

## Introduction to Large Language Models

### Assignment- 9

Number of questions: 10

Total mark: 10 X 1 = 10

---

#### QUESTION 1: [1 mark]

Which of the following statement best describes why knowledge graphs (KGs) are considered more powerful than a traditional relational knowledge base (KB)?

- a. KGs require no schema, whereas KBs must have strict schemas.
- b. KGs store data only in the form of hypergraphs, eliminating redundancy.
- c. KGs allow flexible, graph-based connections and typed edges, enabling richer relationships and inferences compared to KBs.
- d. KGs completely replace the need for textual sources by storing all possible facts.

**Correct Answer:** c

**Explanation:**

- Traditional relational knowledge bases enforce strict schemas (rows, columns, tables). In contrast, KGs store entities as nodes with typed edges (relations) between them, allowing richer and more flexible relationships.
  - KGs can represent complex connections (e.g., multi-edges, different relation types) and support inference by traversing these connections.
  - While some knowledge graphs can be partially schema-less or schema-flexible, choice (c) specifically highlights the flexibility in graph connections and typed edges, which is what makes KGs generally more powerful for representing rich, interlinked data.
- 

#### QUESTION 2: [1 mark]

Entity alignment and relation alignment are crucial between KGs of different languages. Which of the following factors contribute to effective alignment?

- a. Aligning relations solely by their lexical similarity, ignoring semantic context
- b. Transliteration or language-based string matching for entity labels
- c. Ensuring all language aliases are represented identically in each KG
- d. Matching neighbours, or connected entities, across different KGs

**Correct Answer:** b, d

**Explanation:**

- **Transliteration or language-based string matching (b):** For multilingual KGs, matching entity labels across languages (e.g., transliterating names or matching synonyms) is key to identifying equivalent entities.
- **Matching neighbours (d):** Alignment goes beyond just matching labels; you also look at the graph structure - if two entities are neighbours or connected to the same concept in each KG, that supports they might be the same or related.
- **Why not a or c?**

(a) Lexical similarity alone (ignoring context) is insufficient, as terms might match lexically but differ in meaning across languages.

(c) It is often not possible or necessary to have identical labels for all language aliases if they mean the same thing in context; the important aspect is linking them, not forcing identical representations.

### QUESTION 3: [1 mark]

In the context of knowledge graph completion (KGC), which statement best describes the role of the scoring function  $f(s, r, o)$ ?

- It determines whether two entities refer to the same real-world concept.
- It produces a raw confidence score indicating how plausible a triple  $(s, r, o)$  is.
- It explicitly encodes only the subject's embedding, ignoring the relation and object embeddings.
- It ensures that every negative triple gets a higher score than any positive triple.

**Correct Answer:** b

**Explanation:**

- **A raw confidence score for plausibility (b):** In KGC, the scoring function evaluates a triple  $(s, r, o)$  (subject, relation, object) to determine how likely it is to be true according to the learned embeddings. A higher (or lower, depending on the convention) score indicates higher plausibility.

- **Why not the others?**

(a) That relates to entity resolution, not necessarily the KGC scoring function.

(c) The scoring function typically factors in all three embeddings (subject, relation, and object).

(d) While in training we often prefer valid triples to have higher scores than negative ones, the scoring function itself just produces a plausibility value. It doesn't guarantee all negative triples score higher or lower unconditionally.

---

**QUESTION 4:** [1 mark]

One key difference between the differentiable KG approach and the semantic interpretation approach to KGQA is:

- a. Differentiable KG approaches are fully rule-based, while semantic interpretation is purely neural.
- b. Differentiable KG approaches do not require any graph embeddings, relying instead on explicit logical forms.
- c. Semantic interpretation is more transparent or interpretable, whereas differentiable KG is end-to-end trainable but less interpretable.
- d. Both approaches use logical forms; the primary difference is the type of question they can answer.

**Correct Answer:** c

**Explanation:**

- **Semantic interpretation is more interpretable, differentiable KG is more end-to-end (c):**
    - Semantic interpretation methods typically rely on building an explicit logical form of the question, which is transparent and can be easily explained.
    - Differentiable KGQA uses neural embeddings and end-to-end backpropagation over the graph, making it powerful but less human-interpretable.
  - **Why not the others?**
    - (a) Differentiable KG approaches are *not* fully rule-based — in fact, they're more neural and less rule-based.
    - (b) Differentiable approaches definitely *do* use graph embeddings.
    - (d) While both can handle complex questions, the key difference in c is the interpretability vs. end-to-end trainability.
- 

**QUESTION 5:** [1 mark]

Considering the differentiable KG approach, which elements are typically learned jointly when training an end-to-end KGQA model?

- a. The textual question representation (e.g., BERT embeddings)
- b. The graph structure encoding (e.g., GCN or transformer-based graph embeddings)
- c. Predefined logical forms to ensure interpretability
- d. The final answer selection mechanism that identifies which node(s) in the graph satisfy the question

**Correct Answer:** a, b, d

**Explanation:**

- **Textual question representation (a):** The system learns how to embed the input question (often with a neural model like BERT).
- **Graph structure encoding (b):** It also learns how to encode nodes and relations in the knowledge graph (using a graph neural network, attention, etc.).
- **Final answer selection (d):** Finally, the model learns how to map from the question and graph embeddings to the correct node(s) in the KG.
- **Why not (c)?** Predefined logical forms are more typical of semantic-parsing-based KGQA, not differentiable KGQA approaches. Differentiable KGQA is usually end-to-end and does not require manually crafted logical forms.

#### **QUESTION 6:** [1 mark]

Uniform negative sampling can have high variance and may require large number of samples. Why is that the case?

- a. Because the margin-based loss cannot converge without big mini-batches.
- b. Because randomly picking negative entities does not guarantee close or challenging negatives, causing unstable training estimates.
- c. Because negative sampling must ensure every possible negative triple is covered.
- d. Because the number of relations in the KG is too large for small number of samples.

**Correct Answer:** b

**Explanation:**

- **High variance arises when negatives are not challenging (b):** If negative examples are chosen completely at random, many will be too easy for the model to distinguish, providing limited learning signal. The model sees less “borderline” cases, so estimates of how well the model can separate real vs. fake facts fluctuate significantly.
- **Why not the others?**
  - (a) Margin-based losses can converge with or without large mini-batches if sampling is done carefully.

(c) We don't need to cover every possible negative triple, just enough meaningful ones for training.

(d) The number of relations in a KG might be large, but that alone doesn't necessarily drive the variance issue.

---

#### QUESTION 7: [1 mark]

In testing embedding and score quality for KG completion, *mean rank* and *hits@K* are typical metrics. What does *hits@K* specifically measure in this context?

- a. The percentage of queries for which the correct answer appears in the top-K of the ranked list.
- b. The reciprocal of the rank of the correct answer.
- c. The probability of the correct answer appearing as the highest scored candidate.
- d. The margin of the correct triple score relative to all negative triples.

**Correct Answer: a**

**Explanation:**

- **Hits@K = The percentage of queries for which the correct entity (or triple) is in the top-K predictions (a).** This means if the correct answer is within the first K results in the ranking, we call it a "hit." We then compute how many queries achieve this, divided by the total.

- **Why not b, c, or d?**

(b) That is more like Mean Reciprocal Rank (MRR), not hits@K.

(c) If  $K=1$ , that might coincide with hits@1, but the metric hits@K is about the top-K in general, not exclusively the top candidate.

(d) That describes a margin-based idea, not hits@K.

---

#### QUESTION 8: [1 mark]

In the TransE model, the scoring function for a triple  $(s, r, o)$  is typically defined as

$$f(s, r, o) = \|e_s + e_r - e_o\|$$

where  $e_s$ ,  $e_r$ ,  $e_o$  are embeddings of the subject, relation, and object, respectively. Which statement best explains what a low value of  $f(s, r, o)$  indicates in this context?

- a. That  $(s, r, o)$  is an invalid triple according to the learned embeddings.

- b. That  $e_s$  and  $e_o$  must be orthogonal.
- c. That the relation embedding  $e_r$  is zero.
- d. That  $(s, r, o)$  has a high likelihood of being a true fact in the knowledge graph.

**Correct Answer:** d

**Explanation:**

- **A low distance = a high plausibility (d):** In TransE, the model is trained such that  $s + r \approx o$  for a valid triple. If the norm  $\|s + r - o\|$  is small, it means the subject, relation, and object embeddings line up well, indicating that triple is likely true.
- **Why not a, b, c?**
  - (a) A high value would correspond to an invalid triple.
  - (b) Orthogonality is not directly indicated by a small distance in TransE.
  - (c) Zero relation embedding is not required for plausibility; the relation embedding can be non-zero and still yield a low distance.

**QUESTION 9:** [1 mark]

In RotatE, if a relation  $r$  is intended to be symmetric, how would that typically manifest in the complex plane?

- a. The relation embedding  $e_r$  must always equal zero.
- b. The angle of  $e_r$  must be  $\frac{\pi}{2}$ .
- c. The relation embedding  $e_r$  is its own inverse (i.e., a  $180^\circ$  rotation when squared).
- d. The magnitude of  $e_r$  must be greater than 1.

**Correct Answer:** c

**Explanation:**

- **Relation embedding is its own inverse (c):** In RotatE, each relation is modeled as a rotation in the complex plane. For a relation to be symmetric, applying that relation twice would yield the original entity, so  $r^2 = 1$ . A  $180^\circ$  rotation (i.e.,  $\pi$  radians) is its own inverse because rotating twice by  $180^\circ$  brings you back to the same orientation.
- **Why not a, b, or d?**
  - (a) A zero embedding is not characteristic of symmetry in RotatE.
  - (b) An angle of  $\pi$  radians alone indicates an inverse relation, but the question specifically references the notion of the embedding being its own inverse, which implies  $r^2 = 1$ .

(d) The magnitude constraint (often magnitude = 1 in RotatE) is not specifically about symmetry.

---

**QUESTION 10:** [1 mark]

Which main advantage do rotation-based models (like RotatE) have over translation-based ones (like TransE) when it comes to complex multi-relational patterns in a KG?

- a. Rotation-based models cannot model any symmetry or inverse patterns, so they are simpler.
- b. Rotation-based models handle a broader set of relation properties (symmetry, anti-symmetry, inverses, composition) more naturally.
- c. Rotation-based models have no hyperparameters to tune, unlike TransE.
- d. Rotation-based models are guaranteed to yield perfect link prediction.

**Correct Answer:** b

**Explanation:**

- **Rotation-based models can capture more complex relational properties (b):** By representing relations as rotations in the complex plane, RotatE naturally supports symmetric relations ( $r^2 = 1$ ) anti-symmetric relations, inverses (rotations in the opposite direction), and composition (cumulative rotations).
  - **Why not a, c, or d?**
    - (a) In fact, they can model symmetry, inverses, etc.
    - (c) They do have hyperparameters (e.g., embedding dimension, learning rate); it is not hyperparameter-free.
    - (d) No model is guaranteed to be perfect for all link prediction tasks.
-