A Anatomy of the prompt

The example below illustrates the structure of a dataset item:

Entity Linking Example

Instruction: *This is an entity linking task. The goal for this task is to link the selected entity mention in the table cells to the entity in the knowledge base. You will be given a list of referent entities, with each one composed of an entity name, its description and its type. Please choose the correct one from the referent entity candidates.* [...]

Input: `[TLE] List of high schools in South Dakota.`
`col: |school|type|city|county|mascot|`
`row 0: |Aberdeen High School|Private|Aberdeen|Brown|Knights|`
`row 1: |Agar High School|Public|Agar|Sully|`
`[...]`
`row N: |Alcester-Hudson High School|Public|Alcester|Union|Cubs|`

Question: *The selected entity mention is: 'Hyde'. The column is 'city'. The referent candidates are:* `<Dr. Jekyll and Mr. Hyde [DESC] fictional characters [TYPE] fictional characters group>` [...]
What is the correct referent entity for 'Hyde'?

Postilla: *Answer with just a candidate, selected from the provided referent entity candidates list, and nothing else. The selected candidate must be reported verbatim from the list provided as input. Each candidate in the list is enclosed between < and > and reports [DESC] and [TYPE] information.*

Answer:

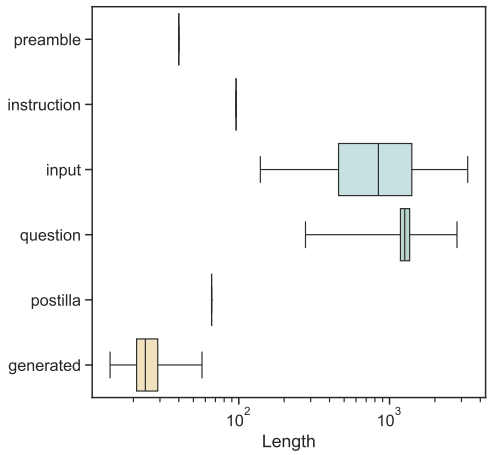

Fig. 9: Distribution of the number of tokens per prompt segment (TableLlama)

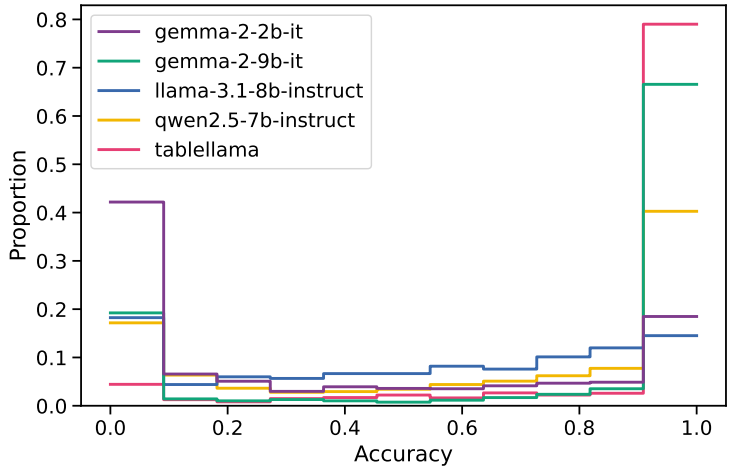

Fig. 10: Accuracy distribution over 10 non-deterministic generation runs derived from the same prompt

B Accuracy distribution

Figure 10 reports the accuracy distributions for the models utilized in the experiments of the paper. The models are evaluated on the target task over $N = 10$ generation runs. Cases with zero accuracy can be further split by uncertainty according to Table 1.

C Answer variability as a function of temperature

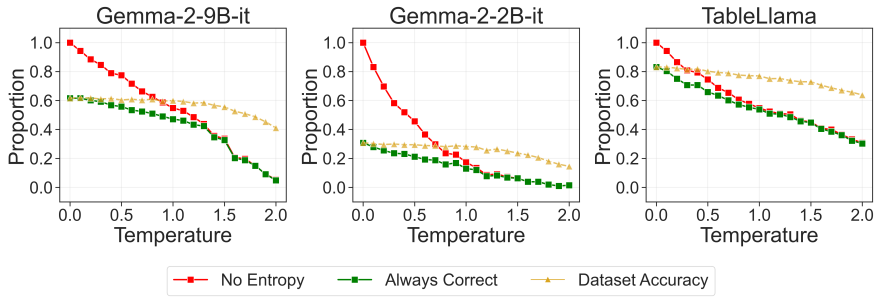

Fig. 11: Sensitivity to temperature of prompts with no output variation (*red*), prompts with correct answers only – still no output variation (*green*), and prompts with correct answer on average (≥ 0.5, *yellow*), **(left)** Gemma-2-9b-it, **(center)** Gemma-2-2b-it, and **(right)** TableLlama.

To evaluate the variability of answers and recoverable cases, we systematically sweep temperature values in the range $0.0 \leq \tau \leq 2.0$ with steps of 0.1 to assess the temperature effect on (1) the number of items that show output variability and (2) the average accuracy. For capacity reasons, we utilize a subset of 200 elements from the original dataset. Figure 11 shows the results of the experiment for two selected LLMs. The *yellow* series shows average accuracy, highlighting how higher τ values deteriorate the performance on the task. At the same time, lower τ settings yield the highest proportion of always correct cases (*green* line). However, again at low τ, output variability is also lowest; the uncertainty of these cases cannot be estimated, making this setting unsuitable for recovery. Operationally, a reasonable trade-off for τ is to minimize the accuracy loss while maximizing the number of cases recoverable through uncertainty, that is, those above the *red* line. Another way to view this trade-off is by examining the difference between the *red* and *green* lines: this difference represents the proportion of unrecoverable cases, that is, cases that are always wrong and have no output variability. Making these cases recoverable requires "paying" by reducing always correct cases, while trying to preserve average accuracy (*yellow*) as much as possible. This experiment illustrates how temperature can be adjusted depending on specific combinations of model, task, and application constraints.

D Spearman correlation between estimated and true PE/SE

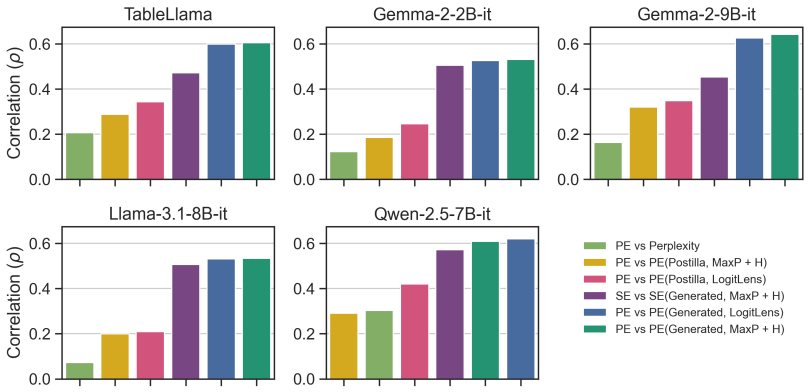

Fig. 12: Spearman's rank correlation coefficient (ρ) between the estimated entropy (PE or SE) and the multiple-generations entropy (PE or SE) across models. In the legend, the notation "`Target(Segment, Observable)`" indicates that the target variable *Target* was predicted using a regressor based on *Observable* features extracted from the *Segment* segment of the prompt.

E Transformer-based time complexity

To further assess the practicality and usability of our approach, in this section we derive the time complexity of a Transformer-based architecture, such as the ones used by the models considered in this work. We consider a Transformer with L layers, and d hidden size. The context length is set to N, while the number of generated tokens is G. The time complexity of a single forward pass over a prompt $X \in \mathbb{R}^{N \times d}$can be decomposed into the following components:

- **Self-attention:** Given the $Q, K, V \in \mathbb{R}^{N \times d}$ matrices, the time complexity of the self-attention mechanism is $O(N^2 \cdot d)$, where N is the context length and d is the hidden size. The attention scores are computed as QK^T, and the output is computed as $\sigma(QK^T)V$. The time complexity of this operation is $O(N^2 \cdot d)$.
- **Feed-forward:** During the feed-forward step, the time complexity is $O(8N \cdot d^2) = O(N \cdot d^2)$ overall, where d is the hidden size. In this we can include also the projection of the input to the QKV space and the final projection to the output space.

In total one has a time complexity of $O(N^2 \cdot d + N \cdot d^2)$ for a single Transformer layer, which becomes $O(L[N^2 \cdot d + N \cdot d^2])$, where L is the number of layers.

If we now suppose to generate G tokens without the use of a KV-cache, the time complexity of the g-th generation step is $O(L[(N+g)^2 \cdot d + (N+g) \cdot d^2])$, for every $g \in [G]$. If we then sum over all the generations, we have:

$$O\left(\sum_{g=1}^{G} L\left[(N+g)^2 d + (N+g)d^2\right]\right) = \tag{4}$$

$$O\left(L\sum_{g=1}^{G}\left[(N+g)^2 d + (N+g)d^2\right]\right) = \tag{5}$$

$$O\left(L\sum_{g=1}^{G}\left[(N^2+2Ng+g^2)d + (N+g)d^2\right]\right) = \tag{6}$$

$$O\left(L\left[(GN^2+2NG^2+G^3)d + (GN+G^2)d^2\right]\right) = \tag{7}$$

$$O\left(L\left[G(N+G)^2 d + G(N+G)d^2\right]\right) = \tag{8}$$

$$O\left(LG\left[(N+G)^2 d + (N+G)d^2\right]\right) \tag{9}$$

which shows that the time complexity of generating G tokens is quadratic in the number of overall tokens $N+G$, when $G \ll N$, otherwise it would become cubic in the number of generated ones.

When a KV-cache is used, while the time for processing the prompt is the same, a major computational saving is obtained during the generation phase. In this case, the time complexity of the g-th generation step can be decomposed into the following components:

- **Self-attention:** During the self-attention, Q reduces to a single vector $q_g \in \mathbb{R}^{1\times d}$, while K, V becomes $K_g, V_g \in \mathbb{R}^{(N+g-1)\times d}$. Overall, the time complexity of this operation is $O((N+g)\cdot d)$.
- **Feed-forward:** Since the output of the Self-Attention has a dimension of $1\times d$, the feed-forward step has a time complexity of this operation is $O(d^2)$, which includes the projection of the input to the QKV space and the final projection to the output space.

If we then sum over all the generations $g \in [G]$, we have:

$$O\left(\sum_{g=1}^{G} L\left[(N+g)d + d^2\right]\right) = \tag{10}$$

$$O\left(L\sum_{g=1}^{G}\left[(N+g)d + d^2\right]\right) = \tag{11}$$

$$O\left(L\left[G(N+G)d + Gd^2\right]\right) = \tag{12}$$

$$O\left(LG\left[(N+G)d + d^2\right]\right) \tag{13}$$

which shows that the time complexity of generating G tokens is linear in the number of overall tokens $N+G$, when $G \ll N$, otherwise it would become quadratic in the number of generated ones.

Table 2: Time complexity of the Transformer-based architecture. L is the number of layers, N is the number of tokens in the prompt, G is the number of generated tokens, and d is the hidden size.

Phase	**Without KV-cache**	**With KV-cache**
Prompt processing	$O(L[N^2\cdot d + N\cdot d^2])$	$O(L[N^2\cdot d + N\cdot d^2])$
Generation	$O(LG[(N+G)^2 d + (N+G)d^2])$	$O(LG[(N+G)d + d^2])$

This simple derivation shows that our proposed approach to learn to estimate the multiple-generations Predictive Entropy (PE) is still computationally promising even when advanced KV-cache techniques are employed during inference.