University of Pisa

Human Language Technologies Project: Zero Shot Learning with LLMs

Pasquale Esposito[1] and **Sergio Latrofa**[2]

[1] *M.Sc. Computer Science, Free Curriculum* - 649153 - p.esposito8@studenti.unipi.it
[2] *M.Sc. Computer Science, Artificial Intelligence* - 640584 - s.latrofa1@studenti.unipi.it

July, 2023

1 Abstract

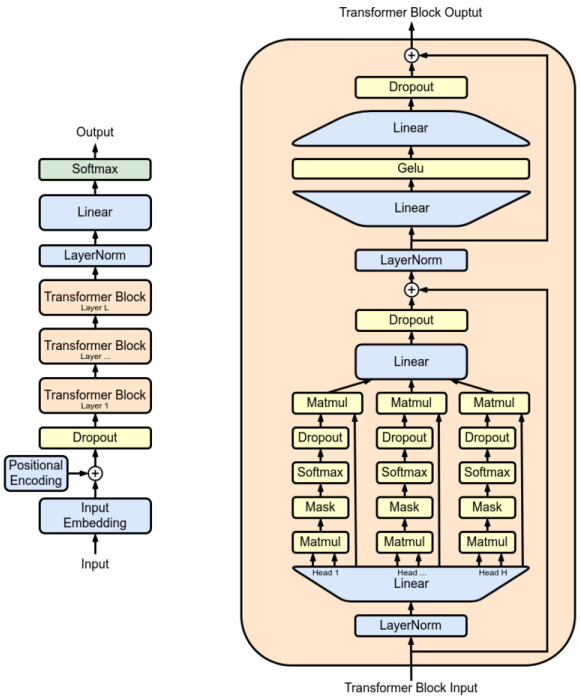
The aim of this work is to verify effectiveness of Large Language Models on Zero Shot Learning tasks, comparing models of different age complexity, prompts and task configuration in order to produce an overall view of model capabilities and characteristics, checking whether the intrinsic capability of adaptation complex reasoning evolves along with the number of trainable parameters and if it can emerge even without further specialization on new data.

2 Background

Around 2018, with the publication of "Attention Is All You Need" [3] and the progressive success of transformers based Large Language Models (LLM) on several classical NLP tasks, researchers started questioning how far the acquired knowledge can be exploited, even with minimal (or without) tuning. Several hypotheses were done, in particular, when releasing the GPT2 model [5], Brown et al. claimed that LLM were general purpose multitask learners, to further re-postulate it with GPT3 [11], claiming LLM to be few shot learners. In the ML community, few-shot learning refers to the practice of presenting a few (training) examples to the model and right after presenting it the test examples. The extreme version of such practice is zero shot learning, whereas no examples at all are presented to the model except the test ones, which are "brutally" asked with just a brief prompt to specify what type of answer is expected given the input. Such last scenario consists in a quite new setup for AI models, where the only hyperparameter to tune is the prompt itself, depending on model and on dataset/task configuration. Surprising results are being reached, even with not too over parameterized models (at least if compared with latest ones).

3 Models

This section consists of a brief description of the LLMs chosen for our analysis, with some hints on their peculiarities and on the reasons behind their pick for our experiments. We decided not



(a) GPT Architectural Schema

to go too much into technical details, assuming a good prior knowledge of the reader and so trying to keep this document closer to a report more than a school book.

3.1 GPT-2

GPT-2 [5] is a Transformer based LLMs released in 2018. It improves its predecessor, the GPT (Generative Pretrained Transformer) model from OpenAI [5] with a layer normalization moved to the input of each sub-block, similar to a pre-activation residual network [2] and an additional layer normalization added after the final self-attention block.

A modified initialization, which accounts for the accumulation on the residual path with model depth, is used. Context size was increased from 512 to 1024 tokens and a larger batch size of 512 is used. The vocabulary was expanded as well counting 50,257 tokens. Finally, weights of the residual layers were scaled at initialization by a factor of $1/sqrt(N)$ where N is the number of residual layers.

GPT-2 was trained on the WebText dataset, created analog with the model itself, scraping "high quality" (according to an internal quality definition) English documents from all around the web (40GB from over 45 million pages). Objective concerned the prediction of a masked world given all the other ones in the sentence (self-supervised), and allowed to learn a wide internal features set to represent the English language.

Such a model was chosen as its publication is one of the initial steps on the research on few-

shot adaptation. In particular, the main motivation behind its creation was the proof of the Unsupervised Multitask Learners hypothesis. GPT-2 in fact was used to demonstrate that unsupervised training of Large Language Models (billions of parameters) on a very large and heterogeneous dataset allows embedding knowledge to solve a variety of tasks without explicit architectural adaptation or fine-tuning, but just performing a proper prompt interaction: the *zero shot learning* that we are trying to explore. Performance of GPT-2, at times, was generally promising but certainly improvable (not better than a random classifier on some tasks). For such reasons, we decided to use it as a "weak benchmark" in order to check whether later models actually outperform it.

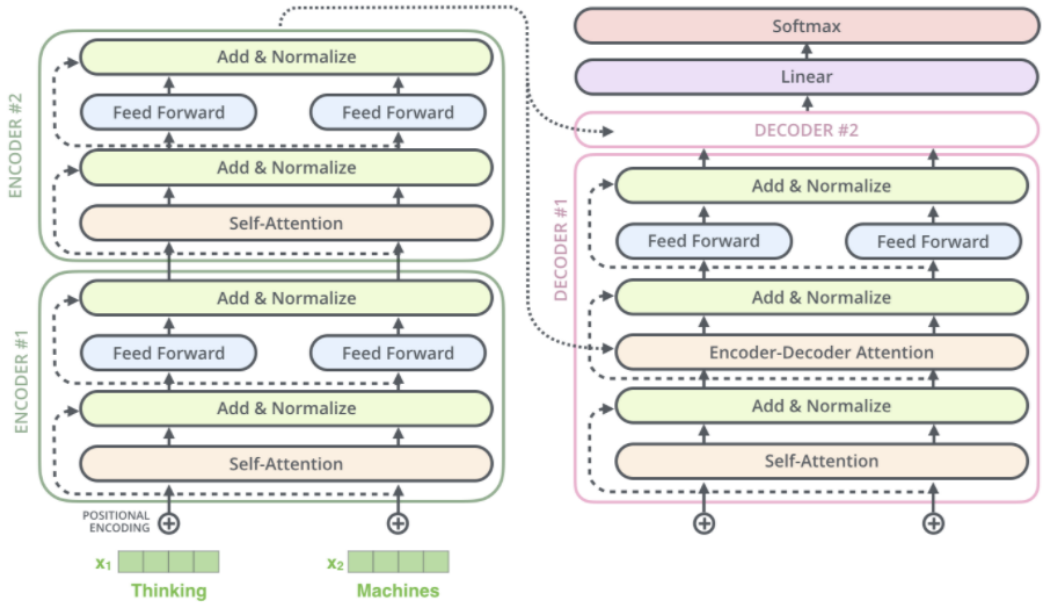
3.2 BART

BART [8] is a pre-trained model, released in 2019, combining Bidirectional and Auto-Regressive Transformers; it is a denoising autoencoders, applicable to a very wide range of end tasks. During pre-training text was first corrupted with an arbitrary noisy function, and then the sequence-to-sequence model is learned to reconstruct the original test. BART uses the standard sequence-to-sequence Transformer architecture from [3] except, following GPT, that they modified changing ReLU activation functions was changed with GeLUs [16] and initializing parameters were sampled from $N(0, 0.02)$. The architecture is closely related to that used in BERT with some difference in attention propagation mechanism. In total, BART roughly contains 10% more parameters than BERT. In general HLT scenarios, BART can be fine-tuned for specific tasks, such as sequence classification, token classification, sequence generation and machine translation. In our case, we chose the `bart-large-mnli` checkpoint with 12 layers in the encoder and decoder and over 400M parameters, wich thanks to its implementations can infer a class over a user specified set of options. We accepted the relatively high number of parameter and an expected high inference time as the cost to pay for a model specifically fine-tuned for zero shot adaptation.

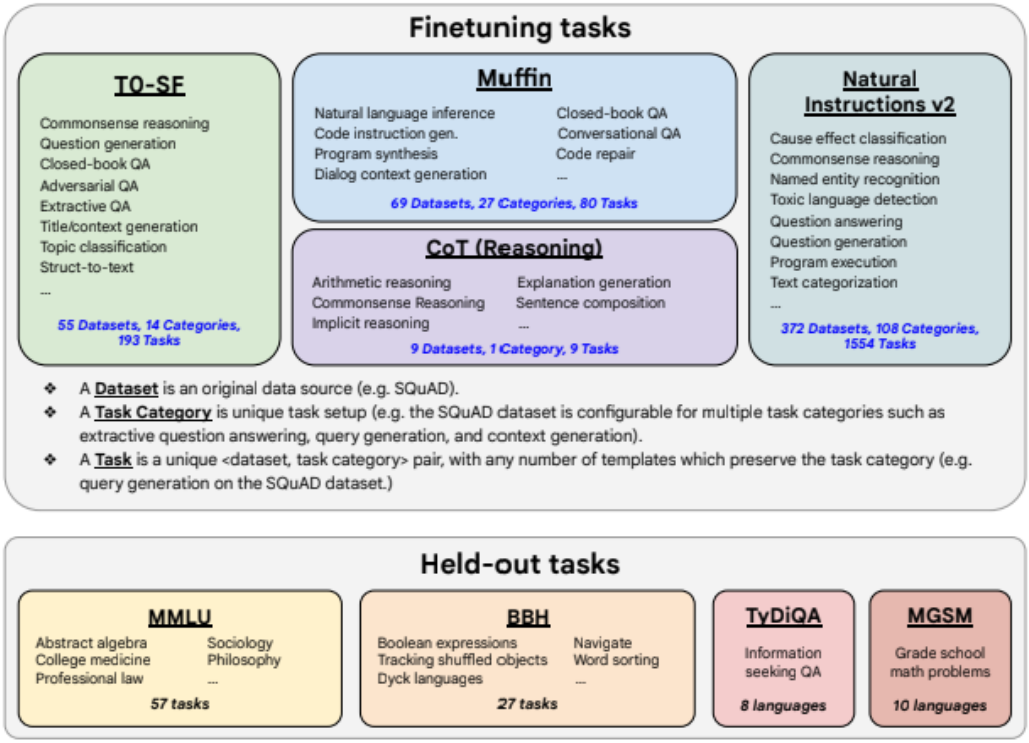
3.3 FLAN-T5

FLAN [15] was released in 2022 as a strong fine-tuned version of T5[13], claiming to have strong zero-shot, few-shot adaptation outperforming state of art public checkpoints such as T5 itself. Prior literature has shown that increasing the number of tasks in fine-tuning with instructions improves generalization to unseen tasks. FLAN-T5 was so fine-tuned on 1,836 tasks using 64 as batch size, 0.05 as dropout, 5e-4 as learning rate and 84k as number of steps; in total, 250M of parameters were trained. For what concerns its architecture, its reported schematically in figure, but the original T5 paper can be referenced [13]. We can just say that it is once again a sequence-to-sequence model with good generative capabilities.

The `base` and the `small` checkpoints were chosen (80M and 250 parameters respectively). Using two of them we expected to analyze whether the few-shot specific training pipeline of FLAN can beat its competitors even with a very smaller number of parameter (`FLAN-small` has 1/5 of the parameters of BART), and if behaviors of the two versions of the same model with respect to the same prompt is similar or generally independent.



(a) T-5 Architectural Schema



(a) FLAN Fine-tuning schema.

4 Methodology

We decided to evaluate the evolution of zero shot capabilities of few recent years models, starting from the oldest GPT-2, then passing to the BERT family, represented by BART, to end up with the latest FLAN-T5 (using two checkpoints for this last model, the "`small`" and the "`base`" ones). Two families of tasks were chosen, multiple options question answering (QA) and Natural Language Inference (NLI), represented respectively by 8 and 5 data-sets with varying configurations (more details and about datasets can be found in the appendix). For the two tasks we also gathered numerous prompts, trying to provide diversified ways to interrogate the models, hopefully finding the most suitable and effective for each one. For example, a NLI query can be provided in several ways, like a unique couple of sentences, or with an isolated context, followed with the prompting sentence and then by the hypothesis to be checked whether if entailed or in contradiction.

A pre-processing pipeline was implemented, studying the configuration of each dataset and generalizing with a unique function able to convert input sentences in the desired parts (question and options for QA, context and hypothesis for NLI). Such a function mainly operates low level string manipulation, adapting to the separation policy of each dataset (i.e. sometimes in NLI context and hypothesis were separate with the "[SEP]" substring).

Once the pre-processing was completed, a prompt selection phase was carried on, using the validation/dev sets of the selected dataset, already provided in a split way thanks to the crossfit challenge [14] configuration. Initial idea was to use the crossfit library itself, but due to numerous out versioning and configuration problems we decided to switch to a manual implementation of the desired experiments, which results simpler in terms of debugging and surely richer in terms of acquisition of developing and coding skills.

Prompt selection was carried on evaluating each prompt on the dev set of each dataset, and then picking the best three prompts for each dataset-model couple, obtaining 28 triplets for QA and 20 for NLI.

Such $< model, dataset, [prompt_1, prompt_2, prompt_3] >$ triplets were used to evaluate test data, picking up the best results in terms of accuracy within the three prompts to compare models. Results were used in the discussion section to delineate an overview on evolution of zero shot capability of LLMs and also to qualitatively analyze how prompt engineering affects such a skill, and how prompt characteristics differently adapt to the various models (and also specifically to different versions of the same models).

4.1 Implementation

The various experiments were carried on few Google Colab Notebooks (Python programming language). For what concern dataset, they were priorly selected from the `crossfit` environment, which we installed on our local machines, before discovering several open issues with Python dependencies, which lead us to move to Colab. We manually attached to the notebooks the *.csv* files of the various datasets previously downloaded along with crossfit installation. Models were taken from `Hugging Face` checkpoints. For what concerns other used libraries, we

have to mention `PyTorch`, `transformers` for models and `pandas` for data visualization.

Between the written code, there can also be found string manipulation based function to uniform dataset syntax by task, and to convert "FLAN" fully generative style answers in one of the precise options provided in the prompt (i.e. "is [context] entailed by "hypotheses?" if FLAN answered "no" it was converted to a Boolean *False*). Another hand-made solution was the one to cut the $glue - mnli$ dataset and evaluate only the the first 1000 rows. Such decision was taken seen the high number of test instances (if compared to the other datasets) and the fact that rows are not ordered by any criteria (no loss of generality); hence a large amount of time and computation during tests has been saved (this would have caused RAM problems with the largest models like BART and FLAN-base).

5 Results

In this section we report the aggregated the results of the experiments previously described. For the two tasks, we aim at analyzing the average and the maximum performance comparing models and prompts in parallel, according to the variation of complexity linked to each datasets.

Another important aspect is the detection of the sources of variance in performance, which can be aided to models, prompts and tasks as well, and can so be observed only with the support of some empirical results.

In the end, we also reported the mean execution time of each `<model, dataset>` couple, in order to observe if some correlation with accuracy and model "age" or model complexity occurs.

5.1 Prompt selection results

In this first section we summarize the results achieved on the validation/dev set of each dataset (splitted in advance), using a wide set of different prompts for each task. More precise details about prompts and individual performance can be found in the appendix. Observing the mean performance for each `<model, task>` couples (??,??) it's easy to spot general information about difficulty of each dataset and prompt related variance. On the other hand, observing the performance distributions, it's also possible to draw an idea of model invariance with respect to the prompt. For example, observing mean performance distribution of FLAN (both small and base) on the QA task, it's easy to notice a certain "dicotomical" behavior, which can be interpreted as the existence of several "bad performing prompts" and contemporary several "good performing" ones, also having seen its mean performance chart. Such behavior suggest a strong dependence on the prompt for FLAN, which, for example is something not so common for GPT-2 on the same task (QA), with a generally lower performance,but more invariant with respect to the chosen prompt. A similar reasoning with the same two models can also be done for the NLI task. BART behavior seems more invariant on the QA task (majority of results very "close" to the mean), and slightly more inconstant on the NLI task.

Not prompt invariant behavior many be warning for the subsequent evaluation phase, in the sense that for a larger amount of unseen qualitative data (as NLP ones are), performance

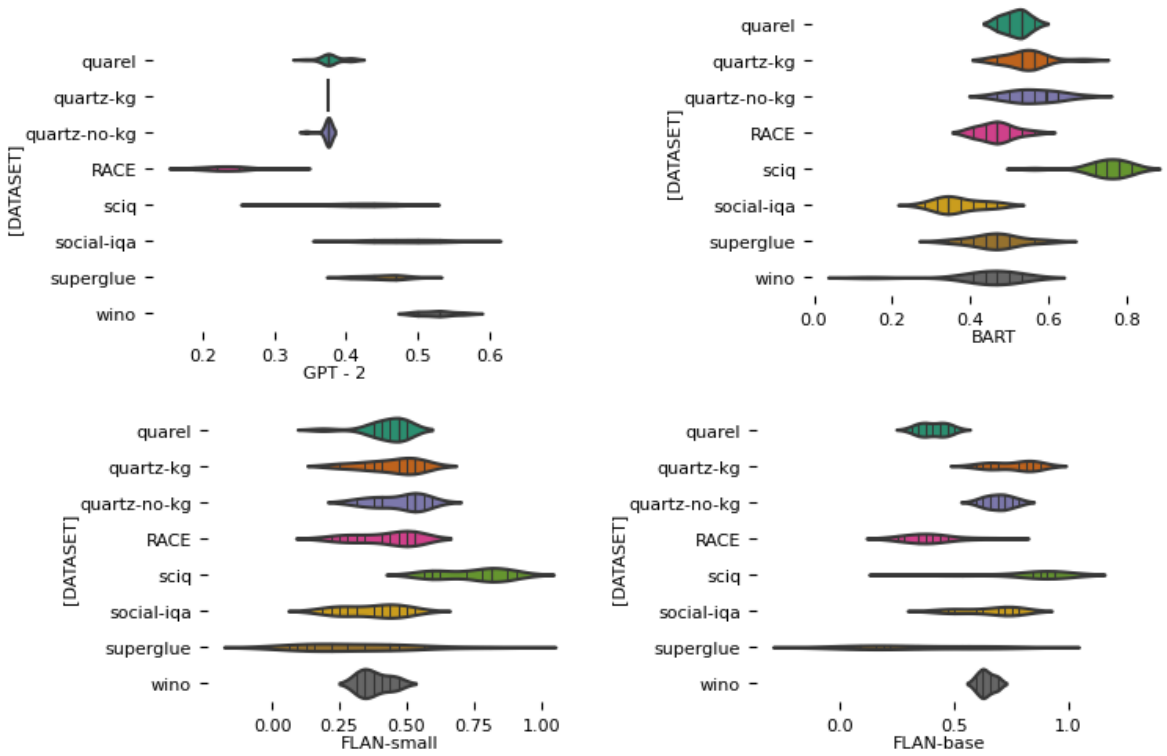


Figure 4: QA Prompt selection performance

achieved with a particular prompt on the validation set are "not so probable" to be replicated as for an invariant model (spoiler - this is exactly what will happen for FLAN on *anli*, observing a dramatic decay of the accuracy).

5.2 Test results

Dataset	GPT-2	BART	FLAN-small	FLAN-base	Mean	STD
scitail	0.535	0.545	0.707	0.811	**0.65**	**0.134**
anli	0.332	0.348	0.382	0.462	*0.381*	*0.0586*
glue-mnli	0.348	0.362	0.64	0.804	0.539	0.223
sick	0.348	0.388	0.665	0.614	0.504	0.147
superglue-cb	0.464	0.464	0.679	0.839	0.612	0.188
Mean	*0.405*	0.421	0.615	**0.706**	0.537	-
STD	*0.09*	0.082	0.132	**0.163**	-	0.172

Table 1: Maximum zero shot accuracy reached on NLI test data.

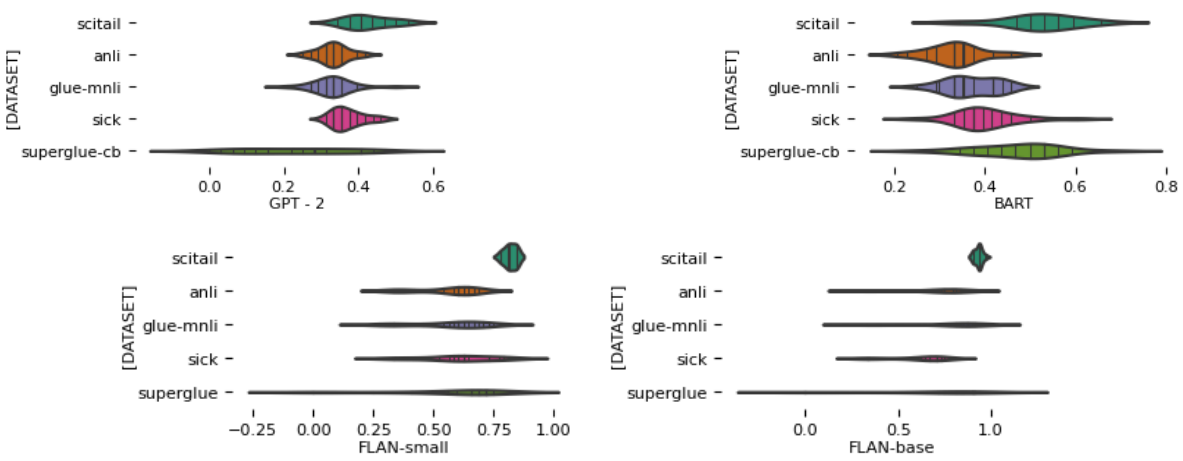


Figure 5: NLI Average Prompt selection performance

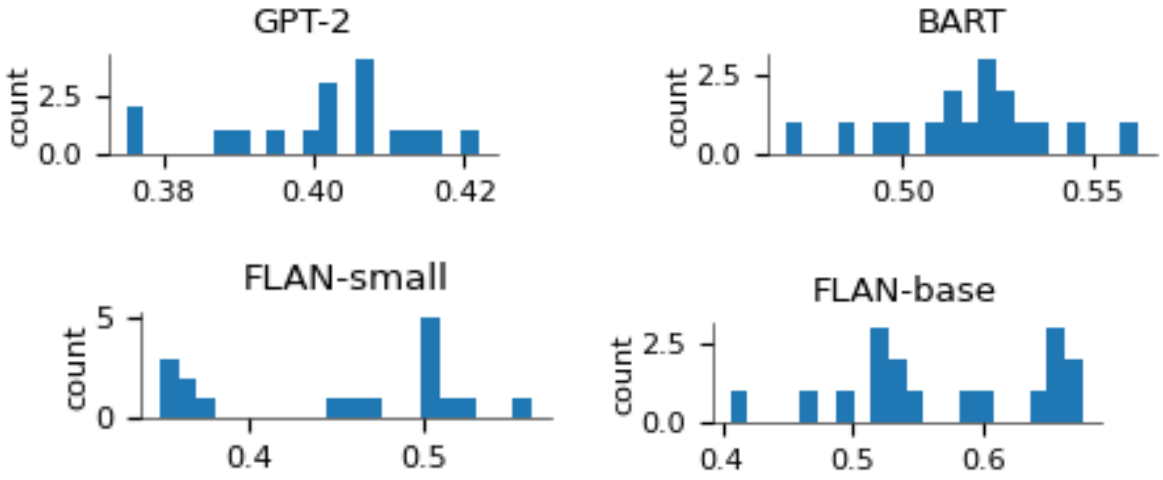


Figure 6: QA Average Prompt performance distributions

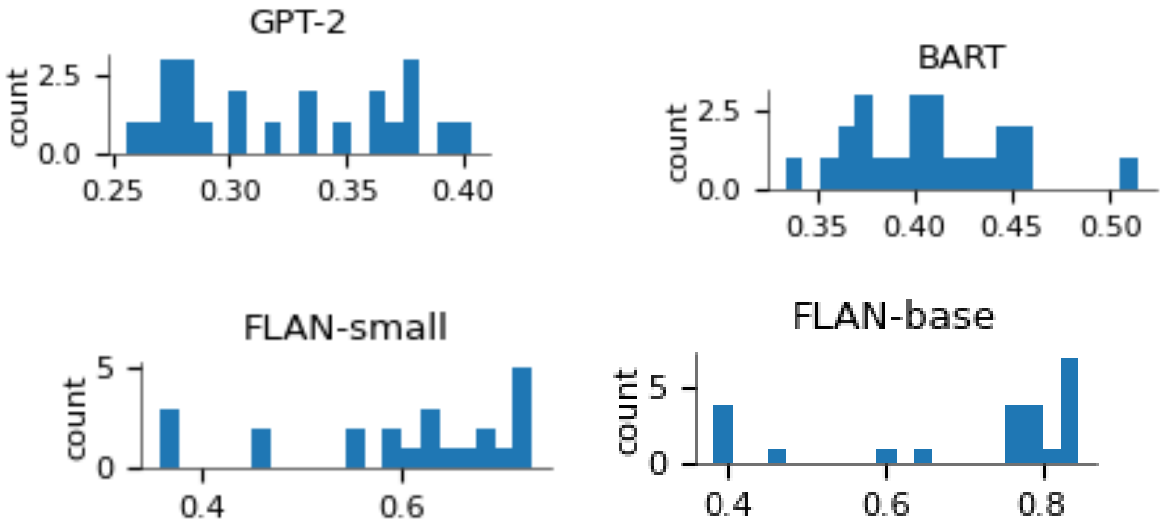


Figure 7: NLI Prompt performance distributions

Best overall results where reached by FLAN-base. For NLI, also FLAN-base behaved quite well (expectable as it is fine-tuned on a bunch of tasks including NLI itself). Anyway, weak points of such last model were highlighted by the probably "hardest" datasets: *anli*. Such weakness may be addressed to the usage of a relatively small version of the model, less depth and so with lower high order information processing capabilities, according to deep learning literature and Additionally, we can quite safely state that test data for such dataset, are harder than validation ones, being the performance decay a constant throughout all the models. Another interesting observation is the poor performance of FLAN-base on *sick*, where it does not even managed to surpass its smaller version.

Dataset	GPT-2	BART	FLAN-small	FLAN-base	Mean	STD
QuaRel	0.504	0.514	0.514	0.651	0.546	0.07
QuaRTz-no_knowledge	0.503	0.555	0.544	0.656	0.565	0.065
QuaRTz-with_knowledge	0.5	0.550	0.529	0.711	0.573	0.095
RACE-middle	0.257	0.414	0.380	0.582	*0.408*	*0.134*
SciQ	0.409	0.822	0.815	0.945	**0.748**	**0.234**
Social-IQA	0.361	0.392	0.420	0.606	0.445	0.11
SuperGLUE-COPA	0.520	0.660	0.37	0.58	0.533	0.123
Wino-Grande	0.510	0.506	0.503	0.55	0.517	0.022
Mean	*0.446*	0.455	0.51	**0.669**	0.54	-
STD	*0.095*	0.138	0.141	**0.133**	-	0.146

Table 2: Maximum zero shot accuracy reached on QA test data.

For QA task BART and FLAN-small performance are generally aligned, with the former model sometimes surpassing the latter (something not observed for what concerned NLI). $RACE$, $SuperGLUE$ and social $Social - IQA$ may be identified as the "hardest" datasets. Such performance decay was easy to expect, especially on $RACE$, give the high number of options and the generally large size of each question. Still on these "hardest" cases, performance of FLAN-small are aligned and sometimes surpassed by BART, which can be viewed as more suitable for complex-context zero-shot inference and weaker on easier ones. On the other hand, such trends clearly confirmed the scalability limitations (in terms of task difficulty) of FLAN-small, already noticed in NLI, which clearly do not concern larger models like BART or FLAN-base.

GPT2 performance, which according to the mean, can be considered the model performing "worse", follow the dataset "difficulty" trend highlighted by the other two models, though still generally surpassing the random baseline (which depends on the number of options of the task), and are hence considerable poor but robust (we may venture to say "high bias and low variance").

5.3 Query time

Finally, in the light of the previously run experiments, we observed that switching between two prompts while keeping fixed the model and the dataset, does not significantly affect average computation times. Hence, querying times were evaluated randomly switching between all prompts suitable for a certain dataset, drawing some average quantities.

Dataset	GPT-2	BART	FLAN-small	FLAN-small	Mean	STD
scitail	19.882	92.758	9.184	24.61	*36.608*	*44.473*
anli	30.937	232.85	8.922	27.113	74.956	124.367
glue-mnli	17.545	148.631	6.694	18.448	47.83	78.774
sick	15.181	123.214	8.743	19.002	41.535	63.336
superglue-cb	26.688	226.216	7.206	43.721	**75.958**	117.334
Mean	22.047	**164.734**	*8.149*	79.298	55.377	-
STD	6.57	**62.417**	*1.121*	10.266	-	71.389

Table 3: Classification time of 32 NLI samples (in seconds).

Dataset	GPT-2	BART	FLAN-small	FLAN-base	Mean	STD
QuaRel	17.788	74.978	6.923	20.62	29.327	29.04
QuaRTz-no_knowledge	16.618	64.880	4.423	16.018	*24.66*	*27.755*
QuaRTz-with_knowledge	18.395	74.232	4.805	19.224	29.164	30.764
RACE-middle	67.361	619.013	17.874	183.851	**222.024**	**273.652**
SciQ	42.081	328.767	10.632	37.13	97.153	154.888
Social-IQA	24.255	136.485	8.589	24.22	48.387	59.193
SuperGLUE-COPA	15.792	76.452	11.101	31.171	33.629	29.808
Wino-Grande	13.64	69.251	4.022	15.178	25.523	29.567
Mean	23.241	**180.132**	*8.133*	43.427	167.972	-
STD	18.193	**198.513**	*5.191*	57.237	-	120.594

Table 4: Classification time of 32 QA samples (in seconds).

BART was the slowest model overall. Comparing it to GPT2, supposed to be less precise, the slight amount of gained accuracy made the trade-off not worth at all. FLAN-small and outperformed both the two competitor models also in terms of time, confirming that the *small* configuration (obtained with a smaller architecture and a smaller training set [**FLAN-small**]) can maintain acceptable knowledge saving a lot of parameters to be tuned and as showed, significantly reducing the feed-forward inference time, independently by the task. The *base* version, in fact, though clearly more precise has significantly larger inference times. Finally, the high difficulty of the *RACE* dataset is confirmed by the average inference time of all the models on it, clearly dependent on the large size of each question. Processing time and dataset complexity, on the other hand, seemed to be generally uncorrelated for NLI.

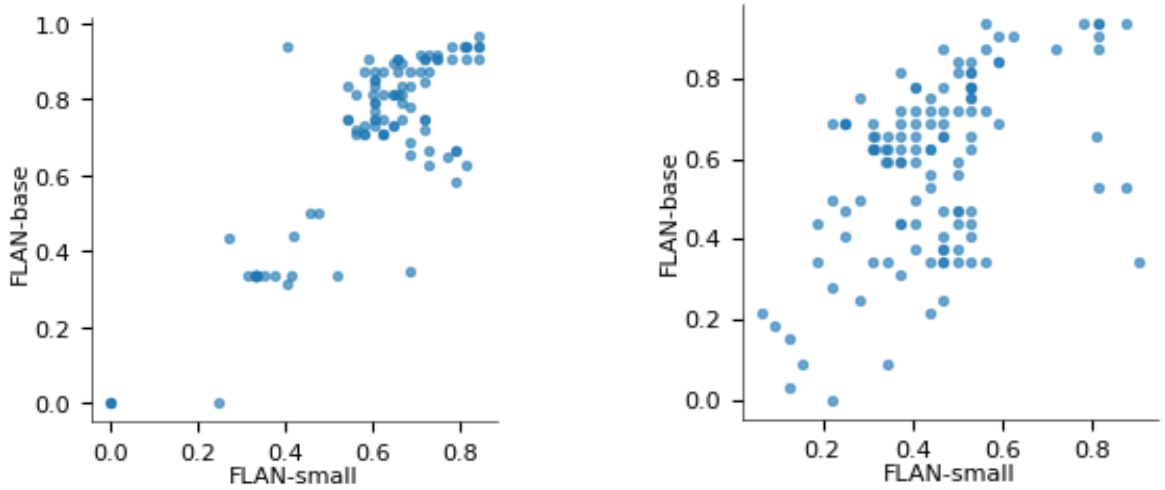


Figure 8: **Prompt correlations** in terms of *validation mean accuracy* between the two checkpoint of FLAN(NLI on the left and QA on the right).

6 Discussion

6.1 The impact of the "right" prompt

The first observation concerns prediction variance. Although in general results are more or less aligned (both *model-wise* and *prompt-wise*), a change in the used prompt may lead to a decay of performance, and, on the other hand, given a new set of data, even in the context of the same task, we have no guarantee that a `<model, prompt>` couple which has previously performed well will repeat themselves. There can easily be found another prompt for the same model, which performed worse on the easier set and better on the worst one, making prompt effectiveness hard to analyze without a well-defined context.

Observing also the tables in the appendix, it's easy to notice that even for cases where a good performance is reached (i.e. with FLAN and *scitail*), such result has quite high variance related to prompts. We did not manage to identify a clear correlation within some certain particular kind of prompt, models and tasks. Anyway our result highlighted the need of deep "cross-prompt" analysis at least in the dimension of tasks and models, hopefully with more data than just the validation sets (and hence with more computational power). As a general learned lesson we may stress the fact the prompt is the crucial hyper-parameter for zero shot inference, and at the state of the art it still needs to be "luckily" hand-tuned to better adapt to the family of question we want our model to address.

Finally, for what concerns the results with the two compared versions of FLAN, we observed that comparing the mean of average results on the two tasks for each prompt, the ones "generally" behaving well for the *small* version, keep behaving well with the *base* one, which is something reasonable but not trivial at all and quite unexpected to be discovered. Thinking posteriorly,

we can address such behavior to the fact the FLAN-small was fine-tuned on a sub-sampled version of the training set used for FLAN-base, and hence, the knowledge of this last version subsumes the knowledge of the tiny one.

6.2 About Task difficulty

We identified some dataset to be slightly harder than the other ones (*anli* for NLI and $RACE$ for QA). Such consideration was based on average performance of the models and mean inference time on them. Trivial correlation was found with poor performance and high query processing time, confirming that the inferences yielding more information are also the most difficult ones to guess. These results allowed us to interpret some unexpected poor performance of FLAN-small as a lack of "deep" reasoning capability. The "shallowness" of such version made it non-competitive with its predecessors on "deeper" tasks. Hence we can observe that "fine-tuned" and "transferable" knowledge are not always sufficient if not supported by a properly "thick" feature extractor (low layers of a deep model), as confirmed by the results obtained with the *base* version.

6.3 Overall evolution trend of Zero shot learning with LLMs

The results clearly depicts an overall sight of the last years' improvement in zero shot learning, starting with the weak GPT2, evolving with the more robust, but slower BART, to finally arrive to the "agile" and wiser FLAN. Improvement in task accuracy should be related not only to the number of parameters but also to the progress in training algorithm and physical infrastructure to handle such large scale optimizations. Progresses in knowledge transfer such as [1] or CoT [15] finally helped to keep knowledge acquired by a large plethora of data and tasks decreasing the number of actual parameters and so the inference time (FLAN). Lastly, also the increase in the number of available high quality NLP dataset should be taken into account in such high level summary of zero shot capabilities of LLMs.

Model	NLI accuracy	QA accuracy	NLI query-time	QA query time	Parameters
GPT-2	0.405	0.446	22.0466	23.241	117M
BART	0.421	0.552	164.734	180.132	407M
FLAN-small	0.615	0.51	8.149	8.133	80M
FLAN-base	0.706	0.66	26.579	43.427	250M

Table 5: **Recap of obtained test results**: average accuracy, average query time and average prompt selection variance on each task (aggregating over datasets); In last columns the number of trainable parameters of each model is reported from original papers

7 Conclusions and Research Directions

Our journey through the world of zero shot inference ends up with several learned lessons. Firstly, we observed that capability of models to understand and properly process unseen prob-

lems strictly depends on the way the question is posed: similar prompts, on the long term (as amount of question grows) have no guarantee at all to reach the same results. Such intrinsic source of variance and uncertainty anyway seems to be intended to continue decreasing as long the complexity and the size of the models grows. Improvement in transfer learning and fine tuning are currently helping (like in the case of FLAN) to mitigate such a need of over-parametrization, at least for user-side inference, but as already claimed by many AI experts, such trend will soon stop to be sustainable. The first sign of alarm were already spotted in this project, when dealing with the long processing time of a moderately large model as BART, without an appropriate gain in precision with respect to such a paid cost. An interesting direction for researches may be represented by large randomized models, like deep ESN, providing common feature extraction backbones and tunable "terminal parts" to adapt to different tasks. Such an approach may help to achieve robust prompt invariance (due to the shared reservoir) and a cheaper computational cost. Another feasible alternative is represented by Continual Learning, which perfectly match with the zero-shot setup, allowing model to not only answer, but to keeping some of the learned new knowledge to progressively improve itself, stopping the trend of companies of designing and training new models from scratch on enormous text corpora.

One good news emerged by this report, is the evidence that prompt correlation can hold within versions of the same model varying in sizes, and hence, in more focused works, it could be investigated whether prompt selection can be carried on with a smaller version of the target model, saving time and FLOPs.

Going back to the present, to conclude, zero shot inference represents a comfortable way for end user to deal with "general purpose" AI model, exploiting implicit adaptation capability intrinsic of the language comprehension skills acquired by Large Language model. As the recent success of ChatGPT is confirming, the more LLMs improves, and larger are the data corpus they are trained on, the more their zero-shot plasticity improves, even on totally unseen tasks. If the research goes on in a sustainable way, we, as the human race, will keep on relying on accurate and adaptable models/experts, with no need to teach them anything ex-novo.

Appendix A Detailed Results

A.1 RESULTS ON NLI

The line of the following tables refers to prompts presented in the previous section. First task is the binary NLI one. hence prompts are different and lower in number. For the remaining tables refers to the ternary NLI and to a larger list of prompts.

A.1.1 RESULTS ON BINARY NLI: *scitail*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.406	0.531	0.813	**0.938**	0.672	0.245
Prompt 2	0.406	**0.656**	**0.843**	**0.938**	0.711	0.234
Prompt 3	0.375	0.5	0.781	0.906	0.640	0.245
Prompt 4	**0.5**	0.5	0.81	**0.938**	0.687	0.222
Prompt 5	**0.531**	0.344	0.781	**0.938**	0.649	0.263
Prompt 6	0.406	**0.563**	0.813	**0.938**	0.680	0.240
Prompt 7	**0.469**	0.5	**0.844**	**0.969**	0.696	0.249
Prompt 8	0.375	**0.563**	0.813	0.906	0.664	0.241
Prompt 9	0.344	**0.594**	**0.844**	0.906	0.673	0.256
Prompt 10	0.437	0.469	**0.844**	**0.938**	0.672	0.256
Mean	0.425	0.522	0.819	0.9315	0.674	-
St.dev.	0.059	0.083	0.025	0.020	-	0.216

Table 6: Binary NLI Prompt Selection Results on *scitail* (32 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	0.525 - *(3)*	0.540 - *(1)*	0.670 - *(7)*	**0.811** - *(1)*
2^{nd} Best Prompt	**0.535** - *(4)*	0.525 - *(5)*	**0.707** - *(12)*	0.801 - *(2)*
3^{rd} Best Prompt	0.487 - *(6)*	**0.545** - *(8)*	0.694 - *(16)*	0.804 - *(7)*

Table 7: Binary NLI TEST Results on *scitail* (1304 samples).

A.1.2 RESULTS ON NLI: *anli*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.333	0.271	0.354	0.333	0.323	0.036
Prompt 2	0.354	0.333	0.333	0.333	0.383	0.011
Prompt 3	0.333	**0.458**	0.333	0.333	0.364	0.063
Prompt 4	0.333	0.271	0.375	0.333	0.364	0.063
Prompt 5	0.354	0.292	0.333	0.333	0.328	0.026
Prompt 6	0.354	0.354	0.583	0.708	0.499	0.177
Prompt 7	0.313	0.229	**0.688**	0.688	0.480	0.243
Prompt 8	0.333	0.208	0.479	0.438	0.349	0.105
Prompt 9	0.292	0.354	0.604	0.75	0.5	0.214
Prompt 10	**0.396**	0.333	0.604	0.792	0.531	0.209
Prompt 11	0.313	0.354	0.646	0.729	0.510	0.208
Prompt 12	0.354	**0.417**	**0.667**	0.75	0.547	0.191
Prompt 13	0.313	0.333	0.542	0.75	0.485	0.205
Prompt 14	0.271	0.333	**0.667**	**0.833**	0.526	0.269
Prompt 15	0.354	0.291	0.646	0.813	0.526	0.246
Prompt 16	0.25	0.354	**0.688**	**0.833**	0.53	0.275
Prompt 17	0.333	0.375	0.646	0.813	0.542	0.228
Prompt 18	0.271	0.3333	0.604	0.813	0.505	0.251
Prompt 19	**0.417**	0.292	0.542	0.75	0.5	0.195
Prompt 20	0.333	0.333	0.604	0.792	0.516	0.224
Prompt 21	0.313	0.354	**0.666**	0.792	0.53	0.235
Prompt 22	**0.396**	**0.458**	0.604	**0.833**	0.573	0.194
Prompt 23	0.313	0.333	0.625	0.813	0.521	0.241
Mean	0.325	0.346	0.629	0.8085	0.472	-
St.dev.	0.054	0.047	0.043	0.026	-	0.190

Table 8: NLI prompt selection on *anli* (48 samples).

A.1.3 RESULTS ON NLI: *glue-mnli*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.333	0.333	0.333	0.333	0.333	0
Prompt 2	0.333	0.417	0.313	0.333	0.349	0.0456
Prompt 3	0.313	**0.458**	0.333	0.333	0.359	0.067
Prompt 4	0.313	0.417	0.271	0.435	0.359	0.080
Prompt 5	**0.396**	0.333	0.333	0.333	0.348	0.032
Prompt 6	0.313	0.333	0.667	0.813	0.532	0.248
Prompt 7	0.313	0.375	0.604	0.854	0.537	0.246
Prompt 8	0.354	0.297	0.458	0.5	0.402	0.093
Prompt 9	0.333	0.25	0.688	0.875	0.537	0.295
Prompt 10	0.333	**0.438**	**0.75**	**0.916**	0.609	0.270
Prompt 11	0.354	0.375	**0.729**	0.875	0.583	0.260
Prompt 12	0.292	0.417	**0.729**	**0.916**	0.589	0.285
Prompt 13	0.354	**0.458**	0.563	0.813	0.547	0.197
Prompt 14	0.271	0.333	0.646	0.896	0.537	0.291
Prompt 15	0.333	0.354	0.708	0.875	0.568	0.268
Prompt 16	**0.5**	0.417	0.667	0.896	0.62	0.211
Prompt 17	0.271	0.333	0.708	**0.917**	0.557	0.308
Prompt 18	0.354	0.354	0.583	0.875	0.542	0.247
Prompt 19	0.333	0.292	0.542	0.833	0.5	0.247
Prompt 20	**0.396**	**0.438**	0.604	0.771	0.552	0.171
Prompt 21	0.271	0.417	0.625	0.875	0.547	0.263
Prompt 22	0.333	0.354	0.604	0.854	0.536	0.246
Prompt 23	0.208	0.333	0.604	0.875	0.505	0.297
Mean	0.327	0.362	0.629	0.867	0.502	-
St.dev.	0.081	0.047	0.054	0.041	-	0.215

Table 9: NLI prompt selection on *glue-mnli* (48 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	**0.348** - *(5)*	**0.362** - *(3)*	**0.64** - *(10)*	0.801 - *(10)*
2^{nd} Best Prompt	0.342 - *(16)*	0.351 - *(13)*	0.63 - *(11)*	0.790 - *(12)*
3^{rd} Best Prompt	0.36 - *(20)*	0.345 - *(20)*	0.636 - *(12)*	**0.804** - *(17)*

Table 10: NLI TEST Results on *glue-mnli* (1000 samples).

A.1.4 RESULTS ON NLI: *sick*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.333	0.354	0.333	0.333	0.338	0.011
Prompt 2	0.313	**0.604**	0.333	0.521	0.443	0.142
Prompt 3	0.333	0.437	0.333	0.333	0.360	0.052
Prompt 4	0.354	0.354	0.417	0.333	0.365	0.036
Prompt 5	**0.458**	0.375	0.333	0.333	0.344	0.021
Prompt 6	0.354	0.375	**0.792**	0.667	0.547	0.217
Prompt 7	0.333	0.375	**0.813**	0.625	0.537	0.225
Prompt 8	0.396	0.25	0.479	0.5	0.406	0.113
Prompt 9	0.354	0.395	**0.792**	0.583	0.531	0.201
Prompt 10	0.417	0.416	0.729	0.667	0.557	0.164
Prompt 11	0.333	0.375	0.771	0.646	0.531	0.212
Prompt 12	0.375	**0.5**	**0.792**	0.667	0.584	0.183
Prompt 13	0.333	0.458	0.729	0.625	0.536	0.176
Prompt 14	0.396	0.375	0.625	0.708	0.526	0.166
Prompt 15	**0.438**	0.416	0.646	**0.729**	0.558	0.154
Prompt 16	0.354	0.395	0.583	0.708	0.51	0.165
Prompt 17	0.396	0.416	0.604	**0.75**	0.541	0.167
Prompt 18	0.333	0.333	0.563	0 .708	0.484	0.184
Prompt 19	0.354	**0.5**	0.604	**0.729**	0.547	0.158
Prompt 20	**0.458**	0.333	0.625	0.708	0.531	0.168
Prompt 21	0.354	0.375	0.625	**0.75**	0.526	0.193
Prompt 22	0.375	0.437	0.625	0.708	0.536	0.156
Prompt 23	0.375	0.375	0.5833	**0.729**	0.516	0.173
Mean	0.383	0.396	0.608	0.723	0.493	-
. *St.dev.*	0.040	0.050	0.026	0017	-	0.159

Table 11: NLI prompt selection on *sick* (48 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	0.234 - *(10)*	**0.388** - *(2)*	**0.665** - *(6)*	0.60 - *(15)*
2^{nd} Best Prompt	0.335 - *(15)*	0.228 - *(12)*	0.632 - *(7)*	**0.614** *(17)*
3^{rd} Best Prompt	**0.348** - *(20)*	0.253 - *(19)*	0.645 - *(12)*	0.612 - *(21)*

Table 12: NLI TEST Results on *sick* (495 samples).

A.1.5 RESULTS ON NLI: *superglue-cb*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.0	0.531	0.0	0	0.133	0.266
Prompt 2	**0.438**	**0.562**	0.25	0.0	0.313	0.245
Prompt 3	0.0312	0.343	0.0	0.0	0.094	0.0 167
Prompt 4	**0.469**	**0.593**	0.406	0.313	0.446	0.118
Prompt 5	**0.406**	0.468	0.0	0.0	0.219	0.253
Prompt 6	0.156	0.468	**0.718**	0.844	0.547	0.304
Prompt 7	0.25	0.375	**0.718**	0.719	0.516	0.240
Prompt 8	0.2812	0.531	0.562	0.719	0.523	0.181
Prompt 9	0.0625	0.406	**0.718**	0.75	0.484	0.321
Prompt 10	0.25	0.343	0.687	0.656	0.484	0.220
Prompt 11	0.063	0.406	0.687	0.781	0.484	0.323
Prompt 12	0.125	0.5	**0.718**	0.75	0.523	0.288
Prompt 13	0.094	0.531	0.406	**0.938**	0.492	0.349
Prompt 14	**0.406**	0.437	0.593	**0.906**	0.586	0.229
Prompt 15	0.313	0.5	**0.718**	**0.906**	0.610	0.258
Prompt 16	0.406	0.468	**0.75**	**0.906**	0.6325	0.236
Prompt 17	0.219	0.531	**0.75**	**0.906**	0.602	0.298
Prompt 18	0.156	0.5	0.656	**0.906**	0.555	0.314
Prompt 19	**0.406**	0.25	0.656	0.875	0.547	0.275
Prompt 20	0.313	0.531	0.687	0.344	0.469	0.175
Prompt 21	0.283	**0.687**	0.656	**0.906**	0.633	0.259
Prompt 22	0.031	0.531	**0.718**	**0.906**	0.547	0.376
Prompt 23	0.125	0.375	0.656	0.813	0.492	0.305
Mean	0.266	0.481	0.684	0.837	0.475	-
St.dev.	0.130	0.114	0.050	0.176	-	0.275

Table 13: NLI prompt selection on *superglue-cb* (32 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	**0.464** - *(2)*	**0.464** - *(2)*	0.643 - *(16)*	0.804 - *(13)*
2^{nd} Best Prompt	**0.464** - *(4)*	0.429 - *(4)*	0.607 - *(17)*	**0.839** - *(16)*
3^{rd} Best Prompt	0.321 - *(19)*	0.411 - *(21)*	**0.679** - *(22)*	0.821 - *(17)*

Table 14: NLI TEST Results o *superglue-cb* (56 samples).

A.2 RESULTS ON QA

The lines of the following tables refers to prompts presented in the previous section.

A.2.1 RESULTS ON QA: *QuaRel*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev*
Prompt 1	0.375	**0.531**	0.406	**0.5**	0.453	0.074
Prompt 2	0.375	**0.531**	0.469	0.375	0.438	0.077
Prompt 3	0.375	0.5	**0.5**	0.375	0.438	0.072
Prompt 4	**0.406**	0.5	0.469	**0.469**	0.461	0.04
Prompt 5	0.375	0.469	**0.5**	**0.469**	0.453	0.054
Prompt 6	0.375	0.469	0.375	0.313	0.383	0.064
Prompt 7	0.344	**0.531**	0.375	0.438	0.422	0.083
Prompt 8	0.375	0.5	0.188	0.344	0.352	0.128
Prompt 9	0.375	0.469	0.375	0.438	0.414	0.047
Prompt 10	0.375	**0.562**	0.469	0.406	0.453	0.083
Prompt 11	**0.406**	**0.562**	0.438	0.344	0.438	0.092
Prompt 12	**0.406**	**0.531**	0.406	0.438	0.445	0.059
Prompt 13	0.375	0.5	0.469	0.375	0.43	0.065
Prompt 14	0.375	**0.531**	**0.5**	**0.469**	0.469	0.067
Prompt 15	0.375	**0.531**	0.469	0.344	0.43	0.086
Mean	0.379	0.514	0.427	0.406	0.432	-
St.dev.	0.016	0.031	0.081	0.058	-	0.072

Table 15: QA Prompt Selection Results on *QuaRel* (32 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	0.5 - *(4)*	**0.514** - *(7)*	0.492 - *(3)*	**0.651** - *(1)*
2^{nd} Best Prompt	**0.504** - *(11)*	0.493 - *(10)*	**0.514** - *(5)*	0.637 - *(5)*
3^{rd} Best Prompt	0.5 - *(12)*	0.468 - *(11)*	0.414 - *(14)*	0.558 - *(14)*

Table 16: QA TEST Results on *QuaRel* (1034 samples).

A.2.2 RESULTS ON QA: *QuaRTz-no_knowledge*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.375	0.531	0.406	**0.719**	0.508	0.156
Prompt 2	0.375	**0.625**	**0.563**	**0.719**	0.571	0.145
Prompt 3	0.375	**0.688**	**0.531**	**0.781**	0.594	0.179
Prompt 4	0.375	**0.625**	**0.531**	**0.781**	0.578	0.17
Prompt 5	0.375	0.531	**0.531**	0.75	0.547	0.154
Prompt 6	0.375	0.469	0.5	**0.719**	0.516	0.146
Prompt 7	0.344	**0.625**	**0.594**	0.688	0.563	0.151
Prompt 8	0.375	0.563	**0.531**	0.656	0.531	0.117
Prompt 9	0.375	0.594	**0.531**	**0.719**	0.554	0.143
Prompt 10	0.375	0.531	0.406	0.688	0.5	0.142
Prompt 11	0.375	0.531	0.344	0.656	0.47	0.155
Prompt 12	0.375	0.469	0.313	0.656	0.453	0.15
Prompt 13	0.375	0.563	0.375	0.625	0.485	0.128
Prompt 14	0.375	0.594	0.531	0.625	0.531	0.111
Prompt 15	0.375	0.5	0.406	0.594	0.469	0.099
Mean	0.373	0.563	0.473	0.692	0.524	-
St.dev.	0.008	0.063	0.088	0.056	-	0.133

Table 17: QA Prompt Selection Results on *QuaRTz-no_knowledge* (32 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	**0.503** - *(1)*	0.539 - *(3)*	**0.544** - *(2)*	0.651 - *(2)*
2^{nd} Best Prompt	0.497 - *(5)*	0.544 - *(4)*	0.5 - *(3)*	0.638 - *(3)*
3^{rd} Best Prompt	0.5 - *(10)*	**0.555** - *(7)*	0.458 - *(7)*	**0.656** - *(4)*

Table 18: QA TEST Results on *QuaRTz-no_knowledge* (384 samples).

A.2.3 RESULTS ON QA: *QuaRTz-with_knowledge*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.375	0.469	0.469	**0.875**	0.547	0.223
Prompt 2	0.375	0.531	**0.531**	0.844	0.57	0.197
Prompt 3	0.375	0.531	**0.531**	0.813	0.563	0.182
Prompt 4	0.375	0.469	0.5	0.843	0.547	0.205
Prompt 5	0.375	**0.563**	**0.531**	0.813	0.571	0.181
Prompt 6	0.375	0.469	**0.563**	**0.875**	0.571	0.217
Prompt 7	0.375	**0.563**	**0.531**	0.781	0.563	0.167
Prompt 8	0.375	0.531	0.469	0.656	0.508	0.118
Prompt 9	0.375	0.531	0.5	0.813	0.555	0.185
Prompt 10	0.375	0.531	0.375	0.813	0.524	0.207
Prompt 11	0.375	**0.563**	0.344	0.656	0.485	0.15
Prompt 12	0.375	**0.563**	0.375	0.594	0.477	0.118
Prompt 13	0.375	**0.563**	0.25	0.688	0.469	0.195
Prompt 14	0.375	**0.594**	0.25	0.688	0.477	0.2
Prompt 15	0.375	**0.688**	0.406	0.625	0.523	0.156
Mean	0.375	0.544	0.442	0.758	0.53	-
St.dev.	0	0.055	0.103	0.096	-	0.164

Table 19: QA Prompt Selection Results on *QuaRTz-with_knowledge* (32 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	**0.5** - *(7)*	**0.550** - *(11)*	0.513 - *(1)*	0.7 - *(1)*
2^{nd} Best Prompt	0.497 - *(10)*	0.552 - *(13)*	0.503 - *(2)*	**0.711** - *(2)*
3^{rd} Best Prompt	0.4974 - *(14)*	0.539 - *(14)*	**0.529** - *(5)*	0.698 - *(4)*

Table 20: QA TEST Results on *QuaRTz-with_knowledge* (384 samples).

A.2.4 RESULTS ON QA: *RACE-middle*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.219	0.406	0.469	0.25	0.336	0.121
Prompt 2	0.219	0.406	**0.531**	0.406	0.391	0.129
Prompt 3	**0.25**	0.438	0.5	**0.563**	0.438	0.135
Prompt 4	0.219	0.469	**0.531**	0.344	0.391	0.138
Prompt 5	**0.25**	0.469	0.5	0.344	0.391	0.116
Prompt 6	**0.313**	0.406	0.438	**0.688**	0.461	0.16
Prompt 7	0.188	0.469	**0.531**	0.406	0.414	0.154
Prompt 8	0.219	**0.5**	**0.531**	**0.438**	0.422	0.141
Prompt 9	0.188	0.469	0.5	**0.438**	0.399	0.143
Prompt 10	0.219	0.469	0.344	0.344	0.344	0.102
Prompt 11	**0.25**	**0.531**	0.25	0.406	0.36	0.136
Prompt 12	0.219	**0.563**	0.219	0.281	0.321	0.164
Prompt 13	**0.25**	**0.5**	0.281	0.25	0.32	0.121
Prompt 14	**0.25**	0.469	0.406	0.375	0.375	0.092
Prompt 15	**0.25**	0.438	0.313	0.344	0.336	0.078
Mean	0.234	0.467	0.423	0.396	0.38	-
St.dev.	0.031	0.045	0.112	0.116	-	0.122

Table 21: QA Prompt Selection Results on *RACE-middle* (32 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	0.242 - *(5)*	0.413 - *(10)*	0.368 - *(4)*	0.462 - *(3)*
2^{nd} Best Prompt	0.251 - *(10)*	**0.414** -*(12)*	**0.380** - *(7)*	**0.582** - *(6)*
3^{rd} Best Prompt	**0.257** - *(13)*	0.409 - *(14)*	0.373 - *(8)*	0.380 - *(8)*

Table 22: QA TEST Results on *RACE-middle* (1436 samples).

A.2.5 RESULTS ON QA: *SciQ*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.438	**0.781**	0.813	0.875	0.727	0.196
Prompt 2	0.438	**0.781**	0.813	**0.938**	0.743	0.214
Prompt 3	0.406	0.75	**0.875**	**0.938**	0.742	0.237
Prompt 4	0.438	0.75	0.781	**0.938**	0.727	0.209
Prompt 5	0.406	0.719	0.813	0.906	0.711	0.217
Prompt 6	**0.469**	0.563	0.813	**0.938**	0.696	0.217
Prompt 7	0.438	0.719	**0.906**	0.344	0.602	0.258
Prompt 8	0.313	0.719	0.813	0.531	0.594	0.221
Prompt 9	**0.469**	**0.781**	**0.875**	0.531	0.664	0.195
Prompt 10	0.406	**0.813**	0.594	0.906	0.68	0.225
Prompt 11	0.313	0.719	0.563	**0.938**	0.633	0.263
Prompt 12	0.344	**0.781**	0.719	0.875	0.68	0.233
Prompt 13	0.406	**0.813**	0.594	0.844	0.664	0.205
Prompt 14	0.375	0.75	0.594	0.844	0.641	0.205
Prompt 15	**0.469**	**0.781**	0.625	0.906	0.695	0.19
Mean	0.409	0.748	0.746	0.817	0.68	-
St.dev.	0.052	0.061	0.12	0.188	-	0.198

Table 23: QA Prompt Selection Results on *SciQ* (32 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	**0.409** - *(6)*	0.809 - *(10)*	0.804 - *(3)*	0.918 - *(2)*
2^{nd} Best Prompt	0.359 - *(9)*	**0.822** - *(12)*	0.812 - *(7)*	**0.945** - *(3)*
3^{rd} Best Prompt	0.384 - *(15)*	0.817 - *(13)*	**0.815** - *(9)*	0.910 - *(4)*

Table 24: QA TEST Results on *SciQ* (887 samples).

A.2.6 RESULTS ON QA: *Social-IQA*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	**0.531**	**0.438**	0.438	0.719	0.532	0.132
Prompt 2	**0.531**	0.406	**0.531**	0.75	0.555	0.143
Prompt 3	0.5	**0.469**	0.406	**0.781**	0.539	0.166
Prompt 4	0.469	0.406	0.438	0.75	0.516	0.158
Prompt 5	**0.531**	**0.469**	0.5	0.594	0.524	0.0534
Prompt 6	0.438	0.344	0.375	0.719	0.469	0.171
Prompt 7	0.5	0.313	**0.469**	0.719	0.5	0.167
Prompt 8	0.469	0.344	0.406	**0.781**	0.5	0.194
Prompt 9	0.5	0.375	**0.469**	**0.781**	0.531	0.175
Prompt 10	0.406	0.344	0.25	0.688	0.422	0.189
Prompt 11	0.438	0.281	0.281	0.5	0.375	0.111
Prompt 12	0.438	0.344	0.188	0.438	0.352	0.118
Prompt 13	0.438	0.313	0.25	0.469	0.368	0.103
Prompt 14	0.5	0.344	0.313	0.625	0.446	0.145
Prompt 15	**0.563**	0.344	0.219	0.5	0.407	0.155
Mean	0.483	0.369	0.369	0.654	0.469	–
St.dev.	0.045	0.057	0.11	0.123	-	0.147

Table 25: QA Prompt Selection Results on $Social - IQA$ (32 samples).

Prompt	GPT-2 (acc - idx_{prpt})	BART (acc - idx_{prpt})	FLAN-small (acc - idx_{prpt})	FLAN-base (acc - idx_{prpt})
1^{st} Best Prompt	0.357 - *(1)*	**0.392** - *(1)*	**0.420** - *(2)*	**0.606** - *(3)*
2^{nd} Best Prompt	0.360 - *(5)*	0.391 - *(3)*	0.392 - *(5)*	0.573 - *(8)*
3^{rd} Best Prompt	**0.361** - *(15)*	0.378 - *(5)*	0.375 - *(9)*	0.593 - *(9)*

Table 26: QA TEST Results on $Social - IQA$ (1954 samples).

A.2.7 RESULTS ON QA: *SuperGLUE-COPA*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.438	**0.594**	0.219	**0.688**	0.485	0.205
Prompt 2	0.438	**0.563**	0.281	0.75	0.508	0.198
Prompt 3	0.438	**0.5**	**0.438**	0.531	0.477	0.047
Prompt 4	0.438	0.469	**0.438**	**0.563**	0.477	0.059
Prompt 5	**0.469**	**0.5**	0.281	**0.656**	0.609	0.157
Prompt 6	0.406	**0.5**	**0.563**	0.344	0.453	0.098
Prompt 7	0.406	0.344	0.438	0.219	0.352	0.097
Prompt 8	**0.469**	0.375	0.344	0.09	0.32	0.162
Prompt 9	0.406	0.469	**0.469**	0.344	0.422	0.06
Prompt 10	**0.469**	0.469	0.125	0.156	0.305	0.19
Prompt 11	**0.469**	0.438	0.156	0.09	0.288	0.193
Prompt 12	**0.469**	0.438	0.094	0.188	0.298	0.185
Prompt 13	**0.5**	0.438	0.219	0	0.29	0.227
Prompt 14	**0.469**	0.469	0.125	0.03	0.273	0.229
Prompt 15	**0.469**	0.406	0.063	0.219	0.29	0.184
Mean	0.45	0.465	0.284	0.325	0.39	-
St.dev.	0.029	0.065	0.157	0.253	-	0.177

Table 27: QA Prompt Selection Results on *SuperGLUE-COPA* (32 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	**0.51** - *(8)*	0.63 - *(1)*	0.33 - *(3)*	**0.58** - *(1)*
2^{nd} Best Prompt	0.44 - *(10)*	**0.66** - *(2)*	**0.37** - *(6)*	0.47 - *(4)*
3^{rd} Best Prompt	0.45 - *(13)*	0.59 - *(5)*	0.32 - *(9)*	0.55 - *(5)*

Table 28: QA TEST Results on *SuperGLUE-COPA* (100 samples).

A.2.8 RESULTS ON QA: *Wino-Grande*

Prompt	GPT-2	BART	FLAN-small	FLAN-base	*Mean*	*St.dev.*
Prompt 1	0.531	0.438	**0.438**	0.625	0.434	0.208
Prompt 2	0.5	0.438	0.344	0.625	0.476	0.119
Prompt 3	0.531	**0.5**	0.313	0.625	0.492	0.131
Prompt 4	0.531	**0.5**	0.344	0.594	0.492	0.106
Prompt 5	0.531	0.469	0.344	0.625	0.485	0.131
Prompt 6	0.5	**0.531**	0.375	0.656	0.516	0.115
Prompt 7	0.531	0.406	0.344	0.594	0.468	0.115
Prompt 8	0.5	0.469	0.313	**0.688**	0.493	0.154
Prompt 9	0.531	0.406	0.344	0.625	0.477	0.126
Prompt 10	**0.563**	**0.5**	0.438	0.625	0.532	0.081
Prompt 11	0.531	0.438	**0.469**	0.656	0.524	0.096
Prompt 12	**0.563**	0.438	**0.469**	**0.688**	0.54	0.112
Prompt 13	0.5	**0.531**	0.406	0.656	0.523	0.103
Prompt 14	**0.563**	0.469	0.375	**0.688**	0.524	0.134
Prompt 15	0.5	0.406	0.375	0.594	0.469	0.099
Mean	0.527	0.463	0.379	0.638	0.496	-
St.dev.	0.023	0.043	0.053	0.033	-	0.113

Table 29: QA Prompt Selection Results on *Wino-Grande* (32 samples).

Prompt	GPT-2 (*acc* - idx_{prpt})	BART (*acc* - idx_{prpt})	FLAN-small (*acc* - idx_{prpt})	FLAN-base (*acc* - idx_{prpt})
1^{st} Best Prompt	**0.510** - *(10)*	0.504 - *(6)*	**0.523** - *(1)*	**0.555** - *(8)*
2^{nd} Best Prompt	0.507 - *(12)*	**0.506** - *(10)*	0.497 - *(11)*	0.518 - *(12)*
3^{rd} Best Prompt	0.508 - *(14)*	0.497 - *(13)*	0.502 - *(12)*	0.516 - *(14)*

Table 30: QA TEST Results on *Wino-Grande* (1267 samples).

Appendix B Prompts

B.1 NLI: binary prompts

Available options are: [*entailment*, *notentailment*]

```
1. "Premise:  CONTEXT Hypothesis:  HYPOTHESIS Does the premise entails the hypothesis?
```

OPTIONS"

2. "Premise: CONTEXT Hypothesis: HYPOTHESIS Is the hypothesis entailed by the premise? OPTIONS"

3. "Here is a premise: CONTEXT Here is a hypothesis: HYPOTHESIS Is it possible to conclude that if the premise is true, then so is the hypothesis? OPTIONS"

4. "Sentence 1: CONTEXT Sentence 2: HYPOTHESIS Is this second sentence entailed by the first sentence? OPTIONS"

5. "Sentence 1: CONTEXT Sentence 2: HYPOTHESIS If the first sentence is true, then is the second sentence true? OPTIONS"

6. "Based on the premise CONTEXT, can we conclude the hypothesis HYPOTHESIS is true? OPTIONS"

7. "Premise: CONTEXT If this premise is true, what does that tell us about whether it entails the hypothesis HYPOTHESIS? OPTIONS"

8. "Premise: CONTEXT Based on this premise, is the hypothesis HYPOTHESIS true? OPTIONS"

9. "If CONTEXT, can we conclude that HYPOTHESIS? OPTIONS"

10. "CONTEXT Does it follow that HYPOTHESIS? OPTIONS"

B.2 NLI: ternary prompts

Available options are: [*entailment*, *neutral*, *contradiction*]

1. "Choose the correct label among: OPTIONS for the following Natural Language Inference task: SENTENCE"

2. "Read the following sentence. sentence: SENTENCE. Is it an 'entailment', a 'contradiction' or is it 'neutral'?"

3. "Assign one of the following labels: OPTIONS to the following sentence S: SENTENCE"

4. "The following assertion: 'SENTENCE' is OPTIONS?"

5. "A sentence can be of one type among: OPTIONS. Which type of sentence is [SENTENCE]?"

6. CONTEXT Based on the paragraph above can we conclude that the hypothesis: 'HYPOTHESIS' is more likely to be a kind of OPTIONS? Choose only one of answer among these three."

7. "CONTEXT Based on that paragraph can we conclude that this sentence is true? HYPOTHESIS OPTIONS"

8. "CONTEXT Can we draw the following conclusion? HYPOTHESIS OPTIONS" "CONTEXT Does this next sentence follow, given the preceding text? HYPOTHESIS OPTIONS"

9. "CONTEXT Can we infer the following? HYPOTHESIS OPTIONS", "Read the following paragraph and determine if the hypothesis is true: CONTEXT Hypothesis: HYPOTHESIS OPTIONS"

10. "Read the text and determine if the sentence is true: CONTEXT Sentence: HYPOTHESIS OPTIONS"

11. "Can we draw the following hypothesis from the CONTEXT? CONTEXT: CONTEXT Hypothesis: HYPOTHESIS OPTIONS"

12. "Determine if the sentence is true based on the text below: HYPOTHESIS CONTEXT OPTIONS"

13. "Premise: CONTEXT Hypothesis: HYPOTHESIS Does the premise entail the hypothesis? OPTIONS"

14. "Premise: CONTEXT Hypothesis: HYPOTHESIS Is the hypothesis entailed by the premise? OPTIONS"

15. "Here is a premise: CONTEXT Here is a hypothesis: HYPOTHESIS Is it possible to conclude that if the premise is true, then so is the hypothesis? OPTIONS"

16. "Sentence 1: CONTEXT Sentence 2: HYPOTHESIS Is this second sentence entailed by the first sentence? OPTIONS"

17. "Sentence 1: CONTEXT Sentence 2: HYPOTHESIS If the first sentence is true, then is the second sentence true? OPTIONS"

18. "Based on the premise "CONTEXT", can we conclude the hypothesis HYPOTHESIS is true? OPTIONS"

19. "Premise: "CONTEXT" If this premise is true, what does that tell us about whether it entails the hypothesis "HYPOTHESIS"? OPTIONS"

20. "Premise: "CONTEXT" Based on this premise, is the hypothesis "HYPOTHESIS" true? OPTIONS'

21. "If CONTEXT, can we conclude that HYPOTHESIS? OPTIONS"

22. "CONTEXT Does it follow that HYPOTHESIS? OPTIONS"

B.3 QA prompts

The number of options is variable, ranging from two (i.e. $[(A), (B)]$) up to even four ($[(A), (B), (C), (D]$).

1. "For the following Question Answering task choose the correct answer between the given options. Question: QUESTION. Options: OPTIONS"

2. "Given the following Question Answering task, choose an answer between the options. Question: QUESTION. Options: OPTIONS"

3. "Give an answer for the following Question Answering task. Question: QUESTION. Options: OPTIONS"

4. "Which is the correct answer among the options? Question: QUESTION. Options: OPTIONS"

5. "Choose the correct answer among the options. Question: QUESTION. Options: OPTIONS"

6. "Given these options OPTIONS, answer the following question: QUESTION. Answer using only the words present in the options."

7. "Answer the following question:\n\nQUESTION\n\nOptions: OPTIONS"

8. "Answer this question:\n\nQUESTION?\nOptions: OPTIONS"

9. "What is the answer to this question? QUESTION\nOptions: OPTIONS"

10. "What is the answer to the following: QUESTION? OPTIONS"

11. "Provide the answer to: QUESTION. OPTIONS"

12. "Answer the following: QUESTION. OPTIONS"

13. "What is the response to: QUESTION? OPTIONS"

14. "In the following text: QUESTION, what is the answer? OPTIONS"

15. "Given the input QUESTION, provide the answer. OPTIONS"

Appendix C Tasks and Datasets

The datasets we used for our evaluation are associated to the two families of tasks considered: Natural Language Inference (NLI) and Question Answering (QA) tasks. In the following pages we are going to explain in detail properties and choices we made during the dataset selection.

C.1 Natural Language Inference (NLI)

Natural Language Inference is the task of recognizing the relationship within two sentences: a context (or premise) and a hypothesis, which can usually have three type of relationships: be one entailed by the other, being in contradiction, or having no direct connection and hence being neutral one to each other. These relations are not symmetrical (it's particular easy to notice thinking of entailements), and so it's important to be aware of the role of the two input sequences and so of the desired direction of inference.

C.1.1 superglue cb

CommitmentBank [9] is a corpus of short texts in which at least one sentence contains an embedded clause. Each of these embedded clauses is annotated with the degree to which it appears the person who wrote the text is committed to the truth of the clause. The resulting task framed as three-class textual entailment on examples that are drawn from the Wall Street Journal, fiction from the British National Corpus, and Switchboard. Each example consists of a premise containing an embedded clause and the corresponding hypothesis is the extraction of that clause.

```
Premise:  "Polly had to think quickly.  They were still close enough to shore for
him to return her to the police if she admitted she was not an experienced ocean
sailor."

Hypothesis:  "Polly was not an experienced ocean sailor."

label:  not_entailment (neutral)
```

In this case the task is binary NLI, with only two possible options: entailment and not entailement. Validation set was composed of 32 instances while the test set by 56.

C.1.2 sick

The Sentences Involving Compositional Knowledge (SICK) [10] dataset is a dataset for compositional distributional semantics. It includes a large number of sentence pairs that are rich in the lexical, syntactic and semantic phenomena. Each pair of sentences is annotated in two dimensions: relatedness and entailment. The relatedness score ranges from 1 to 5, and Pearson's r is used for evaluation; the entailment relation is categorical, consisting of entailment, contradiction, and neutral.

```
Sentence_A: "A group of children is playing in the house and there is no man standing
in the background."

Sentence_B: "A group of kids is playing in a yard and an old man is standing in
the background."

label_AB: A_contradicts_B"

label_BA: "B_neutral_A"
```

C.1.3 scitail

The SciTail dataset [4] is an entailment dataset created from multiple-choice science exams and web sentences. Each question and the correct answer choice are converted into an assertive statement to form the hypothesis. Information retrieval was used to obtain relevant text from a large text corpus of web sentences, and use these sentences as a premise. Crowdsourcing provided the annotation of such premise-hypothesis pairs as supports (entails) or not (neutral), in order to create the SciTail dataset.

Only two labels: entailment or neutral...

```
Premise: "Facts: Liquid water droplets can be changed into invisible water vapor
through a process called evaporation."

Hypothesis: "Evaporation is responsible for changing liquid water into water vapor."

label: entailment
```

C.1.4 anli

The Adversarial Natural Language Inference [12] is a new large-scale NLI benchmark dataset, collected via an iterative, adversarial human-and-model-in-the-loop procedure. Particularly, the data were selected to be difficult to the state-of-the-art models ($\tilde{2}020$), including BERT and RoBERTa.

```
Premise: "The Centralia Massacre was an incident during the American Civil War
in which twenty-four unarmed Union soldiers were captured and executed at Centralia,
Missouri on September 27, 1864 by the pro-Confederate guerrilla leader William T.
Anderson. Future outlaw Jesse James was among the guerrillas."

Hypothesis: "Jesse James was a guerrilla in the Union army during the American
Civil War."

label: contradiction
```

C.1.5 glue-mnli

General Language Understanding Evaluation (GLUE) [6] benchmark is a collection of nine natural language understanding tasks, including single-sentence tasks CoLA and SST-2, similarity

and paraphrasing tasks MRPC, STS-B and QQP, and natural language inference tasks MNLI, QNLI, RTE and WNLI. In particular, for what concerns the MNLI task, the class are the three considered so far: entailment, contradiction and neutral.

```
Premise:  "Hierbas, ans seco, ans dulce, and frigola are just a few names worth
keeping a look-out for."

Hypothesis:  "Hierbas is a name worth looking out for."

label:  entailment
```

C.2 Question Answering (QA)

Question Answering is the task of answering questions (typically reading comprehension questions).

The QA datasets, in general, are composed of a number of different options (within whom there is the correct answer) and the question that has to be answered. Furthermore, there may be the case that there is a context, i.e. a short story, and the question refers to that brief story. The entire premise can be built in two different ways, either there is the question followed by the options first and then the context, or vice versa, e.g. the context first and the question and options. Summing up, the main properties of the QA datasets we found out are: The number of options; The presence of a context; The order of context, question and answer. Moreover, during the dataset selection process, we found out that some datasets contained special separators, such as [SEP] to separate the question+options from the context, so we added a new property, namely the presence of the [SEP] character. Also, "question: ", "context: ", "options: " were used to highlight that the content present after these separators are that part of the question. After we found out these properties, we decided to manipulate the datasets in a way to give them all an uniformed appearance. The appearance we decided it the following one: `<context> <question> <options>`, removing the "question: ", "context: ", "options: " separators (in those datasets where they are present). In the following, we are going to show more in detail each single dataset we used for the QA task, showing their properties and characteristics.

C.2.1 QuaRel

QuaRel is a dataset of diverse story questions involving qualitative relationships that characterize these challenges, and techniques that begin to address them. As you can notice from the sample example, the properties of this dataset are:

- Number of options: 2
- Presence of a context: No

- Order: Question, answer
- Separators: No

We decided to consider this and similar datasets as "no context", going against its definition, since we can assume that the story is implicitly contained in the question, unlike other datasets we will see.

```
William is ice skating and notices that his ice skates glides quicker on wet ice
as opposed to freshly fallen snow.  The reason for this is because there is more
friction on the (A) wet ice (B) freshly fallen snow
```

C.2.2 QuaRTz

QuaRTz is a dataset for reasoning about textual qualitative relationships and contains general qualitative statements. The properties of this dataset are:

- Number of options: 2
- Presence of a context: No
- Order: Question, answer
- Separators: No

Similarly as the previous dataset, also in this case we considered "presence of a context" as a not "real" context.

```
If Mona is creating an acid and she makes it weaker, what happens to the amount
of hydrogen ions that acid can produce?  (A) increases (B) decreases
```

C.2.3 RACE-middle

RACE-middle is a dataset composed of English questions that target Chinese students of middle school exams. This dataset has some peculiarities, such as the variety in comprehension skills, complexity of answers and multistep reasoning. The properties of this dataset are:

- Number of options: 4
- Presence of a context: Yes
- Order: Question, answer, context

- Separators: Yes

```
What can we learn from the passage? (A) There are three terms in Thailand schools.
(B) Students don't go to school in November. (C) Students usually go home at 3:15
p.m. (D) Students have six classes a day. [SEP] For many schools in Thailand,
there are two terms. The first term is from the first week of May to the first
week of October. The second term starts from the first week of November and finishes
at the last week of February or the first week of March. The students don't get
a _ for Christmas . But they get a 3-4 days' break for the New Year. For many
students, a school day is very long. They usually get to school at 7:30 a.m. Classes
begin at 8:00 a.m. there are three classes in the morning and they are 50 minutes
each . Students have lunch at 11:00 a.m. they don't have dining halls so they
have to eat in the classroom. Lunch time finishes at 12:25 p.m. there are three
classes in the afternoon. School finishes at 3:15 p.m. many schools have a "homework"
lesson after school, so students usually go home after 4:45 p.m.
```

C.2.4 SciQ

SciQ is a dataset composed by science questions; it has a multiple choice version, where the task is to select the correct answer using whatever background information a system can find given a question and several answer options, and a direct answer version, where given a passage and a question a system must predict the span within the passage that answers the question. The properties of this dataset are:

- Number of options: 4
- Presence of a context: Yes
- Order: Question, answer, context
- Separators: Yes

```
How many bases does dna have in total? (A) twelve (B) four (C) three (D) six [SEP]
Every DNA and RNA polymer consists of multiple nucleotides strung together into
extremely long chains. The only variation in each nucleotide is the identity of
the nitrogenous base. The figure above shows one example of a nitrogenous base,
called adenine. There are only five different nitrogenous bases found in all nucleic
acids. The four bases of DNA are adenine, thymine, cytosine, and guanine, abbreviated
A, T, C, and G respectively. In RNA, the base thymine is not found and is instead
replaced by a different base called uracil, abbreviated U. The other three bases
are present in both DNA and RNA.
```

C.2.5 Social IQa

Social IQa is a dataset used for testing social common sense intelligence. Social IQa focuses on reasoning about people's actions and their social implications. The actions in Social IQa span a wide variety of social situations, and answer candidates contain both human-curated answers and adversarially-filtered machine-generated candidates. The properties of this dataset are:

- Number of options: 3
- Presence of a context: Yes
- Order: Question, answer, context
- Separators: Yes

```
Why did Austin do this?  (A) settle down (B) go to the bar (C) order a drink [SEP]
Austin had a rough day at work and decided to go to the bar.  Austin had a drink
that night
```

C.2.6 SuperGLUE COPA

SuperGLUE COPA is a causal reasoning task in which a system is given a premise sentence and must determine either the cause or effect of the premise from two possible choices. The properties of this dataset are:

- Number of options: 2
- Presence of a context: No
- Order: Question, answer
- Separators: No

```
The student was in a rush to get to school on time.  (A) He left his assignment
at home.  (B) He brought his lunch to school.
```

C.2.7 Wino Grande

Wino Grande is a dataset built using a carefully designed crowd sourcing procedure, followed by a systematic bias reduction using a novel AFLITE algorithm that generalizes human-detectable word associations to machine-detectable embedding associations. The properties of this dataset are:

- Number of options: 2
- Presence of a context: No
- Order: Question, answer
- Separators: No

```
The doctor offered to treat the patient's illness quickly with surgery or slowly
with diet, and the patient refused the _ because he felt it was dangerous.  (A)
diet (B) surgery
```

C.3 Datasets summary

Dataset	# samples dev set	# samples test set	Binary task
anli	48	1000	Yes
glue − mnli	48	1000 (~~9815~~)	No
scitail	32	1304	No
sick	48	495	No
superglue − cb	32	56	No

Table 31: Information recap for NLI datasets.

Dataset	# samples dev set	# samples test set	# of options	Context	Order	Separators
QuaRel	32	1034	2	No	QA	No
QuaRTz − no_knowledge	32	384	2	No	QA	No
QuaRTz − with_knowledge	32	384	2	No	QA	No
RACE − middle	32	1436	4	Yes	QAC	Yes
SciQ	32	887	4	Yes	QAC	Yes
SocialIQa	32	1954	3	Yes	QAC	Yes
SuperGLUECOPA	32	100	2	No	QA	No
WinoGrande	32	1267	2	No	QA	No

Table 32: Information recap for QA datasets. In the "Order" column, **Q** stands for *Question*, **A** stands for *Answer*, **C** stands for *Context*.

References

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531 [stat.ML].

[2] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[3] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a84
Paper.pdf.

[4] Tushar Khot, Ashish Sabharwal, and Peter Clark. "SciTaiL: A Textual Entailment Dataset from Science Question Answering". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.12022. URL: https://doi.org/10.1609/aaai.v32i1.12022.

[5] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: (2018). URL: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

[6] Alex Wang et al. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. DOI: 10.18653/v1/W18-5446. URL: https://aclanthology.org/W18-5446.

[7] Rewon Child et al. *Generating Long Sequences with Sparse Transformers*. 2019. arXiv: 1904.10509 [cs.LG].

[8] Mike Lewis et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL].

[9] Alex Wang et al. "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[10] Li Zhang, Steven Wilson, and Rada Mihalcea. "Multi-Label Transfer Learning for Multi-Relational Semantic Similarity". In: *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 44–50. DOI: 10.18653/v1/S19-1005. URL: https://aclanthology.org/S19-1005.

[11] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[12] Xiaodong Liu et al. "The Microsoft Toolkit of Multi-Task Deep Neural Networks for Natural Language Understanding". In: Jan. 2020, pp. 118–126. DOI: 10.18653/v1/2020.acl-demos.16.

[13] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. arXiv: 1910.10683 [cs.LG].

[14] Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. "CrossFit: A Few-shot Learning Challenge for Cross-task Generalization in NLP". In: *ArXiv* abs/2104.08835 (2021).

[15] Hyung Won Chung et al. *Scaling Instruction-Finetuned Language Models*. 2022. arXiv: 2210.11416 [cs.LG].

[16] Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs)*. 2023. arXiv: 1606.08415 [cs.LG].