Sample Exam Analysis — AI Engineering, WS 2025/26

Sources: Example_AI_Engineering_WS20252026_Exam.pdf and Example_AI_Engineering_WS20252026_ (Dr.-Ing. Johannes Abel, IfN, TU Braunschweig). The released sample contains only three example questions — one per exercise category — explicitly labelled "(not exam)". Use them to learn the format and style; full content of the real exam is unknown.

---

1. Exam logistics

Item	Value
Date (sample header)	9 February 2026
Duration	120 minutes
Total points	50
Allowed materials	non-programmable calculator only
Language for answers	English
Number of exercises	3 (Fundamentals, Analysis, Application)

Instructions reproduced from the cover page

- Answer all questions in English.
- Round to 2 decimals.
- Use lecture's technical terms (full names, no abbreviations).
- Do NOT provide multiple answers when only one is asked.
- "Short, precise answers save time and are generally all you need."
- Multiple-choice: exactly one correct option per question. Marking more than one ⇒ 0 points (no partial credit).

  Engineering implication. Roughly 2.4 minutes per point. Plan your time on each exercise before writing.

---

2. The three sample questions and their solutions

Sample 1 — Fundamentals (Multiple Choice, 1 P)

Q. Which statement best describes cosine similarity for embeddings?

(a) It compares vector directions and is scale-invariant.
(b) It compares vector magnitudes only.
(c) It requires vectors to be one-hot.
(d) It is identical to Euclidean distance.

Official answer: (a)

Maps to: Chapter 2.4 (Embeddings) and Chapter 4.2 (Embedding Models).

Why other options are wrong: - (b) Cosine ignores magnitude, looking only at angle. - (c) One-hot vectors would always have cosine $\in \{0, 1\}$; cosine works on any real vectors. - (d) Euclidean uses $\|u-v\|$; cosine uses $u \cdot v/(\|u\|\|v\|)$. They are different functions.

Why this is the model fundamentals question: - Tests definitions, not calculation. - Maps directly to a single slide. - Distractors are plausible-sounding but factually wrong — typical exam style.

Sample 2 — Analysis (Free Text, 3 P)

Q. Explain why removing duplicate documents from a pretraining corpus can improve generalization, even though it reduces the total amount of training data.

Official answer (paraphrased to avoid copy-paste): Duplicates make the model encounter identical patterns repeatedly, biasing it toward memorisation rather than abstraction. Deduplication raises diversity per training step, improves effective concept coverage, and reduces overfitting / verbatim leakage — improving generalization despite the smaller token count.

Maps to: Chapter 2.1 (Training Data) and Chapter 2.8 (Pretraining), with an indirect link to Chapter 7.2 (copyright/PII risks of memorisation).

Why this question style appears: - Tests cause-and-effect understanding, not memorisation. - Requires keywords: deduplication, memorisation, overfitting, generalization, diversity, abstraction — all from the lecture. - 3 points usually corresponds to 3 distinct points in the answer (one bullet $\approx$ one point). Train yourself to write three short sentences.

Sample 3 — Application (Mini-Case, 5 P)

Q. You observe that an LLM-based agent repeatedly calls tools without making progress and sometimes enters long reasoning loops. Propose one concrete mitigation strategy and justify it.

Official answer (paraphrased): Add explicit stopping criteria and action budgets to the agent control loop (e.g., max iterations / max tokens / max wall-clock). This prevents infinite loops and forces termination or escalation to a human. Trade-off: may stop some legitimate long-running tasks.

Maps to: Chapter 5.1 (Agent Loop, Failure Modes, Budgets and Stopping).

What "5 points" buys you: - 1–2 points for naming a concrete technique (budget / max-steps). - 1–2 points for justifying it (mechanism: cuts the cycle). - 1 point for noting a limitation / trade-off (legitimate tasks may also stop).

> Pattern: Application questions reward concrete mechanism + justification + trade-off awareness. Always end with a "but…" sentence.

---

3. What the structure tells us about the real exam

3.1 Three-exercise blueprint

Exercise	Style	Pts each	Total share	Maps to
Fundamentals	Multiple choice + 1-line definitions	1–2	~15–20 P	broad coverage of all chapters
Analysis	Free-text "explain why" / derivations	3–4	~15–20 P	depth on key topics

Exercise	Style	Pts each	Total share	Maps to
Application	Mini-case scenarios	5–8	~10–15 P	end-to-end reasoning, often Ch 4–5

3.2 Question style cues

- "Which statement best describes …" → MC, one answer.
- "Explain why …" → 3-point free text; aim for 3 reasons.
- "Propose one concrete strategy and justify …" → application; mechanism + reason + trade-off.
- "Round to 2 decimals" → numerical computation expected.
- "Use the technical terms from the lecture" → the lecture's own exact phrasing is what graders look for.

3.3 Topic coverage in the 3-question sample

Chapter	Hit
Ch 1 Intro	–
Ch 2 Foundation Models	✓ (sample 1: embeddings; sample 2: training data)
Ch 3 Prompting	–
Ch 4 RAG	✓ (sample 1: cosine for embeddings is also a RAG concept)
Ch 5 Agents	✓ (sample 3)
Ch 6 Fine-tuning	–
Ch 7 Legal & Ethical	–

The released sample emphasises Ch 2 + Ch 5, but with only 3 questions the sampling is too thin to draw strong conclusions. Treat all 7 chapters as fair game.

---

4. Per-chapter mapping of sample questions

Sample #	Title	Primary chapter	Secondary chapters
1	Cosine similarity (MC)	Ch 2.4 Embeddings	Ch 4.2 Embedding Models
2	Deduplication & generalization	Ch 2.1 Training Data	Ch 2.8 Pretraining; Ch 7.2 (memorisation/copyright)
3	Agent failure / mitigation	Ch 5.1 The Agent Loop	Ch 5.2 Architectures, Ch 5.7 Safety

There are no unmatched sample questions — all three map cleanly onto chapters covered in the lecture notes.

---

5. Likely exam-topic priority list

Derived from (a) lecture-slide emphasis, (b) the sample's coverage, (c) recurring patterns in TUBS exams of this type.

Tier A — High probability (expect at least one question)

- Cosine similarity / embeddings (Ch 2.4 / 4.2). MC + small calculation.
- Self-attention computation (Ch 2.5). Numeric.
- BPE tokenization with a small example (Ch 2.2).
- Sampling strategies — temperature / top-k / top-p (Ch 2.7). Compute distribution.
- SFT vs RLHF vs DPO comparison (Ch 2.10). Free text.
- End-to-end RAG pipeline (Ch 4.1) and BM25 vs dense vs hybrid (Ch 4.4).
- Agent definition / loop / failure modes (Ch 5.0–5.1). Sample 3 confirms.
- LoRA math: $r(d+k)$ trainable parameters and $\Delta W = (\alpha/r)BA$ (Ch 6.3).

Tier B — Medium probability

- Bigram language model (Ch 1.2).
- Chinchilla scaling law (Ch 2.11).
- Cross-encoder reranker (Ch 4.5).
- Prompt patterns / function calling (Ch 3.6, 3.7).
- ReAct vs Planner-Executor (Ch 5.2).
- EU AI Act risk pyramid (Ch 7.2).
- Disparate impact / 80% rule (Ch 7.3).
- Catastrophic forgetting (Ch 6.1).

Tier C — Lower probability (still worth knowing)

- ASR Bayes formulation (Ch 1.2).
- RoPE / RMSNorm / SwiGLU (Ch 2.6).
- Distributed training (Ch 2.8).
- HyDE / advanced RAG (Ch 4.8).
- Hierarchical / multi-agent (Ch 5.2).
- Adapter vs LoRA latency (Ch 6.3).
- Drift detection PSI (Ch 7).

---

6. Weak-area diagnosis (suggested practice)

Items where many beginners stumble in this exam style:

1. Cosine vs Euclidean distance — they reward opposite things; practise computing both on a 2-D toy example until automatic.
2. Per-token cost of MC mistakes — multi-marking ⇒ 0 points; always erase before final mark.
3. 3-point answer structure — write exactly three short sentences for "Explain why" questions; one cause, one mechanism, one consequence.
4. Stating the trade-off — application questions almost always reward a "but…" sentence.
5. Numerical attention — practise hand computation of $\text{softmax}(QK^T/\sqrt{d_k})V$ for $n = 2$, $d_k = 2$, with row-sum sanity check.
6. LoRA parameter count — drill the formula $r(d+k)$ and ratio against $d \cdot k$.

7. Exact lecture terminology — the exam is in English; always use the precise technical terms as named in the lecture slides.

---

7. New harder questions inspired by the sample style

Following the same three-tier format. Targeted at practising the type of cognition the exam rewards.

7.1 Fundamentals (MC, 1 P each)

N1. Which statement best describes the role of the causal mask in a decoder-only transformer? (a) It scales the attention scores by $\sqrt{d_k}$. (b) It zeros out attention to future positions during training and inference. (c) It removes padding tokens from the loss. (d) It clips gradients during back-propagation.

Answer: (b).

N2. Which retrieval strategy is most appropriate when queries contain rare exact-match terms such as product codes? (a) Dense vector retrieval only. (b) BM25 (sparse) retrieval, possibly fused with dense via Reciprocal Rank Fusion. (c) Cross-encoder reranking with no retrieval. (d) HyDE.

Answer: (b).

N3. Which property is guaranteed at LoRA initialisation when $B = 0$? (a) Loss is exactly zero. (b) The model output equals the frozen base model's output. (c) The optimizer is in steady state. (d) The trainable parameter count equals the base model count.

Answer: (b).

N4. Under the EU AI Act, which of the following is classified as unacceptable risk (banned)? (a) Spam filter using AI. (b) Social scoring of citizens by public authorities. (c) Customer-support chatbot. (d) AI in video-game NPCs.

Answer: (b).

N5. Which sampling configuration gives deterministic generation? (a) Temperature = 1.0, top-p = 0.9. (b) Temperature = 0.0 (greedy decoding). (c) Temperature = 2.0, top-k = 40. (d) Random sampling without temperature.

Answer: (b).

7.2 Analysis (Free text, 3 P each)

N6. Explain why multi-head attention generally outperforms single-head attention with the same total dimensionality d. Suggested 3-bullet answer: (1) different heads can specialize on different relations (syntax, coreference, position); (2) parallel computation enriches the representation more than a single wide head; (3) empirically validated on standard benchmarks. Trade-off: communication / projection overhead.

N7. Explain why Direct Preference Optimization (DPO) is more stable than PPO-based RLHF, despite using the same human-preference data. Suggested 3-bullet answer: (1) DPO uses a closed-form supervised loss; (2) it does not need a separate reward model or critic; (3) optimisation is convex in the log-policy for fixed reference, avoiding RL instability. Trade-off: less expressive when preferences require multi-step credit assignment.

N8. Explain the lost-in-the-middle phenomenon and propose two prompt-design countermeasures. Suggested answer: Transformer attention is bowl-shaped — strong at start and end of the context, weak in the middle (Liu et al. 2023). Countermeasures: (1) place critical instructions at top and bottom; (2) compress / summarise older conversation turns to keep the window short.

7.3 Application (Mini-case, 5 P each)

N9. A start-up wants to deploy a customer-support chatbot. The product team observes that the assistant gives outdated answers about company policies that change weekly. Propose one concrete architectural change and justify it. Mention one trade-off. Suggested answer: Add a Retrieval-Augmented Generation (RAG) layer using the company's policy database as the retrieval corpus; the LLM is conditioned on freshly retrieved chunks at inference time. This keeps knowledge current without retraining. Trade-off: retrieval latency adds 100–500 ms and increases prompt-token cost.

N10. A medical Q&A LLM frequently outputs fluent but incorrect dosage information. Identify the failure mode, name one specific evaluation metric that would catch it, and propose one mitigation. Suggested answer: The failure is hallucination. The metric faithfulness (e.g. RAGAS, NLI-based entailment checking the answer against the retrieved context) catches it. Mitigation: ground the model in a curated medical knowledge base via RAG and reject answers below a faithfulness threshold; for high-stakes queries, route to a human pharmacist. Trade-off: latency and operational cost.

N11. Your team has 50 000 training examples in a niche legal domain and a 70 B-parameter base model. Compare full fine-tuning vs LoRA in this scenario. Recommend one and justify. Suggested answer: Full FT $\approx$ 70 B trainable params $\rightarrow$ ~140 GB optimizer state + base, infeasible on common hardware; risk of catastrophic forgetting of general competence. LoRA with $r = 16$ on Q/K/V/O matrices $\approx$ a few hundred million trainable params, fits on a single 80 GB GPU; base remains frozen $\rightarrow$ general competence preserved; LoRA file ships in tens of MB. Recommend LoRA. Trade-off: small accuracy gap on very large datasets vs full FT.

N12. A team integrates a tool-using LLM agent into their product. After deployment, the agent occasionally executes destructive actions ("delete user account") triggered by a malicious instruction inside a fetched web page. Identify the threat, name two mitigations, and discuss residual risk. Suggested answer: The threat is prompt injection through tool outputs. Mitigations: (1) limit tool privileges (allow-listed read-only tools by default; destructive actions require explicit human approval); (2) sanitise tool outputs (strip / redact instruction-like patterns) before re-feeding them to the LLM. Residual risk: sanitisation is a defence-in-depth layer, not a guarantee — sophisticated attacks can evade regex/classifier filters. Hence principle of least privilege is the durable defence.

---

8. Strategy notes specific to this exam

1. Read the full instructions before exercise 1 — the language rule (English), the round-to-2-decimals rule, and the no-multiple-MC-marks rule are all worth points lost otherwise.
2. Allocate by points, not by question count. Spend 30 % of time on Application even if there is only one question worth 15 P.
3. For MC, eliminate two distractors first; pick the most precise of the remaining two. Do not over-think.
4. For Analysis, write the 3-point answer as three sentences with explicit causal connectors ("because…", "this means…", "therefore…").
5. For Application, structure: (i) name the concept, (ii) describe the mechanism, (iii) justify, (iv) note one trade-off.

6. Write out full technical names — e.g. “Reciprocal Rank Fusion”, not “RRF” (the rules forbid abbreviations).
7. Calculator hygiene: write each step on paper; never round mid-computation.

---

9. Final mock-exam time plan (120 min for 50 P)

Phase	Time	Activity
0–5	5 min	Read whole exam, allocate per-exercise budget
5–35	30 min	Fundamentals (MC + short defs)
35–80	45 min	Analysis questions
80–115	35 min	Application mini-cases
115–120	5 min	Re-check MC erasures, verify decimal rounding

---

10. Cross-references to the existing notes

To go deeper	See
Cosine similarity definition + worked example	Chapter_02_Foundation_Models.md §2.4, Chapter_04_RAG.md §4.2
Deduplication / training data quality	Chapter_02_Foundation_Models.md §2.1
Agent budgets, stopping criteria, failure modes	Chapter_05_Agents.md §5.1
LoRA math	Chapter_06_Finetuning.md §6.3
EU AI Act tiers	Chapter_07_Legal_and_Ethical.md §7.2
Lost-in-the-middle	Chapter_03_Prompt_Engineering.md §3.5
All sample-style new questions	this file §7

End of Sample Exam Analysis.