Introduction to Large Language Models
Assignment- 11

Number of questions: 10 **Total mark: 10 X 1 = 10**

QUESTION 1: [1 mark]

What is the main modification that SimplE makes to DistMult-like models to handle asymmetric relations?

a. Replacing entity embeddings with random fixed vectors
b. Introducing separate entity embeddings for subject and object roles, along with inverse relations
c. Restricting the rank of the relation tensor to 1
d. Using negative sampling for half of the triple set

Correct Answer: b

Explanation:

- **SimplE** extends the DistMult approach by using **two different embeddings** for each entity: one as a subject (head) and one as an object (tail). It also learns inverse relation embeddings to handle asymmetry.

- DistMult alone struggles with asymmetric relations because its scoring function $score(s, r, o)$ is symmetrical in s and o. SimplE partially addresses this by modelling subject/object roles distinctly and considering inverse relations.

QUESTION 2: [1 mark]

Which statements correctly characterize the basic DistMult approach for knowledge graph completion?

a. Each relation r is parameterized by a full D×D matrix that can capture asymmetric relations.
b. The relation embedding is a diagonal matrix, leading to a multiplicative interaction of entity embeddings.
c. DistMult struggles with non-symmetric relations because score(s, r, o) = $a_s^T M_r a_o$ is inherently symmetric in s and o.
d. DistMult's performance is typically tested only on fully symmetric KGs.

Correct Answer: b, c

Explanation:

- **DistMult** stands for "*Diagonal* bilinear model", meaning each relation r has a diagonal embedding M_r. This is effectively a vector multiplied element-wise (since a diagonal matrix just scales each coordinate).

- This diagonal constraint leads to symmetric scoring: swapping s and o yields the same value because of the multiplication pattern.

- Therefore:

    - (b) is **true**: the relation embedding is diagonal.

    - (c) is **true**: it struggles with asymmetric relations because $a_s^T(diag(r))a_o$ is symmetric in a_s and a_o.

- (a) is **false** because DistMult does not use a full D×D matrix; it specifically uses a diagonal matrix.

- (d) is also **false**: DistMult can be tested on general KGs (not only fully symmetric ones), though it does poorly with strongly asymmetric relations.

QUESTION 3: [1 mark]

Which statements about the ComplEx extension of DistMult are true?

a. It uses complex-valued embeddings to better capture asymmetric or anti-symmetric relations.
b. It replaces the multiplication in DistMult with element-wise addition of real-valued vectors.
c. For a perfectly symmetric relation, one could set the imaginary part of the relation embedding to zero.
d. ComplEx requires each entity vector to be unit norm in the complex plane.

Correct Answer: a, c

Explanation:

ComplEx extends DistMult to the complex domain, allowing it to handle asymmetric relations by using conjugate-based interactions. A typical ComplEx scoring function is: $Re(< a_s,\ r,\ \overline{a_o} >)$, where a_s, r, a_o are complex vectors, and $\overline{a_o}$ is the complex conjugate of a_o .

- (a) **True**: The use of complex embeddings allows capturing asymmetry via imaginary components and conjugation.
- (c) **True:** If the imaginary part is zero, the relation effectively behaves like DistMult in the real domain, which is symmetric.

- (b) is **false**: It still uses multiplicative interactions, not element-wise addition.

- (d) is **not necessarily required**; while some implementations might normalize embeddings, ComplEx does not strictly require each entity to have unit norm. It's a common training choice but not an inherent necessity.

QUESTION 4: [1 mark]

Which best describes the main advantage of using a factorized representation (e.g., DistMult, ComplEx) for large KGs?

a. It enforces that every relation in the KG be perfectly symmetric.
b. It ensures each entity is stored as a one-hot vector, simplifying nearest-neighbour queries.
c. It collapses the entire KG into a single scalar value.
d. It significantly reduces parameters and enables generalization to unseen triples by capturing low-rank structure.

Correct Answer: d

Explanation:

- Factorized models (DistMult, ComplEx, etc.) leverage low-rank embeddings for entities and relations, drastically reducing the parameters needed compared to storing a full adjacency or full 3D tensor for the entire knowledge graph.

- This **low-rank structure** also helps the model learn patterns and generalize to unseen triples (i.e., it can guess new edges that weren't explicitly in training).

QUESTION 5: [1 mark]

Which statement best describes the *reshaping* of a 3D KG tensor $X \in R^{|E|\times|R|\times|E|}$ into a matrix factorization problem?

a. One axis remains for subject, one axis remains for object, and relations are combined into a single expanded axis.
b. The subject dimension is repeated to match the relation dimension, resulting in a 2D matrix.
c. Each subject–relation pair is collapsed into a single dimension, while objects remain as separate entries.
d. The entire KG is vectorized into a 1D array and then factorized with an SVD approach.

Correct Answer: c

Explanation:

- A 3D KG tensor $X \in R^{|E|\times|R|\times|E|}$ typically has subject-relation-object axes.

- One common approach is to “unfold” or “reshape” the tensor along one axis to produce a matrix. E.g., you can merge the subject and relation dimensions into a single dimension (subject-relation pairs) and keep the object axis separate, yielding a 2D matrix factorization problem.

- This approach is used in some methods to apply standard matrix factorization to the KG data.

QUESTION 6: [1 mark]

Which key property of hierarchical relationships (e.g. is-a, transitivity) motivates the exploration of specialized embedding methods over standard Euclidean KG embeddings?

a. Symmetry in the relation (A, is-a, B) implying (B, is-a, A)
b. Frequent presence of cycles in hierarchical graphs
c. Transitivity in the form (camel, is-a, mammal) and (mammal, is-a, animal) $\Longrightarrow$ (camel,is-a,animal)
d. The high dimensionality of the entity embeddings

Correct Answer: c

Explanation:

- **Hierarchical** “is-a” relations typically exhibit **transitivity**: if A is-a B, and B is-a C, then A is-a C. Standard Euclidean embeddings (like DistMult, ComplEx) do not directly enforce or encourage transitivity.
- Specialized approaches (e.g., hyperbolic embeddings, order embeddings) do a better job of **naturally capturing** such transitive relations.

QUESTION 7: [1 mark]

Which of the following statements correctly describe hyperbolic (Poincare) embeddings for hierarchical data?

a. They map nodes onto a disk (or ball) such that large branching factors can be represented with lower distortion than in Euclidean space.
b. Distance grows slowly near the centre and becomes infinite near the boundary, making it naturally suited for tree-like structures.
c. They require each node to be embedded on the surface of the Poincare disk of radius 1.
d. They can achieve arbitrarily low distortion embeddings for trees with the same dimension as Euclidean space.

Correct Answers: a, b

Explanation:

- **Hyperbolic (Poincaré) embeddings** place points in a disk (2D) or ball (higher-D) where distances near the boundary can become very large, effectively allowing the model to represent tree-like expansions with less distortion than Euclidean.

- Specifically:

    - (a) **True:** Large branching structures are more compactly represented.

    - (b) **True:** Distances blow up near the boundary, which helps model hierarchical "depth."

- (c) is **not** strictly correct: points reside within the disk, not necessarily on the boundary (though some parameterizations might keep norms < 1).

- (d) is also not entirely correct in claiming "arbitrarily low distortion in the *same* dimension." Hyperbolic embeddings do better than Euclidean in capturing tree metrics, but **not** everything is **guaranteed** "arbitrarily low" in the same dimension. They do reduce distortion significantly compared to Euclidean though.

QUESTION 8: [1 mark]

Why might a partial-order-based approach (like order embeddings) be beneficial for modelling 'is-a' relationships compared to purely distance-based approaches?

a. They explicitly encode the ancestor–descendant relation as a coordinate-wise inequality or containment.
b. They can represent negative correlations (i.e., sibling vs. ancestor) more easily than distance metrics.
c. They inherently guarantee transitive closure of the hierarchy in the learned embedding space.
d. They do not rely on pairwise distances but use a notion of coordinate-wise ordering or interval containment.

Correct Answer: a, d

Explanation:

- **Order embeddings** interpret the "is-a" relationship as a partial order, typically using **coordinate-wise constraints** (for instance, $X \leq Y$ if $x_i \leq y_i \; \forall i$). This can capture ancestor-descendant with a straightforward geometric interpretation (e.g., intervals, boxes, or cones).
- They do not rely solely on a distance measure; they rely on the geometry of "containment" or inequality in each coordinate.

- Hence, (a) and (d) are correct.

- (b) is not necessarily guaranteed or specifically a reason they do it more easily; negative correlation is a different concept.

- (c) They do not *automatically* guarantee full transitive closure in the sense that it's not a built-in "guarantee." The embeddings typically encourage partial ordering, but some additional constraints or training objectives are needed.

QUESTION 9: [1 mark]

Which statement about box embeddings in hierarchical modelling is most accurate?

a. Each entity or type is assigned a single real-valued vector, ignoring bounding volumes.
b. Containment $I_x \subseteq I_y$ all dimensions encodes $x \prec y$.
c. They rely on spherical distances around a central node to measure tree depth.
d. They cannot be used to represent set intersections or partial overlap.

Correct Answer: b

Explanation:

- **Box Embeddings** assign each concept or entity a "box" in some coordinate space, typically described by two vectors $l \leq u$ (lower and upper corners).

- If box X is contained within box Y on every coordinate dimension (i.e., $l_X \geq l_Y$ and $u_X \leq u_Y$), that corresponds to an "is-a" or subset relationship.

- This approach can also represent partial overlaps (intersecting boxes) and so on, so (d) is incorrect.

- (c) describes a more spherical or Euclidean method, not boxes.

- (a) is the opposite of the box approach (which does *not* ignore bounding volumes but explicitly uses them).

QUESTION 10: [1 mark]

What is a key challenge with axis-aligned open-cone (order) embeddings for hierarchical KG data?

a. They enforce that all sibling categories have identical cone apices, which causes overlap.
b. They require symmetrical relationships for all edges.

c. They do not allow partial orders to be extended to total orders.
d. The volume (measure) of cones is the same regardless of how “broad” or “narrow” the cone is, making sub-categories indistinguishable by volume.

Correct Answer: d

Explanation:

- In **axis-aligned cone embeddings**, each entity is associated with an open cone in a coordinate space. The cones can be wide or narrow, but the measure of the open cone does not reflect “how big” or “small” the category is, which makes it difficult to distinguish sub-categories based on “volume.”

- This is a known limitation: the geometry does not allow easily using volume-based notions to reflect hierarchical or subset relationships beyond the angular direction.
