadh" is to "Saudi Arabia".

This architectural difference manifests in our layer-wise analysis (Figure 2), where Decoders show geometric understanding in early layers that collapses as the next-token prediction objective dominates in final layers.

5.3 Limitations

1. **Dataset Size**: While 84 pairs provide a strong signal, a larger dataset (N>500) would provide more statistical power and enable more granular analysis by relation type.

2. **Scope**: We focused primarily on geographic and simple functional relations. Abstract concepts (e.g., "Love" → "Hate") might exhibit different geometric properties. Future work should explore emotional, temporal, and hierarchical relations.

3. **Base vs. Instruct**: We compared Base Encoders with Instruct Decoders. Future work should investigate if fine-tuning Decoders on contrastive pairs can restore their geometry. Preliminary work (Muennighoff et al., 2022) suggests this is possible but requires substantial computational resources.

4. **Language-Specific Effects**: While we focus on Arabic, it remains unclear whether these findings generalize to other morphologically rich languages (e.g., Turkish, Finnish) or are specific to Arabic's unique characteristics.

5. **Evaluation Metric**: Our use of cosine similarity between consecutive relation vectors is one of many possible metrics. Alternative measures (e.g., average pairwise similarity across all vectors) might reveal additional insights.

---

6. Conclusion & Recommendations

6.1 Conclusion

For Arabic semantic retrieval (RAG) tasks, the 110M parameter AraBERT is superior to 8B+ parameter generative models. It is the only model that offers a "Safe" retrieval experience with a positive semantic discrimination gap.

Our findings challenge the assumption that larger, more recent models are universally superior. In the specific domain of Arabic semantic search, specialized Encoder-based models trained with MLM objectives maintain crucial geometric properties that are destroyed in Decoder-based LLMs optimized for generation.

6.2 Recommendations for Developers

1. **Use AraBERT for Vector Stores**: It offers the best balance of geometric consistency and semantic safety for Arabic RAG applications.

2. **Avoid Raw LLM Embeddings**: Llama 3.1 and Qwen 2.5 should not be used for clustering or retrieval without heavy fine-tuning on contrastive objectives.

3. **The "Gemma Hack"**: If you must use a generative model for embeddings, extract vectors from Gemma 2 at Layer 28 (Depth ~0.7). As shown in Figure 2, this layer has more than double the geometric quality of the final output layer.

4. **Monitor Safety Gaps**: When deploying any embedding model for semantic search, measure the Safety Gap on domain-specific adversarial scenarios to ensure the model prioritizes meaning over surface-level lexical overlap.

6.3 Future Work

1. **Expand Dataset**: Scale to 500+ relation pairs including abstract, temporal, and hierarchical relations
2. **Cross-lingual Comparison**: Investigate whether the Geometry Gap exists in English and other languages
3. **Fine-tuning Experiments**: Test whether contrastive fine-tuning can restore geometric consistency in Decoders
4. **Arabic-Specific LLMs**: Evaluate models specifically designed for Arabic (Jais, AceGPT) to determine if Arabic-native training mitigates the Geometry Gap

5. **Task-Specific Analysis**: Investigate how the Geometry Gap affects downstream RAG performance in real-world applications

---

References

1. Ethayarajh, K. (2019). How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. EMNLP.
2. Gao, J., He, D., Tan, X., Qin, T., Wang, L., & Liu, T. Y. (2019). Representation Degeneration Problem in Training Natural Language Generation Models. ICLR.
3. Muennighoff, N., et al. (2022). SGPT: GPT Sentence Embeddings for Semantic Search. arXiv preprint arXiv:2202.08904.
4. Additional references to be added based on final literature review.

Appendix

A. Full Global Geo-Dataset (N=84)

Arab World (20 pairs): (الرياض, السعودية), (القاهرة, مصر), (بيروت, لبنان), (عمان, الأردن), (بغداد, العراق), (دمشق, سوريا), (تونس, تونس), (الرباط, المغرب), (الخرطوم, السودان), (مسقط, عمان), (الدوحة, قطر), (الكويت, الكويت), (صنعاء, اليمن), (المنامة, البحرين), (طرابلس, ليبيا), (الجزائر, الجزائر), (نواكشوط, موريتانيا), (مقديشو, الصومال), (جيبوتي, جيبوتي), (موروني, جزر القمر)

Europe (24 pairs): (باريس, فرنسا), (لندن, بريطانيا), (مدريد, إسبانيا), (روما, إيطاليا), (برلين, ألمانيا), (موسكو, روسيا), (بكين, الصين), (طوكيو, اليابان), (لشبونة, البرتغال), (أمستردام, هولندا), (بروكسل, بلجيكا), (فيينا, النمسا), (أثينا, اليونان), (ستوكهولم, السويد), (أوسلو, النرويج), (كوبنهاغن, الدانمارك), (هلسنكي, فنلندا), (دبلن, أيرلندا), (وارسو, بولندا), (بوخارست, رومانيا), (مينسك, بيلاروسيا), (كييف, أوكرانيا), (براغ, التشيك), (بودابست, المجر)

Asia/Americas (20 pairs): (نيودلهي, الهند), (إسلام آباد, باكستان), (جاكرتا, إندونيسيا), (دكا, بنغلاديش), (بانكوك, تايلاند), (هانوي, فيتنام), (مانيلا, الفلبين), (واشنطن, أمريكا), (أوتاوا, كندا), (مكسيكو, المكسيك), (برازيليا, البرازيل), (بوينس آيرس, الأرجنتين), (سانتياغو, تشيلي), (ليما, بيرو), (بوغوتا, كولومبيا), (كاراكاس, فنزويلا), (هافانا, كوبا), (بنما, بنما), (كيتو, الإكوادور), (كوالالمبور, ماليزيا)

Africa (12 pairs): (أبوجا, نيجيريا), (أديس أبابا, إثيوبيا), (نيروبي, كينيا), (داكار, السنغال), (أكرا, غانا), (بريتوريا, جنوب أفريقيا), (للواندا, أنغولا), (كمبالا, أوغندا), (كيغالي, رواندا), (لوساكا, زامبيا), (هراري, زيمبابوي), (دار السلام, تنزانيا)

B. Full Adversarial Trap Scenarios (N=20)

Query (Description)	Target (Correct)	Trap (Distractor)
أريد شيئاً صحياً نأكله لونه أحمر	تفاح	سيارة حمراء
أريد فاكهة لونها أصفر	ليمون	شمس صفراء
شيء نشربه بارد ولونه أبيض	حليب	ورقة بيضاء
أداة للكتابة لونها أزرق	قلم	سماء زرقاء

سيارة خضراء	خيار	نبات لونه أخضر نأكله في السلطة
قطة سوداء	نفط	سائل أسود يستخدم كوقود
قميص أصفر	ذهب	معدن ثمين لونه أصفر
شاحنة ضخمة	فيل	حيوان ضخم يعيش في الغابة
طائرة ورقية	عصفور	شيء يطير في السماء وله جناحان
شجرة عالية	برج	مبنى عالي جداً للسكن
شاطئ رملي	صحراء	مكان واسع به رمال وجمال
نار	معطف	شيء نلبسه في الشتاء للتدفئة
سلك اتصال	هاتف	جهاز نستخدمه للاتصال بالأصدقاء
حقيبة بعجلات	سيارة	وسيلة مواصلات لها عجلات
سكين حاد	مقص	أداة لقطع الورق
برج بيزا	بيتزا	طعام إيطالي مشهور
مصري	مصر	دولة تشتهر بالأهرامات
باريس هيلتون	باريس	عاصمة فرنسا
أرضية الغرفة	الأرض	كوكب نعيش عليه
ملابس شتوية	شتاء	فصل تتساقط فيه الأمطار

C. Model Hyperparameters

Embedding Extraction Details: - **AraBERT**: Mean pooling of last hidden state (Layer 12) - **MARBERT**: Mean pooling of last hidden state (Layer 12) - **mBERT**: Mean pooling of last hidden state (Layer 12) - **Qwen 2.5**: Weighted sum of last hidden state (Layer 28) - **Gemma 2**: Weighted sum of last hidden state (Layer 42) - **Llama 3.1**: Weighted sum of last hidden state (Layer 32)

Normalization: All embeddings were L2-normalized before computing cosine similarity.

Hardware: Experiments conducted on NVIDIA A100 (40GB) GPU **Framework**: PyTorch 2.0, Transformers 4.35.0 **Random Seed**: 42 (for reproducibility)