Ex1 Answers – Tomer Yaacoby

URL

https://github.com/tomer4185/ANLP-Ex1

First Part:

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 prompting.
Append an explicit instruction such as *"Think step-by-step"* so the model prints a reasoning chain before the final answer . The upside is a large jump in accuracy on arithmetic, logic and science benchmarks "for free"—no extra training, just a different prompt. The cost is that each query now produces many more tokens, so GPU-side compute and context-window memory become the tightest bottlenecks. Parallelism is limited: within one question generation is inherently sequential, but you can still batch several independent questions at once.

Self-Consistency sampling.
Building on Chain-of-Thought, proposes drawing K independent reasoning chains at a modest temperature and letting the majority answer win . Accuracy rises because random errors in individual chains cancel out. The trade-off is straightforward: all costs—tokens, VRAM, wall-time—scale linearly with K. Fortunately every chain is independent, so you can batch or scatter them in parallel on the same GPU and saturate its cores with virtually no orchestration overhead.

Verifier-guided best-of-N (rejection sampling).
Instead of trusting raw generations, you run a lightweight verifier—regular expressions for math, unit tests for code, or an automatic reward model—and keep only the candidates that pass 03_large_LMs. When several answers survive, select the most frequent or highest-scoring one. This strategy often matches the quality of a far larger model while emitting *shorter* answers than full CoT traces. Computation now has two components: generating *N* candidates (GPU-heavy) and scoring them with the verifier (usually much cheaper). Both stages batch cleanly, so they parallelise almost perfectly on a single big GPU.

Iterative planning, back-tracking and self-evaluation (O1-style reasoning).
OpenAI's O1 model, which "learns to recognise and correct its mistakes" effectively turning the LLM into a mini search engine that plans, backtracks and checks itself. Compared with plain CoT, this approach explores many partial solutions and revises them, achieving state-of-the-art performance on long-horizon problems. The flip side is heavy token and memory usage—every intermediate attempt lives in the context—and extra CPU logic to manage the iterative loop. Some parallelism is possible by batching all candidate expansions at a given step, but overall the process remains sequential because each revision depends on the preceding attempt.

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 semantic equivalence and ignores non-contradictory embellishments, whereas the lower-performing one is thrown off by word-level asymmetries like the presence/absence of "Texas" and the added wind-strength detail.