Git: https://github.com/gaiaosadchy/ANLP1.git

Open Questions:

1.

- **Dataset name:** Lots-of-LoRAs/task891_gap_coreference_resolution. https://huggingface.co/datasets/Lots-of-LoRAs/task891_gap_coreference_resolution/viewer/default/train?views%5B%5D=train

  Why it measures an intrinsic property of language understanding: Coreference resolution requires understanding linguistic structure, semantics, and discourse context to link pronouns to their correct antecedents. By evaluating whether a model can accurately identify which entity a pronoun refers to, the dataset directly tests the model's grasp of essential language comprehension mechanisms.
- **Dataset name:** luheng/qa_srl https://huggingface.co/datasets/luheng/qa_srl

  Why it measures an intrinsic property of language understanding: By casting semantic role labeling as question answering, this dataset probes a model's grasp of predicate-argument structure – its ability to identify who did what to whom, when, and how – which reflects core semantic understanding of sentences.
- **Dataset name:** lavallone/selection_semcor https://huggingface.co/datasets/lavallone/selection_semcor/viewer/default/train?row=0&views%5B%5D=train

  Why it measures an intrinsic property of language understanding: Word-sense disambiguation evaluates a model's ability to use contextual cues to distinguish among multiple dictionary definitions of a word. By requiring selection of the appropriate sense of a word in context, the task directly tests fine-grained lexical semantics, an intrinsic property of language understanding.

2. a.

- **Self-Consistency:**

  Description: Sample k independent chain-of-thought (CoT) outputs from a single prompt, then take the most frequent final answer.

  Advantages:

  - Averages out errors in individual reasoning traces.

  - Often yields better accuracy than a single greedy CoT.

  Computational Bottlenecks:

  - k × as many forward passes at inference.

  - Storage for holding all CoT sequences before voting.

  Parallelizable?:

  Yes – each sample is an independent model call, so we can batch them concurrently.

- **Verifiers:**

- Description: First generate one or more candidate answers, then run a secondary "verification" pass using a smaller specialist to rank those candidates for correctness.

  Advantages:

  - Provides an explicit check on answer quality.

  Computational Bottlenecks:

  - Two full inference passes per candidate: one to generate, one to verify.

  - Potential combinatorial blow-up if we verify many candidates.

  Parallelizable?:

  Yes – generation and verification steps for different candidates can be run in parallel.

- **Increasing Compute Budget:**

  Description: Rather than simply "dialing up" to a bigger model or more beams/samples, we can reallocate a fixed compute/runtime budget across multiple calls to a smaller model. For example, instead of running a single

70B model once, we can run a 13B model five times and then select the best output with a unit-test setup.

Advantages:

- Better resource utilization: When we have a reliable unit test setup, repeated small-model samples can outperform the single large-model pass.

- Flexibility: We avoid the latency/memory spikes of a huge model and can more easily parallelize small-model calls.

Computational Bottlenecks:

- We need a mechanism (unit tests) to pick the correct output among many candidates – this can itself be costly if tests are expensive.

- In scenarios where unit-tests are unavailable, a ranking-based selection of candidates from the smaller model falls short of the performance of a single output from larger ones.

Parallelizable?:

Fully parallelizable: each small-model generation and its subsequent test can run concurrently.

- **Length of CoT:**

  Description: Prompt or constrain the model to produce longer, more detailed chain-of-thought rationales before giving a final answer.

  Advantages:

  - Encourages the model to unpack complex reasoning steps, which can improve accuracy on hard problems.

  - Makes mistakes more interpretable (we can see where the chain breaks).

  Computational Bottlenecks:

  - Longer token sequences: quadratic (self-attention) and linear (token generation) compute/memory growth.

  - Higher latency per query.

  Parallelizable?:

Only at the level of independent examples or through model/data parallelism. within one long CoT, token generation is sequential.

b. I would choose **Self-Consistency**, because we can cheaply get robust, diverse reasoning paths from one large model by batching multiple chain-of-thought samples on my high-memory GPU, and then take a majority vote. This requires no extra models or complex prompts, runs each sample in parallel on the GPU, and extracts more reliable answers without needing an even bigger model.

Programming Exercise:

- Yes. The epoch_num: 3, lr: 5e-05, batch_size: 16 run had the best validation accuracy (0.8578) and the best test accuracy (0.82957).
- After comparing the validation dataset predictions of the best and worst configurations, I found 22 examples where the best performing model was correct but the worst performing model failed, and 50% of those (11 out of 22) involved numeric content. So the lower-performing model struggles primarily with **numerical reasoning**.