Ex1 Answers – Tomer Yaacoby

<u>First Part:</u>

Q.1

QA-SRL Bank 2 – Question-Answer Driven Semantic Role Labeling
QA-SRL frames each verbal predicate in a sentence as a crowd-written wh-question whose answer is the argument span (e.g., *"Who **built** something?"* → *"the engineers"*). Correctly producing or interpreting these question–answer pairs requires the model to uncover predicate–argument structure—*who did what to whom, when and where*—an intrinsic facet of sentence-level semantic understanding.

QAMR (Question-Answer Meaning Representation)
QAMR represents the full meaning of a sentence as a set of free-form QA pairs that cover explicit and implicit predicate–argument relations. Because the questions abstract away from any external task and directly probe the underlying propositional content (including implicit roles), answering them demonstrates deep, intrinsic comprehension of sentence semantics rather than task-specific skills.

Quoref – Coreference-focused Reading Comprehension
Quoref's questions can only be answered by tracking entity mentions that co-refer across clauses and sentences (e.g., linking "Dr. Curie" with "she"). Resolving these references is fundamental to discourse-level language understanding and is independent of any downstream application, making the dataset a direct test of an intrinsic property: coreference resolution.

Q.2.

a.

Chain-of-Thought (CoT) prompting.
<u>Brief:</u> The idea is to add a short instruction such as "Think step-by-step" so the language model explicitly prints its reasoning chain before giving a final answer.

<u>Advantage:</u> This simple prompt tweak often boosts accuracy on arithmetic, logical and scientific tasks because the model is encouraged to "show its work."

<u>Bottlenecks:</u> The price you pay is a much longer output sequence, which means more forward passes through the network and higher memory pressure inside the context window—GPU compute time and VRAM are the limiting factors.

<u>Parallelized:</u> Parallelism is limited to batching several independent questions together; within one question the generation remains sequential.

Self-Consistency (SC).
<u>Brief:</u> Instead of trusting a single CoT trace, you sample *K* independent chains at a modest temperature and let the majority answer (or the most confident one) win.

<u>Advantage:</u> The key benefit is robustness: random reasoning errors tend to cancel out, so accuracy can rival that of much larger models.

Bottlenecks: The downside is that you now perform *K* times more generation work, so compute and memory scale linearly with *K*.

Parallelized: Fortunately every chain is independent, so you can parallelize perfectly—e.g., batch the *K* continuations on one big-memory GPU or scatter them across devices.

Tree-of-Thought (ToT) and related search variants.
Brief: Here the model incrementally expands a tree of partial reasoning paths, evaluates or prunes branches using a heuristic or verifier, and continues exploring only the most promising nodes (akin to breadth-first, depth-first or Monte-Carlo tree search).

Advantage: This structured exploration achieves state-of-the-art results on long-horizon reasoning problems, because the search can backtrack and recover from mis-steps.

Bottlenecks: Yet the method generates huge numbers of intermediate tokens and must keep the tree in host RAM, so both VRAM and CPU memory/logic become bottlenecks.

Parallelized: Some parallelism is possible by batching all children at a given depth, but overall the search is inherently sequential because later expansions depend on earlier scores.

Best-of-N (parallel sampling with a verifier).
Brief: You fire off *N* relatively short candidate answers in parallel and use a lightweight verifier—often a smaller reward model or heuristic—to choose the best one.

Advantage: This saves tokens compared with long CoT traces and can give large quality gains if the verifier is well tuned.

Bottlenecks: Compute scales with *N* (for generation) plus a small extra cost for the verifier's passes.

Parallelized: Both stages parallelize cleanly: candidates are generated in one batch and the verifier can score them in another.

ReAct and tool-augmented deliberation.
Brief: In ReAct, the model alternates natural-language thought steps with calls to external tools such as code interpreters, calculators or web search.

Advantage: Off-loading sub-tasks to specialized tools greatly improves factual accuracy and numerical precision.

Bottlenecks: Latency shifts from pure GPU compute to the slower I/O involved in invoking those tools; orchestration logic usually runs on the CPU.

Parallelized: You can sometimes overlap independent tool calls (e.g., multiple API requests), but the dialogue loop itself is sequential: LLM → tool → LLM and so on.

b.

When only one large-memory GPU is available, **Self-Consistency with Chain-of-Thought (SC-CoT)** offers the most effective compromise between reasoning quality and computational cost. Sampling *K* independent reasoning chains and selecting the majority answer markedly improves accuracy over a single CoT trace because random reasoning errors tend to cancel out. At the same time, SC-CoT is considerably simpler and less resource-intensive than Tree-of-Thought search. All *K* chains can be generated concurrently in a single batched forward pass, maximising GPU utilisation without extra control logic. The sole tuning parameter, *K*, provides a straightforward way to trade execution time for solution quality.

Second Part:

Q.2.

a. No.

The best eval accuracy achieved by the configuration:

epoch_num: 5, lr: 0.0001, batch_size: 32, eval_acc: 0.8603, got a test accuracy = 0.8209.

But the configuration that achieved second on the evaluation:

epoch_num: 5, lr: 2e-05, batch_size: 32, eval_acc: 0.8480, got a test accuracy = 0.8290.

b. The worst system is heavily biased toward the entailment class, and whenever the gold label is not entitlement it usually answers “1”.

Looking at the next example:

S1: A tropical storm rapidly developed in the Gulf of Mexico Sunday and was expected to hit somewhere along the Texas or Louisiana coasts by Monday night .

S2: A tropical storm rapidly developed in the Gulf of Mexico on Sunday and could have hurricane-force winds when it hits land somewhere along the Louisiana coast Monday night.

The better model succeed on this while the worst does not.

The two sentences express essentially the same news. We can guess that the strong model did looked past the extra clause (“could have hurricane-force winds”) and recognized it doesn’t contradict the core fact. And also treated the location difference (“Texas or Louisiana” vs “Louisiana”) so the meaning is still compatible.

While the worst model heavily rely on surface overlap. The fact that “Texas” appears in S1 but not S2 drops the lexical similarity score. And the new clause about hurricane-force winds introduces tokens with no counterpart in S1.

So we can conclude the the higher-performing configuration captures <u>semantic</u> equivalence and ignores non-contradictory embellishments, whereas the lower-performing one is thrown off by <u>word-level</u> asymmetries like the presence/absence of “Texas” and the added wind-strength detail.