MSc Artificial Intelligence
Master Thesis

Probabilistic Prompting of Language Models

by
Francesco Tinner
14497425

July 14, 2024

Number of Credits: 48
November 2023 - June 2024

Supervisor:
Dr Wilker Aziz
Examiner:
Dr Sandro Pezzelle
Second reader:
MS Evgenia Ilia

Universiteit van Amsterdam

Abstract

As we rely more on language models (LLMs) in a range of contexts, it is important to determine whether LMs have stored a representation of a fact within their parametric knowledge. Related work tests this by examining the internal representations using a masked language modeling task, or by using prompts to probe for factual knowledge in a question-and-answer format. Both approaches provide no reliable estimation to claim that a LLM has knowledge of a fact or understood the prompt in the intended way, as they either compare outputs only against one reference answer or do not account for the large variety a question or prompt can be formulated. To address this, we frame prompting as a probabilistic inference problem. We use Mistral-7B-Instruct-V0.2 and GPT-2- L's internal sequence probabilities and normalize them against a reference set of possible sequences stating semantically equivalent and closely related, but factually flawed statements, based on facts from LAMA-TRex and PopQA. The probabilities assigned to each of the reference sequences are then used to probe for knowledge using selective prediction. Our analysis across datasets indicates that LLMs are only in 0.8-1.7% of sequences highly confident. At this confidence level, their predictions are between 73 - 98% factually correct.

Contents

Abstract i

Contents i

List of Acronyms iii

1 Introduction 1

2 Background on Language Models 3
2.1 Encoder-only . . . 3
2.2 Decoder-only . . . 4
2.2.1 Decoding Strategies . . . 4
2.3 Encoder-decoder . . . 4

3 Related Work 6
3.1 Methods to Probe LMs for Factual Knowledge . . . 6
3.1.1 Challenges of Prompting . . . 6
3.1.2 Output-based Evaluation of Factual Correctness . . . 7
3.1.3 Factual Correctness Estimated Via Internal Representations . . . 7
3.2 Estimation of a LM's Factual Confidence . . . 8
3.3 Automatic Paraphrase Generation . . . 9

4 Method 10
4.1 Implementation Details . . . 11
4.1.1 Sequence Components: $t(r, s, o)$ . . . 11
4.1.2 Paraphrase Template Generation . . . 11
4.1.3 Generating Factually Wrong Sequences . . . 12
4.1.4 Sequence Probability: $P_{LM}(t(r, s(r), o(r))|c)$ . . . 13
4.1.5 Slot-filling Probability Distribution: $P(O = o(r)|R = r, S = s(r))$ . . . 13
4.1.6 Experimental Methodology . . . 16
4.1.7 Classification . . . 16
4.2 Datasets . . . 20
4.2.1 LAMA T-REx . . . 20
4.2.2 PopQA . . . 21
4.2.3 Hypernym . . . 22

5 Experiment 1: Parametric Knowledge 23
5.1 Predicting Factual Correctness of Sequences . . . 23
5.1.1 Using Argmax O . . . 23
5.1.2 Using $P(o(r)|r, s(r))$ . . . 24
5.2 Evaluating What LMs Think That They Know . . . 25

5.2.1 Per Dataset . . . 25
5.2.2 Per Relation . . . 28
5.2.3 Per Subject . . . 30

6 Experiment 2: Context Understanding 32
6.1 In-context Learning Ability . . . 32
6.1.1 Results at a 0.5 Threshold . . . 32
6.1.2 Threshold-independent Results . . . 33
6.1.3 Results by Relation . . . 35
6.1.4 Changes of Selective Precision on Subject Level . . . 36
6.2 Reading Comprehension (RC) . . . 38
6.2.1 Results . . . 38
6.2.2 Changes on Subject Level . . . 40

7 Discussion 43
7.1 Limitations . . . 43
7.1.1 Qualitative Evaluation of Filled-in Templates . . . 43

8 Conclusion 45

References 47

A Appendix 51
A.1 Dataset Statistics . . . 51
A.1.1 Full Datasets . . . 51
A.1.2 Permutation Stats . . . 53
A.2 Results per Dataset . . . 55
A.2.1 Classification of with Varying Thresholds . . . 55
A.3 Class Separation . . . 55
A.3.1 Ranks Assigned to True and False Sequences . . . 55
A.3.2 Hypernym Class Separation . . . 56
A.4 AUC Scores by Relation . . . 56
A.4.1 Parametric Knowledge . . . 56
A.4.2 In-context Understanding . . . 58
A.5 Full Example for $P(o(r)|r, s(r))$ and Aggregation over Paraphrase Templates . . . 63

List of Acronyms

LLM	Large Language Model
LM	Language Model
NLP	Natural Language Processing
NLI	Natural Language Inference
RC	Reading Comprehension
LAMA	LAnguage Model Analysis
GPT	Generative Pre-trained Transformer
BERT	Bidirectional Encoder Representations from Transformers
MLM	Masked Language Modeling
T5	Text-to-Text Transfer Transformer
QA	Question Answering
OPT	Open Pre-trained Transformer Language Models
API	Application Programming Interface
RAG	Retrieval Augmented Generation
AUPRC	Area Under the Precision-Recall Curve
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
TP	True Positive
TN	True Negative
FN	False Negative
FP	False Positive

Chapter 1

Introduction

Prompting and especially prompt engineering has gained immense popularity with the rise of large language models (LLMs) as it is the only way to interact with such black-box architectures. Depending on a prompt's adequacy, language models (LMs)[1] are enabled to perform more reliably on many tasks. However, there can be a discrepancy between what a prompt means for humans, and what it "means" internally to the LM. LMs could rely on surface-level similarities and not on the semantic meaning a prompt has and thus, it cannot be predicted which phrasing suits the models most.

Therefore, prompting is not well-suited to make claims about a LM's parametric knowledge and whether a LM has a representation of a fact, or linguistic phenomenon, without the comprehensive use of paraphrased prompts. In this setting evaluation becomes the main challenge. For example, in question answering (QA), the open-ended answers generated by the LM need to be semantically matched to the ground truth answer. This involves more than exact or fuzzy matching, as every answer that is semantically equivalent to the true answer needs to be considered as correct, too.

Most datasets create sequences stating facts from triplets of a knowledge base such as WikiData [Vrandečić and Krötzsch, 2014]. Each fact consists of a subject and an object, which are related to each other via a specific relation. E.g., *Joe Biden was born in Scranton*, where *Joe Biden* is the subject, *Scranton* the object and the relation is *born in*.

Related papers make claims about LMs knowing facts without a comprehensive use of paraphrased prompts or reference objects. In this work, we propose a novel method—probabilistic prompting—that address these two shortcomings. We do so by formulating prompting as a probabilistic inference problem. The method applied allows us to investigate more comprehensively whether a LM knows a fact. It requires (1) the creation of equivalence classes that contain realizations of the correct fact and a series of candidate sequences, and (2) the generation of paraphrases in order to approximate the possibility of expressing the same thing in different ways.

To illustrate the creation of (1), the equivalence classes, let us consider the correct fact expressed in the example above. For the subject, *Joe Biden* and the *born-in* relation, possible candidate objects can be all possible towns on earth, e.g. Arlington, Washington. If the LM has a valid representation of our example fact, it assigns a higher probability to the sequence containing *Scranton* than to all other sequences containing the other town candidate objects—in this case we can claim with certainty that the LM knows that Joe Biden was born in Scranton.

The generation of paraphrases, (2), allows us to use various prompts that differ on the surface level, and make our probe more comprehensive without adding evaluation complexities.

[1] We use the terms LLM and LM interchangeably as their main difference is the amount of parameters, and there is no clear dividing line between them, neither in terms of parameters, nor in terms of abilities.

Research Questions

1. Is it possible to distinguish factually correct and incorrect sequences based on the probabilities assigned to the sequences' objects via probabilistic prompting, and how do these predictions compare to top-1 predictions, obtained by ranking objects according to their probabilities?

2. Can we use the probabilities assigned to the reference objects as an indication of factual certainty?

3. Does the inclusion of paraphrased templates improve the probabilities assigned to the objects both as a measure of certainty, and also in the prediction of factual correctness?

4. Can a positive influence be measured on certainty and factual correctness prediction when providing additional context?

Contributions We evaluate our probing method, probabilistic prompting, using two LMs, GPT-2-L (2.5B) [Radford et al., 2019] and Mistral-7B-Instruct-v0.2 [Jiang et al., 2023]. This allows us to compare a LM from an earlier stage of autoregressive LMs with a very recent, and larger LM. The datasets we apply our method on are subsets of the LAMA T-REx PopQA.

This work makes the following three contributions: (1) We release the probabilistic framework in this GitHub repository, (2), propose an evaluation approach using selective prediction which takes into account both factual confidence and correctness. Lastly, (3), we provide an in-detail evaluation of factual knowledge probing results on the levels of the entire dataset, per-relation and per-subject.

Chapter 2

Background on Language Models

LMs are based on the basic idea of next word prediction. Given a sequence of text, a deep neural network based on the transformer architecture, predicts the next subword token, following the sequence. The multi-headed self-attention mechanisms in the transformer architecture allow processing of sequential data in a parallelized way, not relying on recurrent representations and thus long-distance dependencies in these context sequences can be captured better. Using this mechanism the relative importance of the tokens in the sequence can be learned. In current open-source LLMs such as Llama 2, about 10TB of textual data from the internet is compressed to 70 billion of parameters, which require around 150GB to be stored [Touvron et al., 2023]. These parameters were adapted iteratively during pre-training using the next-word-prediction objective. Multiple versions exist, which will be described in the following sections. A second training stage that is needed for LMs such as Mistral-7B-Instruct-v0.2 is instruction fine-tuning. This further improves the pre-trained LMs based on ideal question-answer responses, or labels that compare the quality of two responses [Ouyang et al., 2022].

An important aspect to note is that LMs can generate factually wrong sequences. Consider for example a question-answering system using a LM. The answers of the LM are generated in response to the given prompt (containing system instructions and the question). Due to its pre-training objective, the generated answers can be imperfect and are not constrained to be factually correct, as a LM is an optimization system that selects an ideal output based on the internal probabilities that best correspond to the given context.

2.1 Encoder-only

BERT is an example of a bidirectional, transformer encoder-only LM. It consists of a transformer-encoding module only, which allows processing of the contextual information contained in a sequence and condensing it to a numerical representation. The raw text is tokenized and encoded to a vector representing the semantic information per token, and then in later layers is condensed to a contextualized representation which condenses the whole input sequence to a numerical representation. These representations are optimized during the pre-training objectives of masked language modeling (MLM) and next sentence prediction (where the task is to predict whether two sentences are followed by each other). MLM is a semi-supervised task where the objective is to predict a blanked-out token in a given sequence, based on the context on both sides of the masked word [Devlin et al., 2019]. The logits at a token's position in a sequence of an encoder-only LM correspond to the joint probability distribution over the vocabulary.

After pre-training, the embeddings (the outputs from a encoder-only LM that encodes the sequence to a numerical representation) can be used in downstream tasks such as text classification, natural language inference or named entity recognition. This method is called

fine-tuning. If we wanted to fine-tune a BERT model for sentiment classification, we would train an additional neural network (for example a basic linear layer) that takes the sequence embeddings as input and outputs the sentiment score.

2.2 Decoder-only

Autoregressive LMs, such as GPT (Generative Pre-trained Transformer), consist of a transformer-decoder only, which allows the generation of text sequentially, token by token, based on the tokens in the context sequence. The internal probability per token is a joint probability distribution over the vocabulary.

In contrast to the encoder part, which contains a bidirectional self-attention mechanism, the decoder part contains only a unidirectional self-attention mechanism. For the task of autoregressive generation this makes sense as the next token is not known beforehand and reduces computational cost [Tomczak, 2022; Radford et al., 2018].

2.2.1 Decoding Strategies

Given a sequence of subword units, the next token is predicted by an autoregressive model using the predicted logit distribution over its vocabulary to select the next, most likely token. There are multiple strategies that can be used to make this selection [Shi et al., 2024].

Greedy search selects the token from the logit distribution that has the highest value. This is done for each time step by repeatedly sampling the token with the highest value, considering the available context to the left. Sampling stops once the maximum length is reached, or an end of sequence token is sampled [Shi et al., 2024].

Beam search considers multiple alternative next-tokens based on the highest token probability, instead of repeated greedy sampling which does not optimize further than the next token. At each time step, this allows us to select the beam hypothesis with the highest overall score by planning ahead. Downsides of this method are that the different beams often contain only small modifications. Also, the length of the output is constrained, which makes beam search not suitable for open-ended generation [Shi et al., 2024].

Top-k sampling creates more diverse outputs by considering the k most likely next tokens and then selecting one of these randomly. The temperature parameter sets how many tokens are considered to sample from. The more tokens are considered, the more creativeness is added to the generated sequences. Setting the parameter temperature to 0 uses only the top-1 token and thus corresponds to greedy search. A temperature parameter of 1 allows sampling from the whole vocabulary distribution, maximizing the diversity of the decoded sequences [Shi et al., 2024].

Nucleus sampling uses a dynamic size of tokens that can be sampled. This set is created by including the most probable tokens until its probability mass reaches a cutoff point [Shi et al., 2024].

2.3 Encoder-decoder

An example of an encoder-decoder based LM is T5 (Text-to-Text Transfer Transformer). For this architecture, the logits per token correspond to the conditional probability over the vocab-

ulary, that is conditioned on the context. T5's pretraining includes the standard self-supervised tasks of predicting masked-out tokens but furthermore, task-dependent, supervised training on downstream tasks such as natural language inference (NLI) is added. These downstream tasks are framed as sequence to sequence tasks to be compatible with the LM directly. This supervised training makes T5 well-suited for tasks that require both sequence understanding and generating an output conditioned on the input. This is helpful for sequence-to-sequence tasks such as machine translation, NLI, and summarization. Furthermore, the encoder component allows to extract and condition the decoder on information extracted from other modalities than text [Raffel et al., 2023].

Chapter 3

Related Work

3.1 Methods to Probe LMs for Factual Knowledge

According to [Youssef et al., 2023] there exist 33 different probing methods to evaluate the factual knowledge stored in the parameters of LMs. They can be categorized into three categories, depending on where modifications occur: (1) in the input space, by prompt design, (2) on the LM itself, by either adapting their parameters or using their internal representations, or (3) on the output space, either through restriction, or debiasing. We focus on (1) the input space and (2) using the LM's internal probabilities to normalize with respect to a selection of candidate sequences.

3.1.1 Challenges of Prompting

A prompt serves as a starting point for an autoregressive LM to generate a sequence that addresses the instructions described in the prompt. Due to its training objective, the LM generates the most likely sequence that follows the sequence of tokens in the prompt. As there are multiple ways of expressing instructions, and not all of them generate the same sequences, prompting provides only a lower-bound estimation of the amount of factual knowledge stored in the LM's parameters. It can be that knowledge can only be accessed from a certain viewpoint or by using the right phrasing, as LMs are sensitive to the exact wording used in prompts. Lexically distinct phrased prompts can have different meanings to the model internally. Therefore it is challenging to say when we know for certain that a LM knows a fact. Certainly, it is not sufficient to only prompt the model once, and then make deductions about the presence or absence of knowledge regarding the fact in the prompt. To address this, other work such as uses paraphrased prompts to verify the LM's factual knowledge is semantically consistent [Jiang et al., 2021; Elazar et al., 2021; Gonen et al., 2023]. We review some of this work in Section 3.1.3.

Automatically Optimized Prompts [Shin et al., 2020] construct a method that creates optimized prompts for a specific task and masked LM. Their method works without fine-tuning the MLM, but can create prompts that lack interpretability, as they do not need to make sense linguistically. They optimize prompts in order to increase performance on the factual cloze filling task, and report an increase in precision of 12 percent points for BERT compared to using the standard prompts in the LAMA data[1] [Petroni et al., 2019].

[1] LAMA is an abbreviation for LM Analysis [Petroni et al., 2019, 2020a]

3.1.2 Output-based Evaluation of Factual Correctness

Constrained Outputs [Luo et al., 2023] create three types of questions automatically (true / false, multiple choice, short answer questions) using knowledge graphs with templates and LM-based question generation using GPT-3.5. Evaluation of short answer questions is done using fuzzy matching, to check whether the reference answer appears partially in the answer sequence generated by the LM.

Open or unconstrained outputs [Rabinovich et al., 2023] use generative LMs in greedy decoding mode on the QA task. They address the challenge of measuring open-ended answers to reference answers by using (1) a partial exact match metric, which judges a generated answer as correct if at least one reference token is contained in the generated answer. (2), for longer answers, they measure semantic consistency using a natural language inference classifier. If the reference answer entails the generated answer, the output is judged as being correct.

3.1.3 Factual Correctness Estimated Via Internal Representations

Metalinguistic Prompting vs. Probability Measurements [Hu and Levy, 2023] compare metalinguistic prompting to direct probability measurements on the tasks of: word prediction, predicting semantic plausibility between two alternative words, and syntax correction. They show that the internal consistency between direct internal evaluation and prompted evaluation diverges, because there is no guarantee that the generated response to a metalinguistic prompt corresponds to the internal representation of the LM. Therefore, they claim, linguistic knowledge is better evaluated on the internal probabilities alone, and exclude possible disturbances induced by the decoding. Similarly on the other tasks, the accuracy on word prediction and sentence syntax judgment based on internal sequence probabilities is higher than the accuracy of the metalinguistic judgements obtained from the outputs.

For this reason, and the additional challenges the comparison between generated responses and ground truth labels confront us, we focus on methods estimating whether a LM has a representation of a fact in its parametric knowledge via the internal representations.

LAMA Probe [Petroni et al., 2019] converts subject, relation, object triples or QA pairs to cloze statements using manually defined templates. This generates cloze statements containing factual knowledge in the form of subject, relation, object. The LM's logit distribution at the mask's position is used to select the most likely object using argmax over the logit distribution. If the token that received the most probability corresponds to the ground truth object, they conclude the LM has a correct representation of this fact contained in its parameters. This mask prediction approach is only possible for bidirectional LMs, as these can attend to contexts on two sides of the masked token. Unidirectional LMs need to have the masked token at the last position in order to attend to the context on the left.

Stability of Factual Knowledge under Paraphrased Prompts [Jiang et al., 2021] extend [Petroni et al., 2019] by creating semantically equivalent prompts containing a masked object. If the prediction is the same as the ground truth as often as possible over all semantically equivalent prompts, they conclude that BERT knows the specific fact expressed in the prompts. However, they note that BERT is very sensitive to the phrasing of prompts. Depending on the relation, they find sequences that lead to an accuracy improvement of 5-60% [Jiang et al., 2021]. Similarly, [Elazar et al., 2021] focus on pre-trained masked LMs and investigate consistency of the mask token prediction in paraphrased sentences. A LM with solid parametric and linguistic knowledge, should exhibit invariance in its prediction of the masked token. They

use a subset of relations from LAMA T-REx [Petroni et al., 2019], and release their paraphrase templates. Also they introduce a new pre-training objective, the consistency loss. For large autoregressive models the influence of paraphrased prompts is studied in [Gonen et al., 2023], which evaluates OPT (175B) [Zhang et al., 2022] and the familiarity of Bloom (176B) [Workshop, 2023] to paraphrased prompts. A LM attributes a low perplexity score to prompts that it is familiar with. They show that there is a strong correlation between a prompt's perplexity and the answer accuracy highlighting the sensitivity that even LLMs exhibit towards paraphrased prompts.

All these works suggest that it is necessary to include paraphrased prompts in our probe. In this work, we will therefore account for semantically equivalent prompts, as we cannot assume that LMs have a lexically invariant representation of semantically equivalent prompts.

3.2 Estimation of a LM's Factual Confidence

An estimation of a LMs factual confidence can be a valuable addition to a generated answer, as it is not possible for a user to judge confidence based on the tone of the generated answers alone.

There are two different notions for factual confidence of LMs. A scoring variation and a generative variation. The "scoring method" assesses a given sequence and predicts how confident the LM is that the given sequence is factually true. The "generative method" estimates how likely it is that given a prompt, the right factual answer can be generated. In [Kadavath et al., 2022] and [Mahaut et al., 2024] the former is referred to by *P(True)*, the latter by *P(I know)*. This work will focus on the former—P(True)—and evaluate a LMs factual confidence based on sequence probabilities that are normalized to a reference set.

According to [Mahaut et al., 2024] there exist three methods to estimate factual correctness of sequences: (1) trained probes, (2) sequence probability, (3) verbalization, (4), surrogate token probability, and (5), output consistency.

[Mahaut et al., 2024] obtain cloze-style prompts from LAMA T-REx and open questions from PopQA, which they paraphrase automatically using Mixtral8x7B-Instruct-v0.1. This is done to obtain meaning preserving alterations of the prompts to test whether the probes are estimating confidence invariant of the specific wording used.

Trained Probes Trained probes use an additional neural network that predicts factual confidence scores from the LM's final layer outputs, or on a combination of hidden layers. This method requires full access to the model's parameters, and labeled training data [Azaria and Mitchell, 2023; Mahaut et al., 2024].

[Kadavath et al., 2022] investigate if LMs can correctly estimate whether they can give a truthful response to a prompt. This is done by training a classifier predicting the probability that the answer to a question is known based on an additional trainable per-token value head. This approach was more successful than to prompt the model to verbalize its confidence in addition to the answer. Also, they investigate in-context learning and whether the estimated certainty of correctly answering a question is influenced by the additional context. They notice a strong relationship between long term memory and how information in the context is used, as this increases the certainty of the LM.

Trained probes perform most reliably on predicting factual confidence compared to the remaining probes. However, as all other probes are 0-shot based, this method has an advantage, as it is fitted to the data in [Mahaut et al., 2024].

Sequence Probability The probability of a sequence can be used as an estimation of factual confidence. However, a LM's estimation of a sequences' probability does not indicate the probability of a sequence being factually correct. It depends on the individual tokens in the sequence and is influenced much by the individual tokens it contains. Semantically equivalent sequences that differ only in the amounts of tokens or words they contain, are associated with very different sequence probabilities. These differences can be of the same magnitude as sequences that express two separate facts. This non-existent calibration of sequence probabilities to factual correctness allows us to use this method only as an unreliable estimation of a model's factual confidence [Xiong et al., 2024; Mahaut et al., 2024]. This method requires access to the LMs weights. In comparison to the trained probe, this method predicts factual confidence 40% worse (in terms of AUPRC scores on true / false prediction of sequences in T-REx) [Mahaut et al., 2024].

Verbalization The verbalization method requires the LM to report its confidence level in addition to the answer directly in the generated output. Some literature reports that instruction-tuned LMs are able to make correct estimates [OpenAI, 2024]. However, prediction of factual correctness is not an objective in the pre-traininig task. This method is well-suited for black-box model settings, as no access to the weights is required. [Mahaut et al., 2024] recommend to use this estimator of factual correctness only for large, instruction fine-tuned LMs.

Surrogate Token Probability This probe is a combination of the verbalized probe with the model's probabilities attributed to specific tokens, which are requested in the prompt, in addition to a claim. For example the requested tokens can be "true" or "false". They are used to judge a claim. The probabilities attributed to these tokens is then used to make an estimation about the factual correctness of the claim [Kadavath et al., 2022; Mahaut et al., 2024]. This probe requires access to the LM's weights, but, as with the probe based on sequence probabilities, the probabilities can be approximated using sampling. [Mahaut et al., 2024] recommend to use this estimator of factual correctness only for large, instruction fine-tuned LMs.

Output Consistency By sampling multiple outputs to a given prompt, we can estimate how consistent they are in terms of factuality. The assumption is that the more similar the outputs are, the higher the confidence of the LM has to be. This method is well-suited for a black-box model setting, as no access to the weights is required [Manakul et al., 2023]. [Mahaut et al., 2024] note that the confidence scores predicting a LMs knowledge remain stable (between 8 paraphrased templates) for a proportion of $\approx 10\%$ of the facts in the T-REx data, for the remaining facts, they mostly note standard deviations between $0.1 - 0.5$.

3.3 Automatic Paraphrase Generation

Previous to the era of LLMs, various approaches have been used to generate semantically equivalent versions of a given sequence [Zhou and Bhat, 2021; Dong et al., 2017; Sharma et al., 2023], most notably back-translation [Sennrich et al., 2016], which was used in [Jiang et al., 2021]. However, if semantic equivalence can be ensured by manually checking the paraphrases generated, it is more convenient to use a LM. Various work has utilized LMs for paraphrasing successfully, therefore we also decide to create paraphrases using Mistral-7B-Instruct-v0.2 [Mahaut et al., 2024].

Chapter 4

Method

In this research we create a probe to better estimate whether a LM has a representation of a fact stored in its parametric knowledge. We refer to this probe as "Probabilistic Prompting", which we design in the form of a "slot-filling"-probe, that tasks a LM to select an object from a given set of options that best fits into a given sequence. These sequences express facts in natural language and consist of three elements: a relation, a subject and an object placeholder. For every relation and subject, we create sets containing possibly relevant, related objects that could be filled at the placeholder's position: $O(r)$, with o being any object of this set. We refer to these sequences with a filled in object as $t(r, s, o)$. One of these objects creates a factually valid statement, all others create sequences that are factually incorrect, but still consist of valid natural language. Therefore $O(r)$ consists of one true object and various factually false objects, $o(r)^+ \cup O(r)^-$.

We consider every sequence, $t(r, s, o)$ to be an output that was generated by an autoregressive LM and obtain its corresponding log-sequence probability, $g(x) = \log P_{LM}(t(r, s, o)|c)$, by summing the assigned token log-probabilities in $t(r, s, o)$. As c can be any context, instructions, or few-shot demonstrations, that is kept constant for every relation, subject pair, we can omit the context's log-probability, and obtain a probability distribution of all realizations of sequences with different objects at the object placeholder position by renormalizing th LM's log-sequence probabilities over these sequence realizations:

$$P(O = o|T = t, R = r, S = s) \triangleq \frac{\exp(g(t(r, s, o)))}{\sum_{o' \in O(r)} \exp(g(t(r, s, o')))} , \tag{4.1}$$

From this conditional distribution over possible objects, we can observe, which object the LM considers most likely to be in a given sequence and also how certain the LM is in a specific object realization.

Paraphrase Templates In natural language, sequences that differ on the surface level can express the same semantics. To address this, we utilize several paraphrase templates $T(r)$ that express a given relation r into our probe and make the assumption that each individual paraphrase template $t(r) \in T(r)$ is equally likely to occur. Then, we can derive a probability distribution over objects that is dependent only on a relation and a subject, by combining the conditional distributions in Equation (4.1) under a uniform prior over paraphrase templates:

$$P(O = o(r)|R = r, S = s(r)) \triangleq \sum_{t \in T(r)} \underbrace{P(T = t|R = r)}_{=1/|T(r)|} P(O = o|T = t, R = r, S = s) , \tag{4.2}$$

with $|T(r)|$ being the number of different paraphrase templates we defined per relation.

This probe allows us to get insights into how certain the LM is that a fact is true or wrong, using the conditional probabilities assigned to sequences normalized over a reference set of sequences that contain closely related but factually wrong objects. The main considerations regarding the quality of our probe are how many negative objects we sample, and how many paraphrase templates we consider.

We describe implementation details, design choices, and evaluation in more detail in the following section. In Section 4.2, we explain how we pre-process the sequences from the original LAMA-TRex, PopQA and hypernym data.

4.1 Implementation Details

4.1.1 Sequence Components: $t(r, s, o)$

We explain the components of the sequences needed for our "slot-filling" probe using the example "Joe Biden was born in Scranton." With *Joe Biden* being the subject, *Scranton* the object, and the relation being *born in*: $t(r = \textit{born-in}, s = \textit{Joe Biden}, \textit{Scranton})$. The specific template defines how the relation is expressed lexically.

- $R = \{r_1, \ldots, r_I\}$ with every element r being a relation from the set of all relations R in a given dataset. r connects a subject and an object, e.g., Joe Biden *was born in* Scranton.
- $S(r) = \{s_1, \ldots, s_J\}$, with the set S being specific to the subjects occuring in relation r. This can be *Joe Biden* or any person.
- $O(r) = \{o_1, \ldots, o_K\}$, with set O being specific to the objects occuring in relation ri. Furthermore, these objects can be categorized into two proper subsets based on whether a realization of o renders the sequence as objectively correct or incorrect. o^+ is then additionally dependent on the subject: $O \subseteq o^+ \cup O^-$. In our example o^+ is only one element, Scranton, and O^- contains any other city names.
- $T(r)$ is a function that, given a relation r, returns a list of all paraphrased templates of that relation. For instance, $t(r)_1$ of the *born-in*-relation denotes a specific instance of a paraphrased relation template: "s *took their first breath in* o.". The template defines how the relation is expressed lexically.

A factually true statement can thus be expressed by a triplet of $t(r, s(r), o(r)^+)$. This sequence can be expressed in multiple, equivalent ways by inserting $s(r)$ and $o(r)^+$ into a different relation template $t(r, s, o)_h$. In contrast, a sequence that is invalid, or factually incorrect, is obtained by sampling $o(r)^- \in O(r)^-$.

4.1.2 Paraphrase Template Generation

In our work, the generation of paraphrases is fully automated by prompting Mistral-7B-Instruct-v0.2 with an instruction to create 19 more paraphrase templates like the original. As a second step, we postprocess the generated paraphrase templates to (1) ensure that the placeholders for subject and object remain intact, and (2) add a rule-based check to assert that the paraphrases end with a colon. To ensure the paraphrased templates are semantically equivalent to the original relation template, we also check them manually.

4.1.3 Generating Factually Wrong Sequences

Each row in the datasets presented in Section 4.2 states a valid fact or true relation between a subject and an object of a relation type: $t(r, s, o)$, such as "*Rome is the capital of Italy.*".[1] After obtaining factually true (or positive) sequences from the original data, we create negative instances by defining for each relation a set of negative objects ($O(r)^-$). We obtain the elements of this set, from the factually true sequences of all other subjects (e.g. *Paris, France*) in the original data, e.g. for the *capital-of relation* we obtain country names such as *France, United Kingdom*. For a specific relation and a subject, we create negative (factually wrong) candidate instances by filling in all objects of the negative set into the relation template: $o^- \in O(r)^-$, e.g. *Rome is the capital of France*.

Sampling Object Candidates: $O(r)^-$ The larger the cardinality of the set $O(r)^-$, the more realistic the estimations of the probability distribution over sequences, $P(O = o(r)|R = r, S = s(r))$. A larger set allows a more complete picture of the world to be modeled. Taking into account a smaller subset of possible objects makes our estimate of an object's probability, $P(o(r)|r, s(r))$, more optimistic, as the probability mass is shared between fewer members. Consequently, the estimate is to be understood as an upper bound of reality, and therefore an optimistic estimate. The sets of objects with negative examples ($O(r)^-$) are sampled once, and are kept constant across all experiments. In Chapter 6, we investigate how, for example, providing additional context increases the factual certainty estimation of LMs in comparison to the 0-shot setting. It is therefore necessary to keep any other factors as comparable as possible.

The only change to the size of this set occurs on the subject level if an element in the negative set is equal to the true object. In this case the true object is removed from the negative set of this r, s triplet. This leads to minor side effects: When the true object of a subject occurs in the negative set, this specific r, s, o triplet is compared to one fewer negative object. However, for our analysis it is more convenient to keep these sets stable, and thereby excluding one more factor of complexity that individual $O(r, s)^-$ sets per r, s pair would confront us with.

The set ($O(r)^-$) is constructed by using objects that occur in one relation. Only if the required number of objects per relation is higher than the available number of objects in this specific relation, the difference is sampled from other relations. For a $O(r)$ cardinality of 50, this is needed for six of T-REx's relations: *S plays O music.*, *S is affiliated with the O religion.*, *S plays in O position.*, *The native language of S is O.*, *S is represented by music label O.*, *S was originally aired on O.*; and two relations of PopQA: *The color of S is O.* and *S play O.*

Restrictions on the Number of Sequences As we have limited compute available, and encoding all sequences separately increases costs, we use only a subset of the T-REx and PopQA datasets by sampling 50 subjects randomly per relation. Therefore, we predefined three parameters (all apply per relation): The number of subjects, the number of negative objects and the number of paraphrased templates. We are more interested in the effects semantically equivalent sequences have on factual knowledge probing, and how the probabilities are distributed between the different objects, thus we also prefer larger $O(r)^-$ sizes. Restricting the number of subjects per relation to 50, leads to a total number of 500,000 to 1,200,000 sequences per dataset, of which there are 50x more negative examples than positive ones—our dataset is therefore heavily imbalanced in terms of factuality labels, and a classifier that predicts every sequence to be factually wrong would already achieve an accuracy of $\frac{50}{51} \approx 98\%$. This is one of the reasons to use selective prediction.

[1] The relation in the original data is the original relation that is paraphrased into various templates.

4.1.4 Sequence Probability: $P_{\boldsymbol{LM}}(t(r, s(r), o(r))|c)$

We use the two autoregressive LMs, namely GPT-2-L [Radford et al., 2018] and Mistral-7B-Instruct-v0.2 [Jiang et al., 2023] on a NVIDIA A100-SXM4-40GB GPU. The models are used in evaluation mode, allowing us to obtain the logits assigned to each token in the output sequence. At each time step the next token is chosen based on a distribution over the vocabulary, which is used to predict the next token. As the full sequence, $t(r, s(r), o(r))$, is known beforehand, we can obtain the logits assigned to the sequence, simply summing the log-probabilities corresponding to the output sequence's tokens.

These logits correspond to the joint probability of the context[2] and the output sequence: $P_{LM}(t(r, s(r), o(r))|c)$. But as the context is equivalent across all sequences of a $s(r), r$ pair, we can omit the context, as every sequence is conditioned on an equivalent distribution.

To obtain the probability of an output sequence, we first convert the logits to log probabilities by applying log softmax. Then, we can sum the log-probabilities of each output token. These log-probabilities are usually very small as the outcome space consists of the realizations of every possible string. We impose more constraints as we want to have a more realistic impression about the sequences the model considers to be more probable, which we think should be the sequences that are in line with the ground truth. Factually wrong sequences should not occur in the training data as they do not represent valid knowledge, and be therefore considered less likely. The calculation and re-normalization is described in the next section.

4.1.5 Slot-filling Probability Distribution: $P(O = o(r)|R = r, S = s(r))$

For every relation, subject pair, we obtain a probability distribution over the objects in the set $O(r)$: $P(O = o(r)|R = r, S = s(r))$ from $P_{LM}(t(r, s(r), o(r))|c)$. This distribution indicates how probable it is to fill the object slot of a sequence for a given relation and subject pair with object $o(r) \in O(r)$. The probability assigned to a specific object, is denoted as $P(o(r)|r, s(r))$. The larger this value, the more confident is the LM that this specific object needs to be filled into the sequence. In Section 4.1.7 we then evaluate whether this probability distribution allows us to distinguish factually correct from incorrect sequences.

Without Paraphrases: First, we will explain the calculation of $P(O = o(r)|R = r, S = s(r))$ in a setting without paraphrased templates. In this setting, we obtain a probability distribution over all sequences with objects related to a specific instance of r and $s(r)$ by applying the softmax function on the log-probabilities, as described in Section 4.1.4. This distribution describes how likely a LM considers it that a given sequence $r, s(r), o(r)$ is realized, and in a further sense, how likely it is that this sequence is true. Sequences that state a valid fact should be more likely than sequences expressing a fact that is not valid. See Figure 4.1 for an example.

If we consider all realizations of triplets of a relation and subject pair, each realization contains a different $o(r)$ from the $O(r)$ set. We normalize these log-sequence probabilities by applying the softmax function to obtain a probability distribution over all elements in $O(r)$ (see Equation (4.3)). This distribution is then indicative of which object is most likely to be the factually valid realization of $o(r)$ associated to the subject $s(r)$ by relation r.

[2]In the experiments in Chapter 6, we provide additional context such as the Wikipedia article of a subject or n-shot examples always per instance of $(r, s(r))$ pair. This ensures we can safely ignore the sequence probability assigned to the context, and only consider the probabilities assigned to the tokens of the factual sequence triplet, because the first part of the joint probability is equal for all instances

$$P(O = o|T = t, R = r, S = s) \triangleq \frac{\exp(\log P_{LM}(t(r, s(r), o(r))|c))}{\sum_{o' \in O(r)} \exp(\log P_{LM}(t(r, s(r), o')|c))}, \tag{4.3}$$

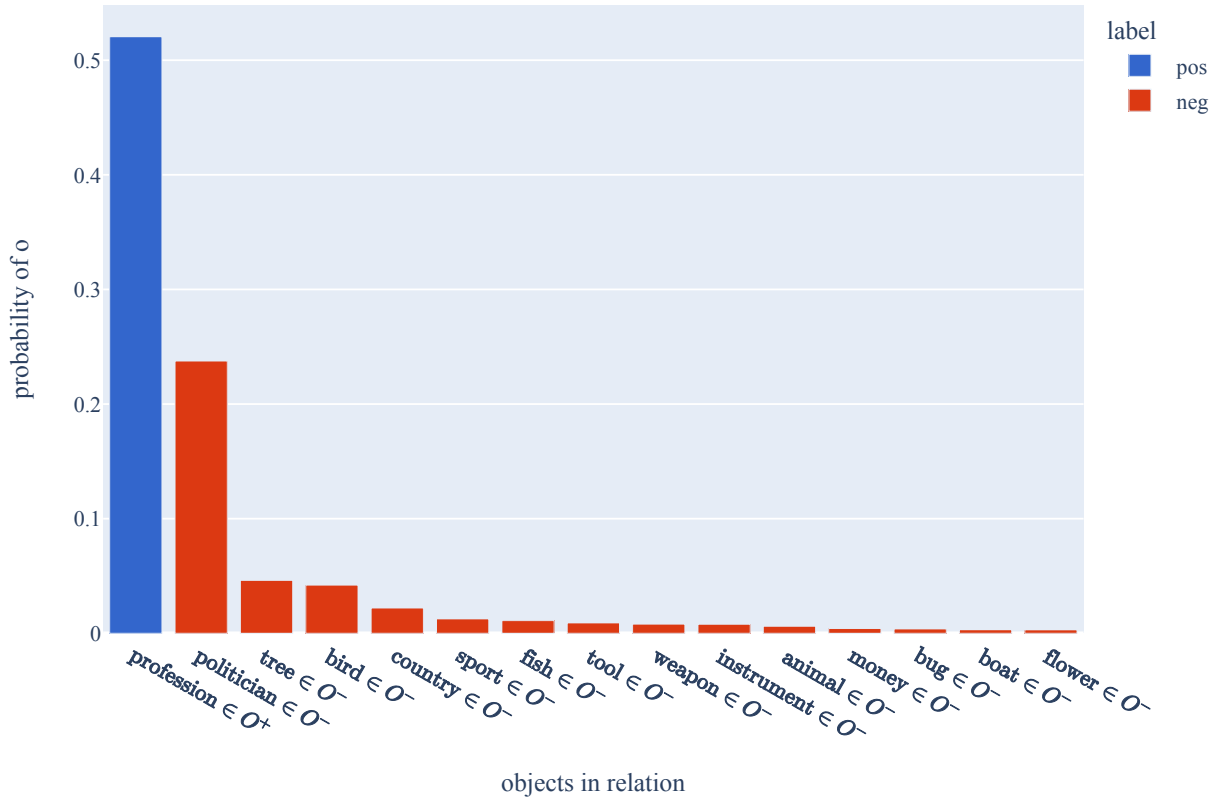


Figure 4.1: Illustration of the $P(O|R = r, S = s)$ distribution assigned for the hypernym relation and subject psychologist. The hypernym relation relates the hyponym (subject) e.g. *psychologist* to the hypernyms in $O(r)$ e.g., *profession* / *politician*. An example of such a sequence is "Psychologist is a type of profession.". The visualization includes a subset of 15 objects (of the 50 objects in total), this is the reason the probabilities in the graph do not sum up to 1. This example includes no paraphrased templates. The colors of the bars denote the ground truth label of a sequence $s(r), t(r), o(r)$.

With Paraphrased Templates As natural language allows us to express the same semantic meaning in various ways, our probe needs to consider one more dimension. To address this, we use paraphrase templates ($T(r)$) that express the same semantic relationship as r, but articulated differently. The exact approach we used to obtain these is described in Section 4.1.2 and an example is provided in Figure 4.2. Paraphrases have an impact on the calculation of $P(O = o(r)|R = r, S = s(r))$ as the normalization needs to be extended to consider not only all realizations of objects in $O(r)$ but also all different paraphrases of $T(r_i)$. This makes the estimated distributions of $P(O = o(r)|R = r, S = s(r))$ invariant to the surface level differences of prompt phrasing, as in theory every possible way to phrase a prompt is considered in the probe.

As paraphrased templates have different lengths, the log-sequence probabilities will differ based on the differences in length and it can be the case that longer sequences receive almost

no probability mass. We therefore impose the assumption that every paraphrased template is equally important, and apply the softmax function not over all paraphrased templates of an $r, s(r)$ pair, but over each $t(r), s(r)$ pair. $|T(r)|$ denotes the number of paraphrased templates including the original template of a given relation r. We can then obtain $P(O = o(r)|R = r, S = s(r))$ by combining the conditional distributions per paraphrase template uniformly (see Equation (4.2), Equation (4.4) for the implemented version and Figure 4.3 for a visualization).

$$P(O = o(r)|R = r, S = s(r)) \triangleq \frac{\sum_{t \in T(r)} \exp(\log P_{LM}(t(r, s(r), o(r))|c))}{\sum_{o' \in O(r)} \frac{1}{|T(r)|} \sum_{t \in T(r)} \exp(\log P_{LM}(t(r, s(r), o')|c))} \tag{4.4}$$

Intuitive Definition of Factual Certainty Our obtained probability value $P(o(r)|r, s(r))$ measures how much weight is assigned to the specific object of an $r, s(r)$ pair (of all the objects in the set of $O(r)$). In the case the fact is stored in the LM's parametric knowledge, most weight (more than 0.5) is assigned to the object with the positive label. Consequently, the combined probability assigned to every element of the negative set is less than the probability assigned to the positive object.

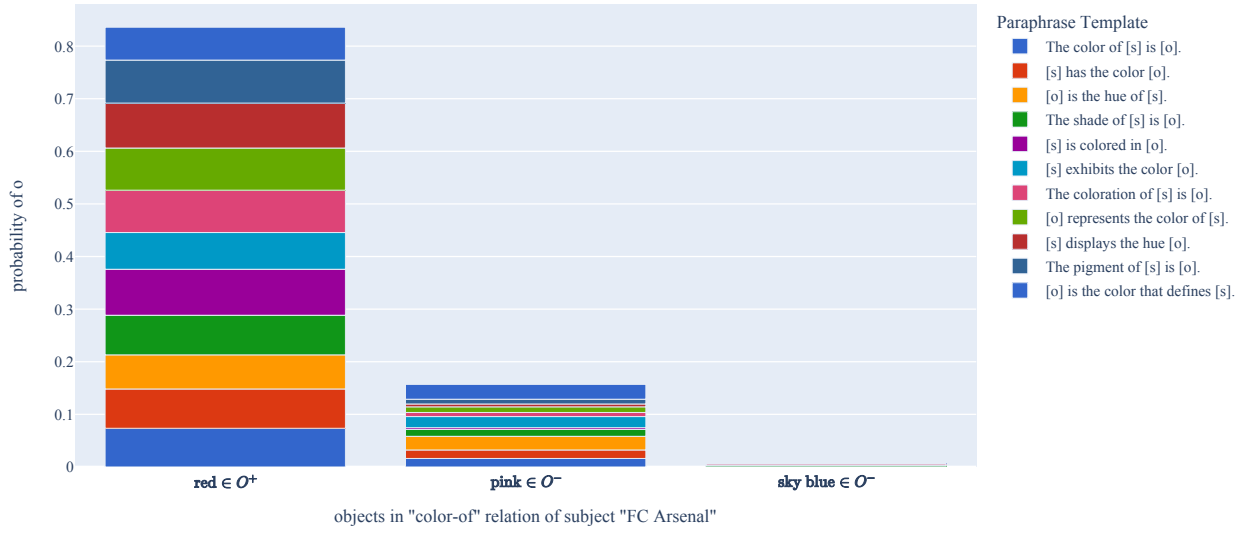


Figure 4.2: Illustration of $P(o(r)|r, s(r))$ distribution assigned for the *color* relation and subject *Arsenal FC*. Visualization includes an exemplary set of 3 objects, and 10 paraphrased templates in addition to the original. $P(o(r)|r, s(r))$ values are assigned to each object and each paraphrased template of r. Text in the bars denotes the ground truth label of a sequence $s(r), t(r), o(r)$.

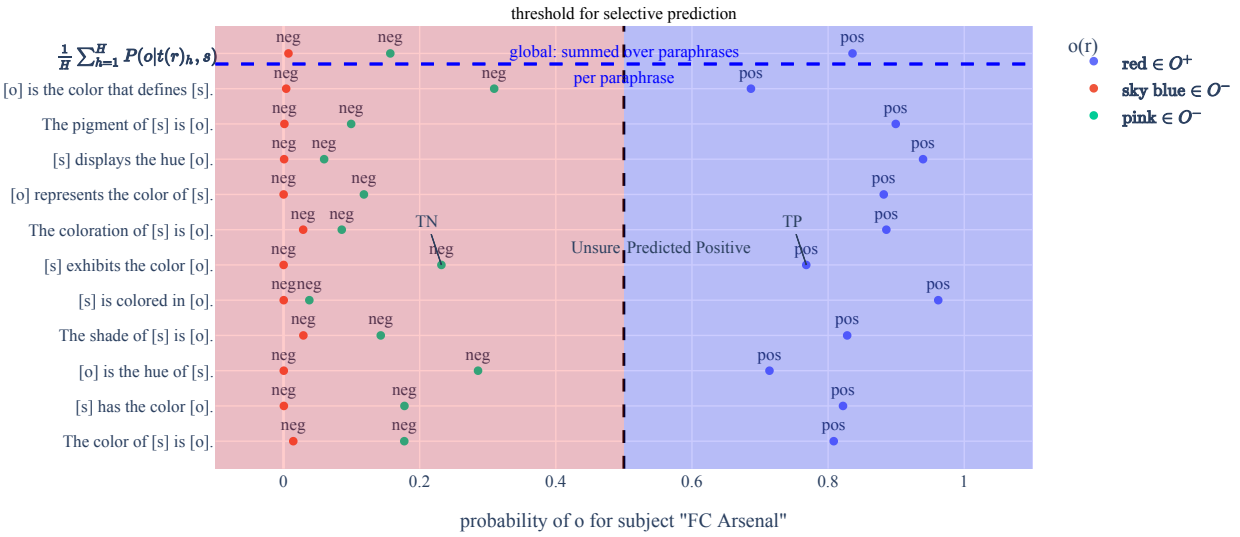


Figure 4.3: Exemplary illustration of selective prediction based on the probability of an object for relation *color-of* and subject *FC Arsenal* showing how the probabilities of different paraphrases are aggregated. In selective prediction, only objects that receive a probability above the threshold are considered in the precision calculation, the objects with a smaller probability are abstained from prediction. The threshold of 0.5 assures a high confidence in the predictions, as for such a subject, all other objects combined received a probability mass of less than 0.5. The remaining rows show how the probabilities per object and individual paraphrase template are distributed for one relation, one subject and a selection of three objects. A full example with more objects can be found in Figure A.8.

4.1.6 Experimental Methodology

As there are many sampling processes involved in selecting the sets of objects and subjects, we only sample these instances once for every dataset and reuse the same samples for all experiment configurations. This way we can assure the measured differences per experiment are attributed to the changes we want to evaluate.

We also limit the sizes of the datasets we conduct our experiments on, in order to not create millions of permuted sequences that need to be encoded, as compute is limited. This has implications on the reliability of the results, as only 50 different subjects per relation can be evaluated. A comparison to the full datasets is thus not made in this thesis. However, every dataset in the domain of factual knowledge has these limitations as no dataset can provide a complete list of all existing factual knowledge.

4.1.7 Classification

We evaluate a LM's amount of knowledge via a binary classification task, in which given a subject and a relation, the task is to predict whether an instance of $O(r)$ makes the sequence factually true or false. Ground truth labels are available, and true instances correspond to the original sequences in the dataset. The sequences generated by substituting $o(r)^+$ by an element in $O(r)^-$ are labeled as negative as they make sequences factually incorrect. However, as our dataset is imbalanced we adapt the classification setting to selective prediction, where in addition to the binary classification into factually correct and incorrect, it is also possible to abstain from making a prediction if the confidence is not high enough (Section 4.1.7).

The classification performance is evaluated using standard metrics to evaluate a binary classifier, either on predictions obtained from the $P(o(r)|r, s(r))$ values, or by ranking objects according to the log-sequence probability a LM assigned these permutations. We use the confusion matrix in Table 4.1 to evaluate the classifier's performance.

		prediction	
		negative	**positive**
label	**negative**	TN	FP
	positive	FN	TP

Table 4.1: Confusion matrix for binary classification.

Binary Classification of $P(o(r)|r, s(r))$ Values Given a relation, a subject, and a probability threshold t, our approach predicts all sequences that receive $P(o(r)|r, s(r)) > t$ as being factually correct and assigns a positive prediction to them. Values of $P(o(r)|r, s(r))$ below the threshold are predicted to be factually incorrect. The ground truth labels remain the same as in Section 4.1.7. Intuitively, if the model assigns a probability value, $P(o(r)|r, s(r)) > 0.5$, it is certain that this fact is correct, as the entirety of objects in $O(r)^-$ is assigned less probability mass than to the object in $O(r)^+$. This method alone does not take into account the confidence value that is that is incorporated in $P(o(r)|r, s(r))$, as we enforce a decision for every sequence. We address this shortcoming using selective prediction.

Selective Prediction takes only into account values of $P(o(r)|r, s(r)) > t$. Every $P(o(r)|r, s(r)) < t$ receives no predicted label. The classifier thus has the ability to dynamically abstain from making predictions if its confidence is not large enough. Hence, we obtain correctness predictions only for values of $P(o(r)|r, s(r)) > t$, where the LMs confidence is larger than the given threshold. In the last step, we compare the sequences that are predicted to be factually correct with a high confidence to the ground truth labels of the sequences. This approach is illustrated in Figure 4.4. For our experiments (Chapter 5 and Chapter 6), we focus on selective precision as in our setting, precision and accuracy are equivalent: The classification results of sequences that are situated above the threshold can either be TP or FP. Sequences lying below the threshold are abstained from prediction, thus there are no TN and no FN sequences. A threshold-independent description of the classifier's performance can be obtained by calculating the performance for many thresholds separately in a risk-coverage curve. The better the predictions are, the closer to one is the area under the risk-coverage curve. We will explain these methods in the following paragraphs.

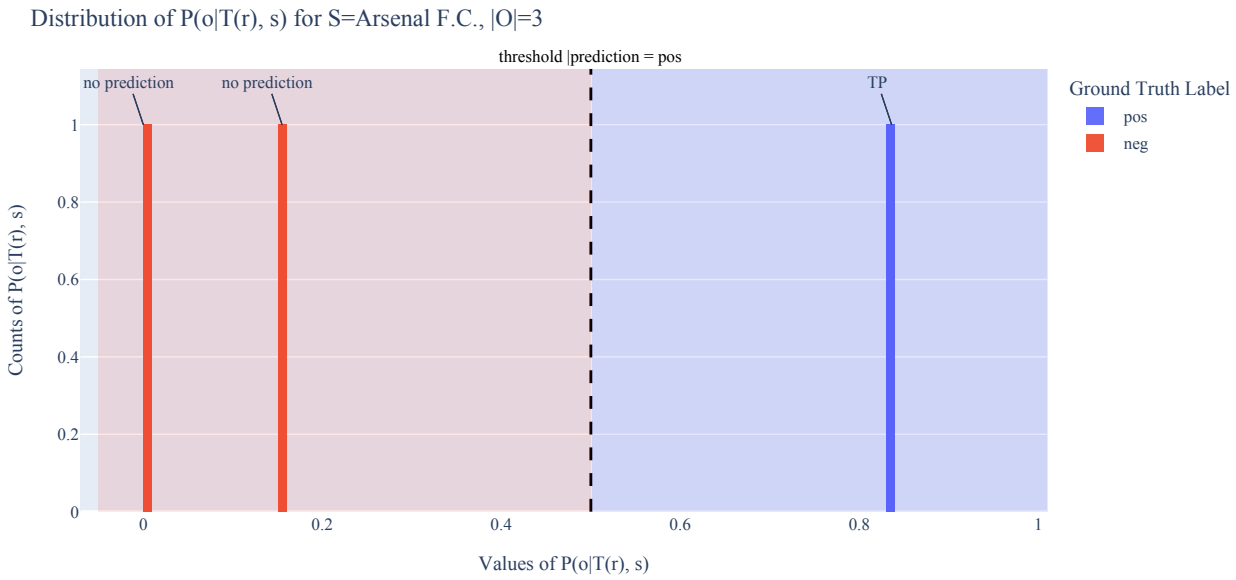


Figure 4.4: Histogram visualizing the classification of $P(o(r)|r, s(r))$ for subject *Arsenal F.C.* and three different objects. Elements below the threshold are abstained, therefore no factual prediction is made for these objects. Only one sequence is predicted to be factually correct (which is indeed true). The object *red*, received a probability of more than 0.5—therefore we can assume a high certainty in this prediction.

Binary Classification According to Rank This setting considers all paraphrase templates of a relation separately and ranks all sequences with different objects from $O(r)$ for a subject according to the logarithmic probability of the sequences $(P_{LM}(s, o(r)_k, t(r)))$. The object with the highest probability is considered to be the object that makes the sequence factually true, all others are considered to be false. There are two possibilities to calculate this ranking of objects: (1) ranking the sequence probabilities of the objects aggregated over paraphrased templates, or (2) ranking the objects according to their sequence probabilities separately per paraphrased template and therefore obtaining multiple positive predictions. For an illustration of these versions, see Figure 4.5. In our results we focus on version 1.

Both versions differ from the approach described in Section 4.1.7, bacause there is no threshold used to generate the predictions, as they depend only on the rank.

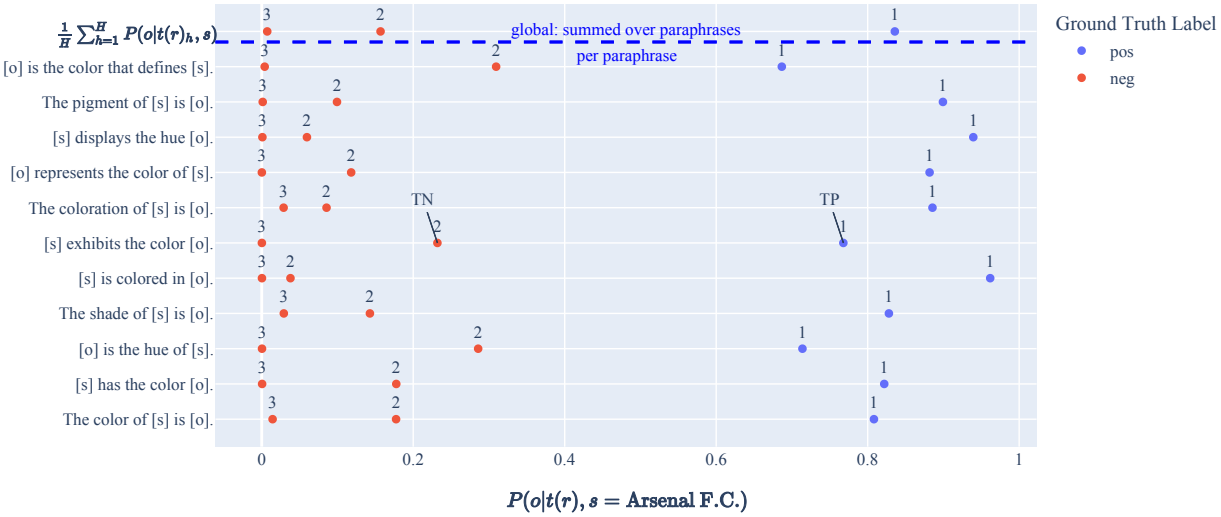


Figure 4.5: Exemplary illustration on a subset of one relation, one subject, two paraphrase templates, and a selection of six objects. It visualizes the two approaches to classify the predictions according to the ranks assigned to the objects. The first row illustrates rank-1 based prediction over paraphrases, where the object that received most probability is predicted to be factually correct. This approach is not dependent on paraphrases, as the probabilities are aggregated over the individual templates. The bottom rows show the approach of ranking objects per paraphrase template individually. Equivalently predicting the object that received the most probability mass to make a sequence factually true, but for each paraphrase template separately.

General Metrics The following metrics are applied to both types of predictions: Argmax-based Section 4.1.7, and $P(o)$-based Section 4.1.7. An ideal model has only true negative and true positive predictions (and in the case of selective prediction only true positive predictions, as values below the threshold are abstained from being predicted)—this means that the $P(o(r)|r, s(r))$ value associated with each sequence is perfectly predictive of factual correctness.

Precision is the amount of times the prediction was indeed correct, relative to all instances, where the model score was indicating the sequence to be correct. In other words, this is the amount of ground truth positive instances, relative to all sequences that received a score above the classification threshold. This indicates how much certainty we get from a high P(o) value and thus how many of the cases that receive high values are indeed true. In selective prediction precision is equal to accuracy.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.5}$$

Recall measures the amount of times where the prediction was indeed correct, relative to all instances that have a gold label of correct. This measures the amount of times the model was able to recognize true sequences.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.6}$$

F_1-Score is the harmonic mean of precision and recall.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.7}$$

Accuracy measures the number of times the prediction corresponds to the ground truth label, relative to all predictions that were made. This score is less indicative of our data, as the positive and negative classes are imbalanced.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.8}$$

False Positive Rate also called fall-out or false alarm ratio, measures how many times a high value of P(o) was indeed factually untrue, relative to all factually wrong instances.

$$FPR = \frac{FP}{FP + TN} \tag{4.9}$$

True Positive Rate is the same as recall.

Coverage is used in the selective prediction setting, where the value indicates how many data points are taken into account (**absolute coverage**). **Relative coverage** is defined as the number of datapoints above the threshold, relative to the total amount of data points.

The Receiver Operating Characteristic (ROC) Curve illustrates the performance of a binary classifier, under different threshold values, by measuring TPR and FPR at each threshold. Graphically, it is the recall (or TPR) as a function of FPR. A random guess lies on the angle bisector, where FPR is equal to TPR. A perfect classification has TPR of 1 and FPR of 0.

The Risk Coverage Curve is the risk as a function of coverage. Risk can be any metric such as precision or recall. Coverage measures the amount of data points situated above a classification threshold.

The Area Under the Curve (AUC) measures the area under the curve, for example under the ROC or risk-coverage curve. The larger the area, the better is the classifier performance under varying thresholds. A perfect classification results in an AUC value of 1.

4.2 Datasets

All datasets in this work use the following structure, where each sequence expresses a fact using a triplet containing a relation, a subject and an object.

4.2.1 LAMA T-REx

LAMA T-REx is a subset of the LAMA dataset, which probes LMs for factual knowledge. This subset contains 1.3 million Wikidata knowledge triplets that were automatically aligned to extracts of Wikipedia natural language sequences. 41 different relations exist, with 1,000 different facts per relation. As the sequences were extracted verbally from Wikipedia it is very likely that the facts are contained in the pretraining data of LMs [Petroni et al., 2019, 2020b].

Elazar et al. use LAMA-TRex data for factual consistency probing of MLMs, and released their data cleaning process in the *ParaRel* dataset. We follow their approach and keep only N-1, and 1-1 relations, and remove duplicate UUIDs. This amounts to 20,943 examples and 22 relations in total. Furthermore, we keep their manually created paraphrase templates and extend them using our automatic approach in order to obtain 20 paraphrase templates per relation (Section 4.1.2). Detailed statistics on the amount of unique subjects and objects per relation can be found in Table A.1.

The exact amount of knowledge triplets obtained after selecting a random subset of subjects per relation is presented in Table 4.2, the full version can be found in Table A.4. The reason for the total number of positive sequences being larger than the number of subjects is that the statistics show only the number of unique subjects and objects. It can be that the data contains duplicates or that a subject occurs multiple times with another object.

relation	#o	#s	$\lvert O^- \rvert$	#p	#pos	#neg	total
S is located in O.	86	50	50	22	1100	54890	55990
S plays O music.	49	50	49	22	1100	53900	55000
S was founded in O.	78	50	50	22	1100	54604	55704
S plays in O position.	48	50	49	22	1100	53746	54846
S is part of O.	79	49	50	22	1100	54934	56034

Table 4.2: Sequence statistics for the first five relations of T-REx data. Full version can be found in Table A.4. #o denotes the number of unique objects in a relation, #s the number of unique subjects, $\lvert O(r)^- \rvert$ the sampled number of unique objects in the reference set $O(r)^-$, #p the number of paraphrase templates per relation, #pos the resulting number of sequences that are factually correct, #neg the number of incorrect sequences, and total denotes the total amount of sequences per relation.

4.2.2 PopQA

The PopQA dataset contains 14,000 question, answer pairs in 11 different categories, created from Wikipedia knowledge tuples [Mallen et al., 2023]. As neither question nor answer are literal Wikipedia extracts, we can consider this data to be out of domain for LMs which should make the task of factual correctness prediction more challenging (compared to T-REx which is in-domain). We create one manual relation template with placeholders for subject and object per category. To ensure comparability with our probe, we convert the questions into sentences containing the three elements needed to express the factual statements: relation, subject and object, following the format of LAMA T-REx. We use automatic paraphrasing to obtain a total of 20 paraphrase templates per category (Section 4.1.2). The dataset contains multiple possible answers and we use only the first one, as they are semantically equivalent. Detailed statistics on the amount of unique subjects and objects per relation can be found in Table A.2.

The exact amount of knowledge triplets obtained after selecting a random subset of subjects per relation is presented in Table 4.3, the full version can be found in Table A.5. The number of subjects in the *color* relation is smaller, as PopQA only contains 34 subjects for this relation and the relations *capital* and *religion* contain one duplicate subject. The number of unique objects per relation can be smaller than 50 if the relation contains a total of fewer than 50 objects—in this case the remaining O^- are sampled from all other relations. For an individual subject, this set however, cannot include the true object, and we need to exclude the true object

if it was included in the sample of $O(r)$. This leads to some instances that have object sets that are one cardinality smaller.

relation	#o	#s	$\lvert O^- \rvert$	#p	#pos	#neg	total
S's occupation is O.	56	50	50	21	1050	51639	52689
S was born in O.	94	50	50	21	1050	52395	53445
The color of S is O.	50	34	49	21	714	34986	35700
O is S's father.	95	50	50	21	1050	52395	53445
S is located in O.	60	50	50	21	1050	52164	53214

Table 4.3: Sequence statistics for the first five relations of PopQA data. Full version can be found in Table A.2. #o denotes the number of unique objects in a relation, #s the number of unique subjects, $\lvert O(r)^- \rvert$ the sampled number of unique objects in the reference set $O(r)^-$, which make the sequence factually false, #p the number of paraphrase templates per relation, #pos the resulting number of sequences that are factually correct, #neg the number of incorrect sequences, and total denotes the total amount of all factually false and true sequences per relation.

4.2.3 Hypernym

This dataset contains only subjects and objects that are connected by the hypernym relation. We create one manual paraphrase with placeholders for instances of hyponyms (S) and hypernyms (O) such as "*S is a type of O*". The reason for this is that this relation is a N-1 relation in most cases[3]. Using hypernyms as subjects, we would have a 1-N relation which makes evaluation more challenging as there exist multiple answers. The sets of hypernyms and instances thereof are obtained from [Hanna and Mareček, 2021; Van Overschelde et al., 2004; Battig and Montague, 1969].

Detailed statistics on the amount of unique subjects and objects per relation can be found in Table A.3. The exact amount of knowledge triplets obtained after selecting a random subset of subjects per relation is presented in Table 4.4.

relation	#o	#s	$\lvert O^- \rvert$	#p	#pos	#neg	total
S is a O.	30	499	49	21	10500	507675	518175

Table 4.4: Sequence statistics for the hypernym data. #o denotes the number of unique objects in a relation, #s the number of unique subjects, $\lvert O(r)^- \rvert$ the sampled number of unique objects in the reference set $O(r)^-$, which make the sequence factually false, #p the number of paraphrase templates per relation, #pos the resulting number of sequences that are factually correct, #neg the number of incorrect sequences, and total denotes the total amount of all factually false and true sequences per relation.

[3]Some ambiguous hyponyms exist though.

Chapter 5

Experiment 1: Parametric Knowledge

This chapter describes the experimental setup, the results and an analysis of parametric knowledge found in the LMs GPT-2-L and Mistral-7B-Instruct-v0.2.

We illustrate similar outcomes using Mistral-7B-Instruct-v0.2 on the datasets PopQA, LAMA-TRex, hypernym data, and refer the reader to Appendix A for analogous results using GPT-2-L.

5.1 Predicting Factual Correctness of Sequences

In this section we test whether a LM's parametric knowledge is able to distinguish between factually correct and incorrect statements. This addresses research question 1.[1]

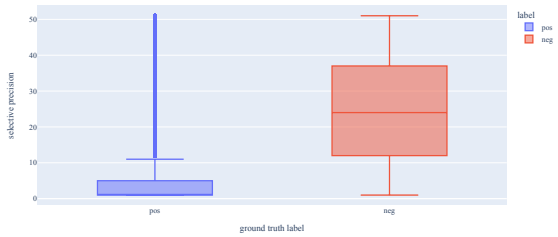
As predictors for parametric knowledge we use (1) the standard approach of argmax, where the classification is done by selection of the most probable object, regardless of the probability mass. We use two versions of argmax prediction: A prediction for each different paraphrase template; and argmax based on the probabilities aggregated over the paraphrase templates. As prediction method (2), we use the values of $P(o(r)|r, s(r))$ as described in Section 4.1.5. The boxplots illustrating this separation are all based on T-REx data as the results on PopQA and hypernym data follow the same trend (Figure 5.1.

5.1.1 Using Argmax O

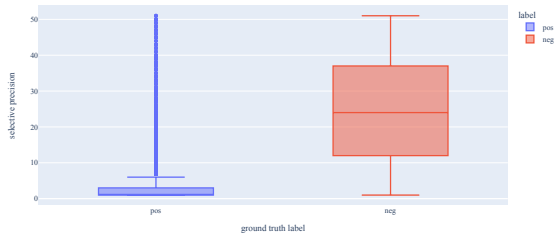
We investigate how well the first ranked object of the set $O(r)$ (the argmax prediction, given any subject, and paraphrased relation template) is suitable to distinguish between factually correct and incorrect sequences. Rank 1 is assigned to the object that received the most probability and rank 50 corresponds to the object the least probability was assigned to.

Results Figure 5.1a shows values of prediction method (1) and indicates that the distributions are indeed different, and on average it would be possible to correctly distinguish between a factually correct sequence and an incorrect sequence by always selecting the object of $O(r)$ that receives the highest sequence probability. However, a significant amount of outliers exist, which leads to an overlap of ground true positive and negative sequences—a perfect separation is thus not possible.

[1] **RQ1**: Is it possible to distinguish factually correct and incorrect sequences based on the probabilities assigned to the sequences' objects via probabilistic prompting, and how do these predictions compare to top-1 predictions, obtained by ranking objects according to their probabilities? (Chapter 1)



(a) Method 1: ranking per template $T(r)$



(b) Method 2: aggregated over paraphrases

Figure 5.1: Two different versions of argmax-based prediction. Most probable object is assigned rank 1 either per paraphrase template (a) or once over all paraphrase templates (b). Underlying sequence probabilities are obtained from Mistral-7B-Instruct-v0.2.

For prediction method (2) whose results can be seen in Figure 5.1b the same issues are persisting. However, the median value is lower and also the interquartile range of the ranks assigned to positive sequences is smaller. This illustrates that different ways of formulating a relation influences the sequence probabilities, and the argmax object is more indicative of factual truth. Nonetheless, distinguishing between positive and negative sequences is not possible, but we see first indications that if a sequence is assigned a high probability, it is most likely true. Conversely, this is not true, low sequence probabilities can be factually true or false. There is no predictive power in the sequence probabilities alone.

5.1.2 Using $P(o(r)|r, s(r))$

If we instead use the probability mass assigned between the different objects of a subject and relation (see Section 4.1.5) as a predictor for factual correctness, a higher value should correspond to factually true sequences, and a low value should correspond to factually wrong sequences.

Results We observe the same trends, as in Section 5.1.1. The approach based on $P(o(r)|r, s(r))$ does not allow to distinguish factually wrong sequences from true instances (Figure 5.2). However, low $P(o(r)|r, s(r))$ are on average a good indication that a sequence is indeed factually wrong. We can therefore use $P(o(r)|r, s(r))$ as an indication of certainty a LM has in its parametric knowledge—more in the next Section 5.2.

Results for the hypernym dataset can be found in Figure A.4 and Figure A.5.

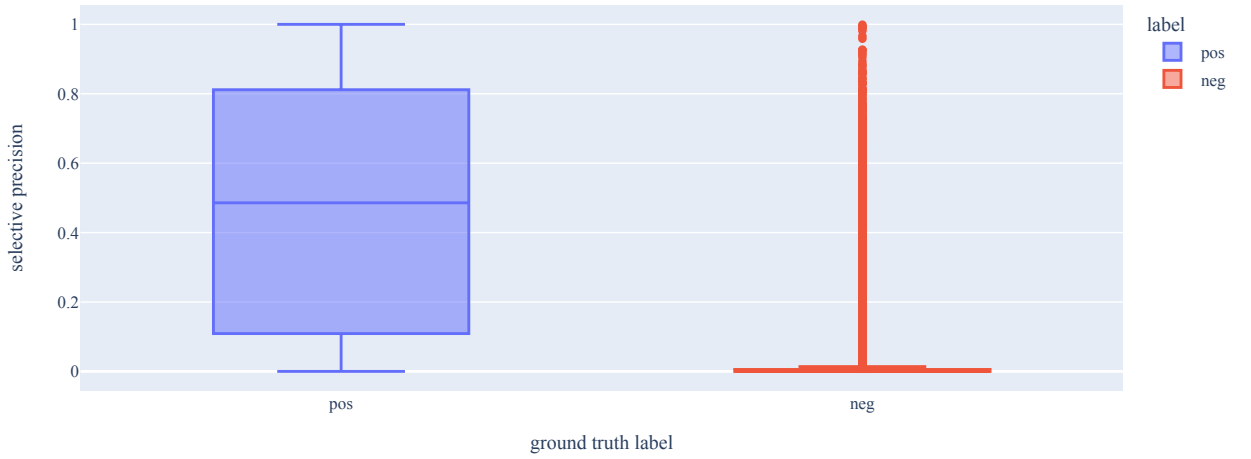


Figure 5.2: Selective precision vs. ground truth label, aggregated over different paraphrases. Underlying sequence probabilities obtained from Mistral-7B-Instruct-v0.2.

5.2 Evaluating What LMs Think That They Know

Both methods, the greedy selection of the highest valued object and selection based on $P(o(r)|r, s(r))$ are relying on the underlying sequence probability the LM assigned. The main issue is that the separation from factually incorrect sequences does not work reliably based on $P(o(r)|r, s(r))$ alone. Thus, in this section, we do not evaluate whether LMs know about the validity of facts in general and can use their parametric knowledge to correctly assess facts, but we focus our investigation on whether LMs "know when they know".

As $P(o(r)|r, s(r))$ values are probabilities normalized over the reference objects in $O(r)$ and also account for multiple ways of phrasing a relation, we can use the value of $P(o(r)|r, s(r))$ as an indication of certainty towards a specific object. Using selective prediction, we consider only objects whose probabilities are above a certain threshold (the remaining ones are abstained, and thus excluded from being predicted) and evaluate whether these are indeed corresponding to ground truth positives. This is first evaluated over all sequences in a dataset, and then in Section 5.2.2 by individual relation. This section will address research question 2,[2] and research question 3.[3]

5.2.1 Per Dataset

We present per-dataset results using the intuitive threshold of 0.5 in Figure 5.3 and compare the precision of selective prediction with the standard approach of ranking objects by their sequence probability. A high precision indicates how many of the sequences that receive high values are indeed true.

[2] **RQ2**: Can we use the probabilities assigned to the reference objects as an indication of factual certainty? (Chapter 1).

[3] **RQ3**: Does the inclusion of paraphrased templates improve the probabilities assigned to the objects both as a measure of certainty, and also in the prediction of factual correctness? (Chapter 1).

Results For the results using other thresholds see the risk-coverage graphs in Figure 5.4.

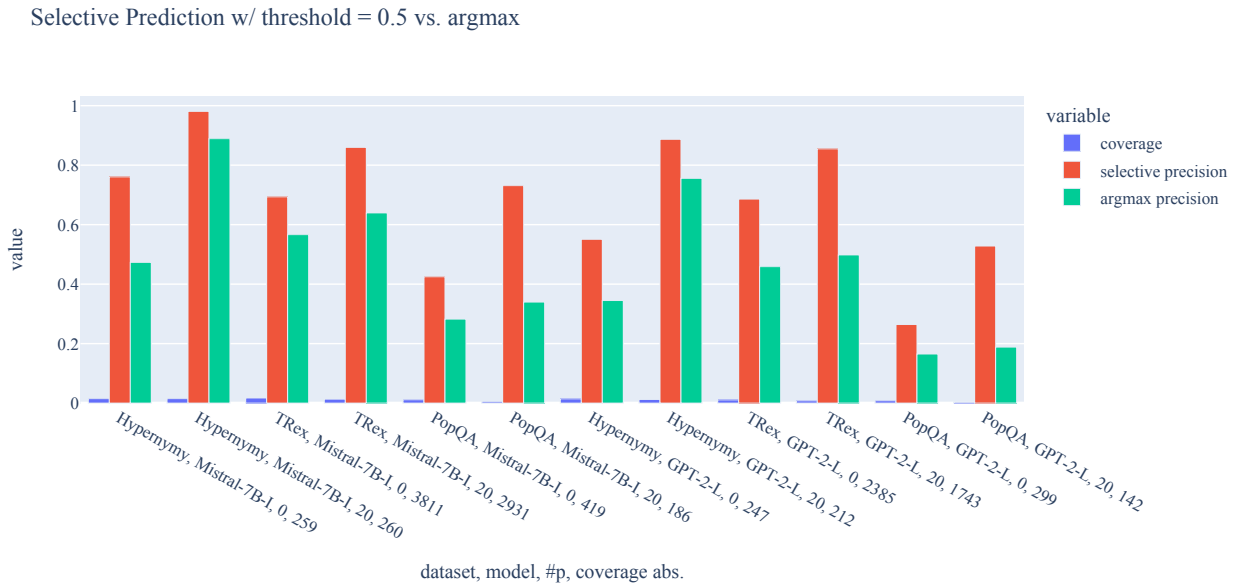


Figure 5.3: Comparison of the selective precision and argmax ranking precision, plotted by dataset, model, and the relative amount of objects receiving probability > 0.5. The threshold of 0.5 is applied only in the selective prediction setting. $|T(r)|$ denotes the number of paraphrase templates available in a dataset, including the original template.

Precision at a 0.5 threshold In Figure 5.3 a threshold of 0.5 limits coverage significantly to approximately 1.7% per dataset. Only in this very small fraction of sequences, one object received more than half of the probability mass, therefore the precision of selecting the ground true object is based on very few sequences. However, in these instances, the selective approach outperformed the ranking based approach. Using selective prediction compared to argmax precision leads to a relative increase of precision by 10 - 115%. These differences seem to be larger between different paraphrase templates than between the datasets. In general the differences on PopQA are largest, followed by the hypernym dataset and T-REx.

Effect of Using Paraphrase Templates Between using no paraphrased templates and using 20 paraphrased templates, the coverage decreased between 0.4% for hypernym dataset and 55.6% for PopQA data, with similar changes for the two models. This answers research question 3[4] Precision increases consistently between 30% - 100%, where PopQA results increased most across the two models. The precision for GPT-2-L increases more than for Mistral-7B-Instruct-v0.2 when adding paraphrases on PopQA and hypernym dataset, but remains similar for T-REx data. Precision using argmax also increases from adding paraphrases: Compared to $P(o(r)|r, s(r))$-based precision, the increase on hypernym dataset is overproportional, and under proportional for PopQA and TRex. Selective precision vs. argmax prediction results are diverging more in settings where no paraphrase templates are used. Using paraphrases leads to a decrease in the relative difference of the precisions. The effect on argmax precision is smaller than on the selective prediction, and further varies per dataset as well. Overall, we can be fairly confident that the results of Mistral-7B-Instruct-v0.2, with all paraphrase

[4]Does the inclusion of paraphrased templates improve $P(o|s, r)$ both as a measure of certainty , and also in the prediction of factual correctness? (Chapter 1)

templates, are factually correct. At this threshold, the selective precision based on predictions from $P(o(r)|r, s(r))$ is at least 0.82 on all datasets.

Precision by Threshold A threshold of 0.5 to evaluate what LMs know indeed might be intuitive, however as the coverage is very small at this level, we further evaluate how the precision changes under a moving threshold. Given the distribution of $P(o(r)|r, s(r))$, a threshold of 1, leads to a coverage of close to 0. Coverage increases when decreasing the threshold and for $t = 0$, every value of the dataset is considered. This dependence is visualized in the following risk-coverage plots and can be summarized in one value by calculating the area under this curve (AUC).

Over all thresholds we obtain the following AUC values presented in Table 5.1 (detailed metrics per relation can be found in Table A.6). For visual representations, see Figure 5.4 and Figure A.1 for the equivalent visualization of GPT-2-L.

AUC Values by Dataset in Figure 5.4 show that Mistral-7B-Instruct-v0.2 performed best on the hypernym data, followed closely by the T-REx dataset. PopQA seemed to be the most challenging dataset to Mistral-7B-Instruct-v0.2. This holds true for GPT-2-L as well, but the AUC values are on average 10% smaller. This is in line with related work, as T-REx is generally considered to be in-domain data for LMs (as this data was encountered in pretraining), while PopQA is out-of-domain (the statements are not encountered in an equivalent way in pretraining). The influence of the number of paraphrase templates is smaller than expected from the large differences at the 0.5 threshold in Figure 5.3 and Section 5.2.1. Over all thresholds, the increase in AUC scores is 27 percent points for the hypernym dataset; for PopQA and T-REx AUC scores decreased by $3 - 4$ percent points. Thus, only for the hypernym dataset, the addition of paraphrases is beneficial to the selective precision. For GPT-2-L an increase occurs on all datasets, $2 - 3\%$ for PopQA and TRex data, and for the hypernym dataset the benefit on AUC by adding paraphrases is 36 percent points. This shows that GPT-2-L's sequence probabilities are more dependent on the different phrasings of the paraphrases, and it is not able to account for the fact that all paraphrases are semantically equivalent.

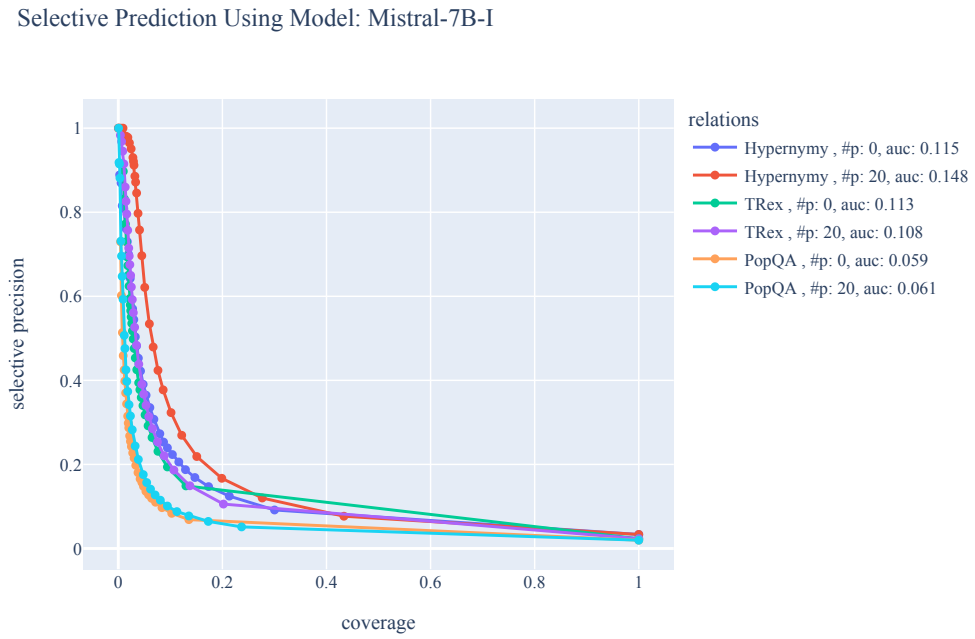


Figure 5.4: Precision-coverage plot, visualizing the dependence of the threshold (and by extension of the coverage) on precision. Evaluation is done over entire datasets, with all paraphrase templates (20) or 0—considering only the original relation template.

			AUC	Δ
	#p	0	20	
dataset	model			
Hypernymy	GPT-2-L	0.103	0.140	0.354
	Mistral-7B-I	0.115	0.148	0.279
PopQA	GPT-2-L	0.044	0.045	0.026
	Mistral-7B-I	0.059	0.061	0.038
TRex	GPT-2-L	0.091	0.094	0.024
	Mistral-7B-I	0.113	0.108	-0.044

Table 5.1: AUC values for selective prediction per dataset, model and number of paraphrase templates. The rightmost column shows the relative difference between the #p.

5.2.2 Per Relation

Having evaluated the amount of instances, where the LM indeed was correct in its predictions about the factuality of factual statements, we now use the same metrics and methods as before in Section 5.2.1, but carry out evaluation on a per-relation basis. In the following risk-coverage curves, we focus our analysis on configurations with the maximum number of paraphrases available. The full data is provided in Table A.6.

Results Recall-coverage curves for Mistral-7B-Instruct-v0.2 can be found in Figure 5.6 for PopQA data, and for the T-REx data in Figure 5.5. The hypernym data contains only one relation. For the sake of completeness, GPT-2-L results can be found in Figure A.6, and Figure A.7.

The Influence of Paraphrase Templates is dependent on the relation type. Using paraphrased relation templates can increase the selective precision values, or the paraphrases can also lead to decreased selective precision results. The relative differences range from -30% to +100%, see Table A.6. The differences furthermore vary depending on the LM but no clear trend is noticeable.

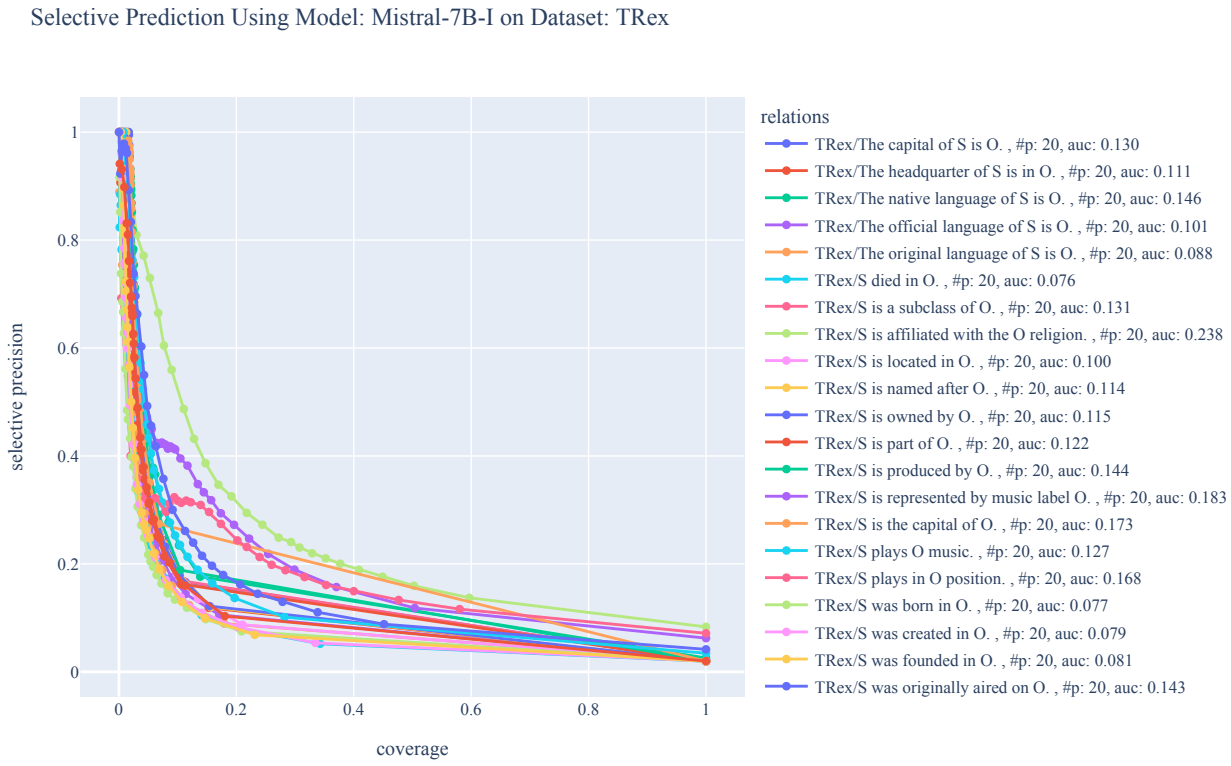


Figure 5.5: Classification performance trade-off for precision per T-REx relation. Every configuration uses the maximum number of paraphrase templates available (20 per relation). More detailed overviews can be found here: Table A.6.

Relations with Fewer Original Objects are Less Challenging By the precision AUC, Mistral-7B-Instruct-v0.2 on the individual relations of T-REx performs best on the religion relation, followed by the music label relation (Figure 5.5). On PopQA data the *color-of* relation is the outlier (Figure 5.6). These relations have in the original data fewer objects than most other relations (Table A.1). This is only natural as there exist fewer religions than e.g. capital cities. We compensated for this by sampling additional objects from other relations to achieve an equivalent set size of $(O(r)^{-})$. However, they are semantically further away from plausible objects associated with religions, and the semantically larger difference between the right vs. false objects seems to be well reflected in $P(o(r)|r, s(r))$, as these random false objects receive very little probability mass.

The Most challenging relations T-REx's born-in relation is most challenging with an AUC of 0.059. This is surprising as the $O(r)$ set of this relation contains only geographic objects, and not the year a person was born in. We cannot make a comparison of how well the model would perform if the objects were years as in the PopQA data, the **born-in** relation also does not contain numbers.

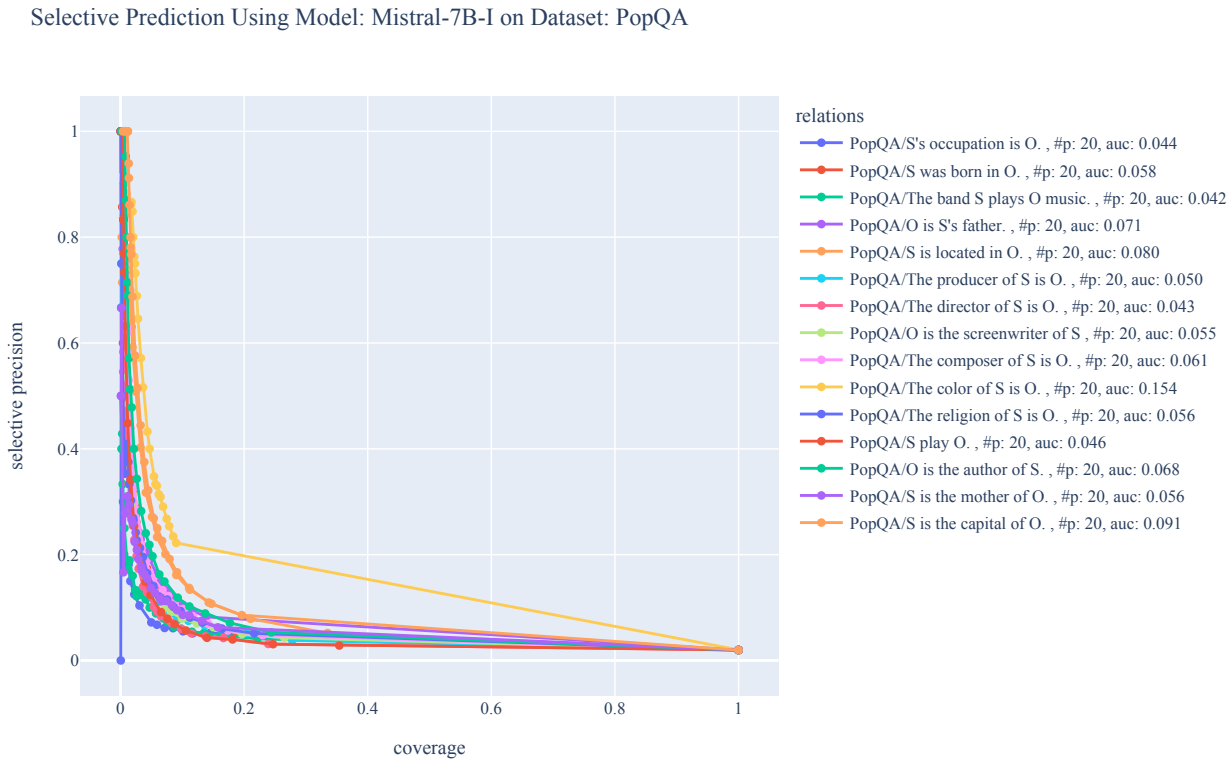


Figure 5.6: Classification performance trade-off for precision per PopQA relation. Every configuration uses the maximum number of paraphrase templates available (20 per relation). Detailed data can be found here: Table A.6.

5.2.3 Per Subject

In this section, we compare ground truth labels with the argmax object prediction of r, s, o triplets that received $P(o(r)|r, s(r)) > 0.5$. These are sequences where the LM attributed more probability to one specific object than to the rest. We first analyze instances where this approach has not selected the correct object.

Results

Incorrect Knowledge is Mostly Due to Close Similarities Between Two Objects. For the hypernym relation there are five sequences where the model's knowledge does not correspond to the ground truth labels. Tomatoes and cucumbers are predicted to be most likely fruits, but the ground truth label is vegetable. This is indeed a critical case as both are true depending on the context (botanically vs. common usage). The other cases are shoes that belong to the footwear category but are labeled as belonging to the clothing category), and trailers (home vs. vehicle).

In the T-REx data, most cases where the LM's knowledge was false occurred between two objects that share similarities. For example airplanes that were produced at that time by McDonnell Douglas are predicted to be produced by Boeing mistakenly. These two companies merged 30 years ago and continued under the name Boeing. A similar explanation can be found for the other examples.

Limitations of the Data come to light at this high confidence level, as the LM's judgements are very reliable and better than the labels. In some cases the data's labels are factually wrong, or ambiguous. Most of these cases can be found in the T-REx data. For example *Beijing is the capital of Taiwan*, *Kabul is the capital of Kabul*, or *Serbian is the official language of Albania.* These noisy instances are the downside of the automatic construction of the data.

Chapter 6

Experiment 2: Context Understanding

In Chapter 6, we evaluated a LM's parametric memory using our slot-filling probe in a "0-shot" fashion, as we expect a LM to be able to choose the object that makes a sequence factually correct without any instructions, and assign low probabilities to sequences that make no sense factually. However, it is possible that the parametric memory contains facts but these remain inaccessible by our probe as the task is not "understood" in the intended way thought slot-filling only. In this chapter, we explore whether additional context can help to clarify the task more, or to resolve ambiguities. Context can be additional instructions, demonstrations or supplemental information. In this chapter, we focus on (1), in-context learning (Section 6.1) and (2), context understanding abilities of LMs (Section 6.2). We evaluate these two abilities by providing either example sequences that demonstrate the relation, or additional information in the form of the subject's Wikipedia abstract, and analyzing the resulting changes in selective precision. In our setting of selective prediction, selective accuracy and precision are equivalent (see Section 4.1.7). This chapter answers research question 4[1].

6.1 In-context Learning Ability

3-shot Prompting We compare the 0-shot setting of Chapter 5 to a 3-shot setting. As additional context to the sequence of interest, we provide three sequences from the same relation, but with different subjects as demonstrations. From an improvement in precision, we can conclude that the LMs indeed exhibit in-context understanding abilities, as the additional context leads to an increasing performance in terms of factual knowledge prediction.

Negative Demonstrations As a control experiment, we include a version, where only factually wrong random examples are provided. In this setting, we expect a lower selective classification performance, as the model should not be able to gain additional knowledge, or even be put on a wrong path, from these random demonstrations. This control experiment with negative examples was only conducted on the hypernym dataset.

6.1.1 Results at a 0.5 Threshold

Figure 6.1 compares classification performance of $P(o(r)|r, s(r))$ using a classification threshold of 0.5 in two settings: using 0-shot vs. 3-shot prompting and 3-shot prompting with negative examples on the hypernym dataset. For a threshold-independent assessment, see Figure 6.2 and Table 6.1.

[1] **RQ4**: Can a positive influence be measured on certainty and factual correctness prediction when providing additional context? (Chapter 1)

Selective Precision Decreases with Demonstrations over all datasets and on two models by -25% for selective precision, and by -10% for rank-based precision. This is probably due to the larger coverages which overall increased by 52%. E.g., on T-REx data the coverage in absolute terms increased from 1743 to 2253. Nevertheless, coverage remains significantly smaller than 10%, also with the demonstrations. Selective precision decreases, as more values receive high probability scores, more sequences are classified as false positive and the number of ground true positive sequences remains constant. Thus, providing more context using exemplary demonstrations leads to more negative sequences receiving higher probability values. The additional context increased the LM's certainty in a specific $o(r)$, but this selection does not correspond to the ground truth label of the r, s pair.

Mistral-7B-Instruct-v0.2 Benefits From Demonstrations While GPT-2-L Does Not. Argmax-based precision increases between 4-6%, while GPT-2-L's argmax predictions decrease by 5-60%. However, for selective precision based on $P(o(r)|r, s(r))$ this not true. When providing demonstrations, selective precision decreases for both models at the 0.5 threshold.

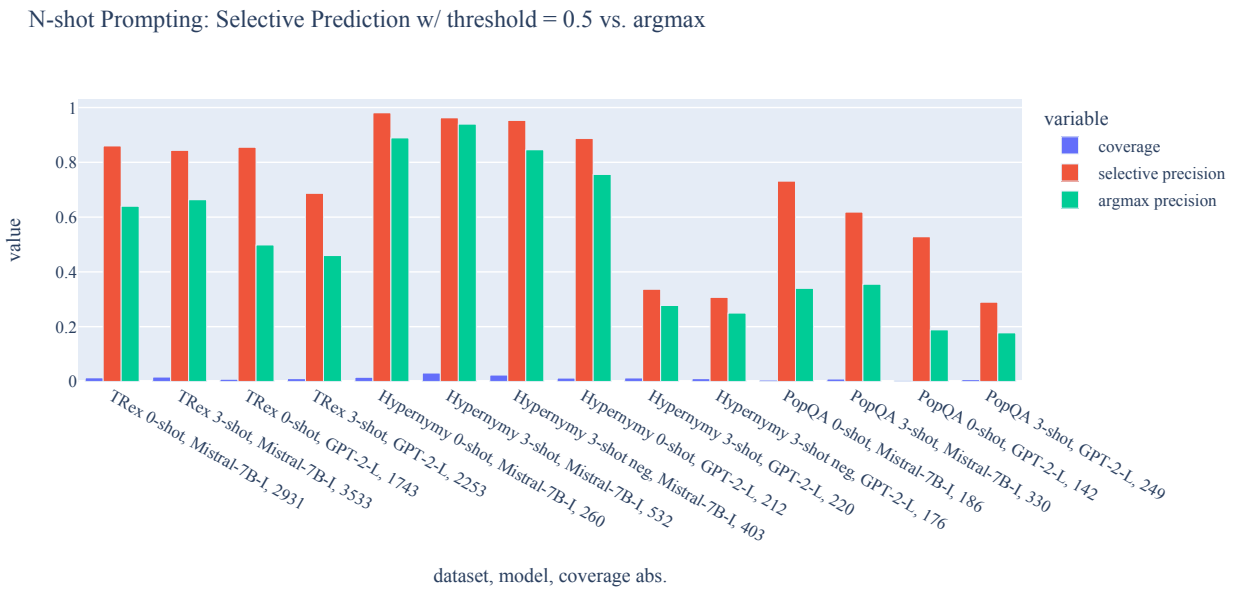


Figure 6.1: 0-shot vs. 3-shot prompting: In the 3-shot setting we provide additional sequences that illustrate the specific relation between different subjects and their objects. The classification threshold is 0.5. Additionally, on the hypernym dataset, we include a control run with unhelpful examples.

6.1.2 Threshold-independent Results

We notice similar results using a threshold independent analysis of precision as we did when using a threshold of 0.5 in the last Section. Using the area under the precision-coverage curve, a clearer picture emerges (Table 6.1): Mistral-7B-Instruct-v0.2 benefits from demonstrations, while GPT-2-L does not. Adding negative demonstrations leads to a decreased selective precision for GPT-2-L which is even smaller than if no examples were provided. Mistral-7B-Instruct-v0.2 benefits from demonstrations in both cases, but the increase in performance is larger using normal demonstrations compared to using random demonstrations.

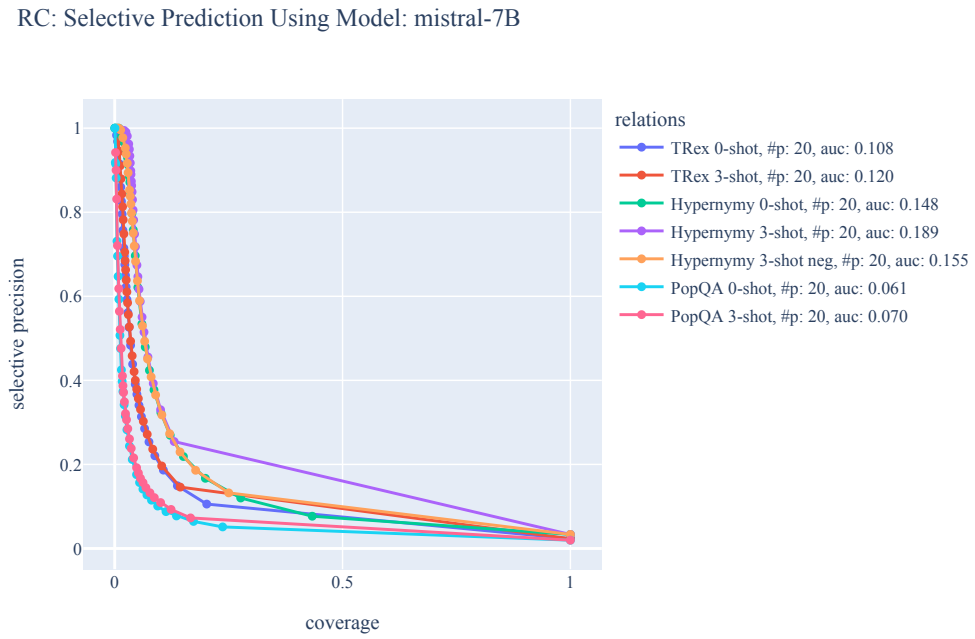


Figure 6.2: Classification performance trade-off per dataset, in the 0-shot and 3-shot setting.

		AUC			Δ
	run attributes	0-shot	3-shot	3-shot neg	0 vs. 3
dataset	model				
Hypernym	GPT-2-L	0.121	0.107	0.097	-0.120
	Mistral-7B-I	0.132	0.215	0.152	0.631
PopQA	GPT-2-L	0.045	0.039	NaN	-0.125
	Mistral-7B-I	0.060	0.071	NaN	0.185
TRex	GPT-2-L	0.092	0.076	NaN	-0.175
	Mistral-7B-I	0.111	0.125	NaN	0.130

Table 6.1: AUC values for selective prediction in 0-shot or 3-shot configuration, and for hypernym dataset also for negative 3-shot (control experiment).

Negative Demonstrations The evaluation at a threshold of 0.5, reveals that adding unhelpful demonstrations in our control experiment has the effect that coverage decreases by 1.5% on average. We note one increase in coverage (+55%) for Mistral-7B-Instruct-v0.2 on the hypernym dataset, when comparing 0-shot and 3-shot with negative examples. Note that coverage still remains very small. On the hypernym dataset, the effect on selective precision from using no demonstrations or three negative demonstrations is negative for argmax and selective precision for both LMs. However, compared to the other datasets, the effect on argmax and selective precision cannot be noticed from the individual differences per dataset and model, as there exist similar outliers, e.g. providing 3-shot demonstrations on the hypernym dataset restricts selective precision in the same amount for GPT-2-L.

Based on the area under the precision coverage curve, we achieve the desired results for both models (Table 6.1). AUC shows that providing negative demonstrations hurts precision compared to using valuable demonstrations.

6.1.3 Results by Relation

In this section, we investigate how the LM performed on each individual relation, and whether we can detect in-context learning abilities that result from the provided demonstrations in the context. Results on PopQA data are provided in Figure 6.3 and T-REx relations here: Figure 6.4. Detailed results per relation and number of paraphrases can be found in Table A.7.

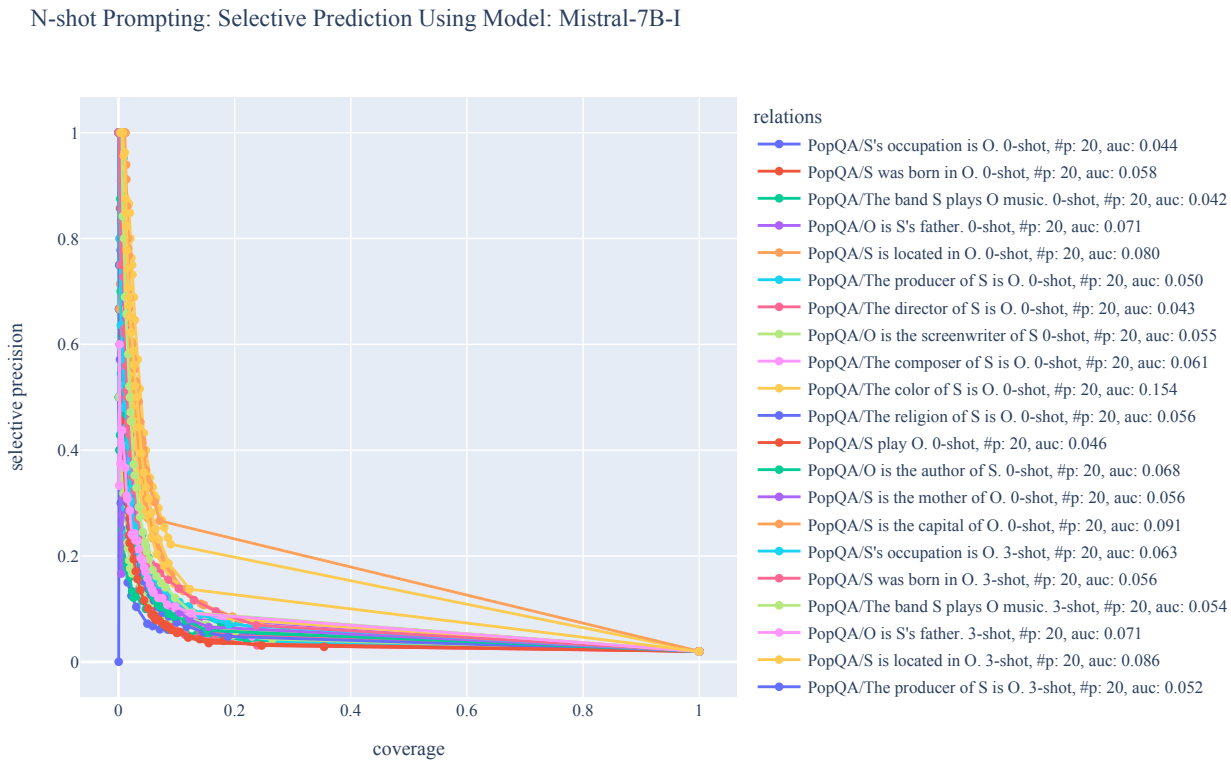


Figure 6.3: Classification performance trade-off for each relation of PoPQA, in the 0-shot and 3-shot setting.

In Figure 6.3 and Figure 6.4, we observe that Mistral-7B-Instruct-v0.2's relative precision improvements from adding three demonstrations ranges between 187% and -18%.

Relations Where Selective Precision of Mistral-7B-Instruct-v0.2 Improved Most are PopQA 's relations *S play O.* and *S's occupation is O.* The improvements of +187% and +137% are only achieved by not using paraphrase templates. The same applies to T-REx's relations where most improvement is achieved: *S is affiliated with the O religion.* and *S plays in O position.* Other relations of LAMA T-REx with improvements larger than 70 percent points include: *S is part of O*, *S is named after O*, *S is a subclass of O.*

Relations Where Selective Precision of GPT-2-L Improved Most are similar to Mistral-7B-Instruct-v0.2, however they were achieved using paraphrases, in contrast to Mistral-7B-Instruct-v0.2 which was not able to improve as much with paraphrases. On PopQA, besides

the relations where Mistal improved by the largest margin, GPT-2-L improved also on *The religion of S is O*. Improvements range between 29 - 122 percent points.

Relations Where Mistral-7B-Instruct-v0.2 Could not Benefit From Demonstrations are PopQA's relations *S was born in O*, *The producer of S is O*, *The director of S is O*, and T-REx relations *The original language of S is O*, *The native language of S is O*, *S plays in O position*, and *S was born in O* with declines in selective precision between 4 - 18 percent points.

Relations Where GPT-2-L Could not Benefit From Demonstrations are PopQA's relations *S is the mother of O*, *S plays in O position S is part of O*, and *S is the capital of O* with declines in selective precision of 42 - 63 percent points. These relations are not the same ones where Mistral-7B-Instruct-v0.2 was influenced negatively due to the added demonstrations. On T-REx however, the most challenging relation is *The original language of S is O* for both models. For GPT-2-L, relations where demonstrations led to declining precision are additionally: *The headquarter of S is in O*, *S is named after O*, and *S died in O*. Interestingly, these declines in performance occur exclusively in the setting where no paraphase templates are used.

Reasons for an overproportional improvement in this setting cannot be attributed to relations that contain fewer subjects or original objects. Firstly, because all relations, with the exception of the religion- and play-relation, contain more than 50 subjects and objects, and secondly, the quality of the reference objects has no influence on this relative increase as we keep the sets equally between experimental configurations.

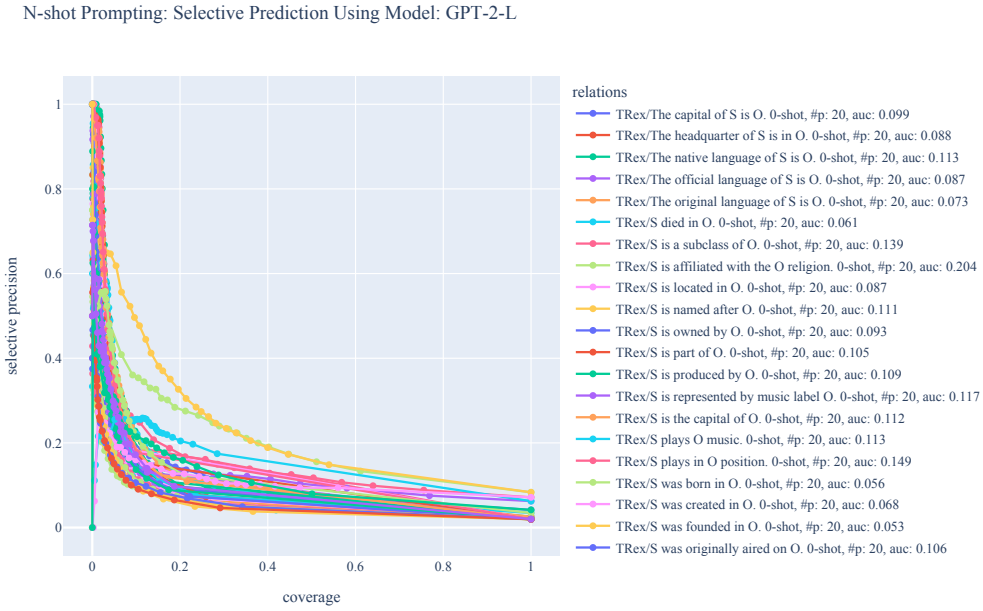


Figure 6.4: Classification performance trade-off for each relation of T-REx, in the 0-shot and 3-shot setting.

6.1.4 Changes of Selective Precision on Subject Level

Table 6.2 makes a comparison between the standard 0-shot setting and each run attribute modification, e.g. adding 3-shot demonstrations, or adding negative demonstrations. Changes from incorrect to correct predictions or vice versa are measured using selective precision. Hence,

changes to correct (from incorrect) are due to precision increasing above the 0.5 threshold. However, it does not mean the model's argmax prediction was incorrect before, but that the degree of certainty improved—the same applies for the shift of certainty from correct to incorrect predictions.

The fewest changes with approximately 20% occur on PopQA and T-REx data. For the hypernym dataset, the amount of sequences with changing predictions are between 30% and 50%. Sequences where the object was changed to a correct prediction due to positive demonstrations occurred more often for Mistral-7B-Instruct-v0.2 (31%) than for GPT-2-L (6%). Changes from correct to incorrect with positive demonstrations occurred more often for GPT-2-L (12%) compared to Mistral-7B-Instruct-v0.2 (3%). On the hypernym dataset, Mistral-7B-Instruct-v0.2 was most sensitive to negative demonstrations, in 25% of the cases there have been changes from predictions to incorrect predictions. GPT-2-L showed fewer changes: 11%.

		to correct		to incorrect		no changes	
	run	3-shot	3-shot-neg	3-shot	3-shot-neg	3-shot	3-shot-neg
dataset	model						
Hypernymy	GPT-2-L	0.054	0.076	0.251	0.111	0.695	0.813
	Mistral-7B-I	0.456	0.031	0.010	0.253	0.534	0.716
PopQA	GPT-2-L	0.040	-	0.044	-	0.917	-
	Mistral-7B-I	0.121	-	0.029	-	0.850	-
TRex	GPT-2-L	0.089	-	0.078	-	0.833	-
	Mistral-7B-I	0.141	-	0.051	-	0.808	-

Table 6.2: Comparison 0-shot vs. the run attibute modification of 3-shot or 3-shot negative. Values denote the relative amount of sequences that changed predictions.

Examples of Sequences We analyze a subsection of sequences of the hypernym dataset individually at a threshold of 0.5 and use all 20 paraphrase templates for selective prediction. Interactive visualizations for all models and datasets are provided in the GitHub repository accompanying this work.

Examples of Sequences Where Demonstrations Lead to Correct Predictions can be factually wrong predictions that were corrected with additional context such as the subject *Piccolo*, for which the prediction without examples was *music*, with examples the most likely object became *instrument*. Another example is *mother* which was incorrectly predicted to be a *home* category, with examples this changed to the correct category, *relative*. Also present are ambiguous subjects where both categories satisfy the hypernym relation such as *quartz* which was predicted to belong to the hypernym *time*, with examples this changed to *gem*. However, in most of the instances where demonstrations lead to a selective precision above the 0.5 threshold, the changes are restricted to an increase in $P(o(r)|r, s(r))$, and not due to a change of the argmax object.

Examples of Sequences Where Demonstrations Lead to Incorrect Predictions can be found in Table 6.3 These subjects all exhibit a degree of ambiguity making it challenging

to predict the correct hypernym. In these cases, the demonstrations could not help resolving the ambiguities.

	hypernym	
hyponym	**prediction**	**reference**
Penguin	animal	bird
rape	vegetable	crime
country	country	music
guppy	animal	fish

Table 6.3: Examples where demonstrations lead to incorrect predictions of the hypernym.

6.2 Reading Comprehension (RC)

Figure 6.5 compares classification performance of $P(o(r)|r, s(r))$ using a threshold of 0.5 in two settings on the hypernym data: With additional context on the subject and in the standard 0-shot setting, without any context. As a control experiment, a setting is added where additional context on a random subject is provided, which does not correspond to the subject in the sequence. Results over thresholds are presented in Table 6.4 and precision-coverage plots are provided in Figure 6.6. We do not conduct this experiment on PopQA and T-REx data.

Context for RC Task For each hyponym in the hypernym data, we probe the LM to predict which hypernym satisfies the "hypernym relation". Instead of confronting the LM directly with factually correct and incorrect sequences such as textitA diamond is a type of gem", *A diamond is a type fruit.*, we provide additional information about the subject, *diamond* in this example. We obtain this information from Wikipedia abstracts using the Wikipedia package. We make sure the abstracts' lengths are smaller than the context size of the LMs we are using, by mining only the first three sentences of the abstract. We resolve cases manually where there exist multiple articles with the same title as the subject. In 93.03% or 534 / 574 cases the true object occurs unmodified in the abstract, essentially providing the answer. In the remaining cases a synonym of the object is present.

6.2.1 Results

Context influences LMs differently Compared to the standard, 0-shot setting, adding context influences precision results differently per model. Mistral-7B-Instruct-v0.2's performance both in terms of selective precision at a 0.5 threshold, and argmax precision, is -1.3% respective -0.4% lower when given additional context about a subject. Precision using GPT-2-L is increased by 2.3%, and even by 13.5% for argmax precision. Additional context helped increase performance for the smaller model, while a large model like Mistral-7B-Instruct-v0.2 was not able to improve on factual precision. In absolute terms, Mistral-7B-Instruct-v0.2 outperformed GPT-2-L in all configurations.

Irrelevant Contexts Degenerate Selective Precision in the control setting, leading to a decrease in selective precision for both models. The average selective precision decreased by 51 percent points, compared to the standard 0-shot setting. Precision argmax decreased by

57%. The decrease in performance for the control experiment is consistently larger than the improvement due to adding contexts. The influence on the results for GPT-2-L are larger than for Mistral-7B-Instruct-v0.2.

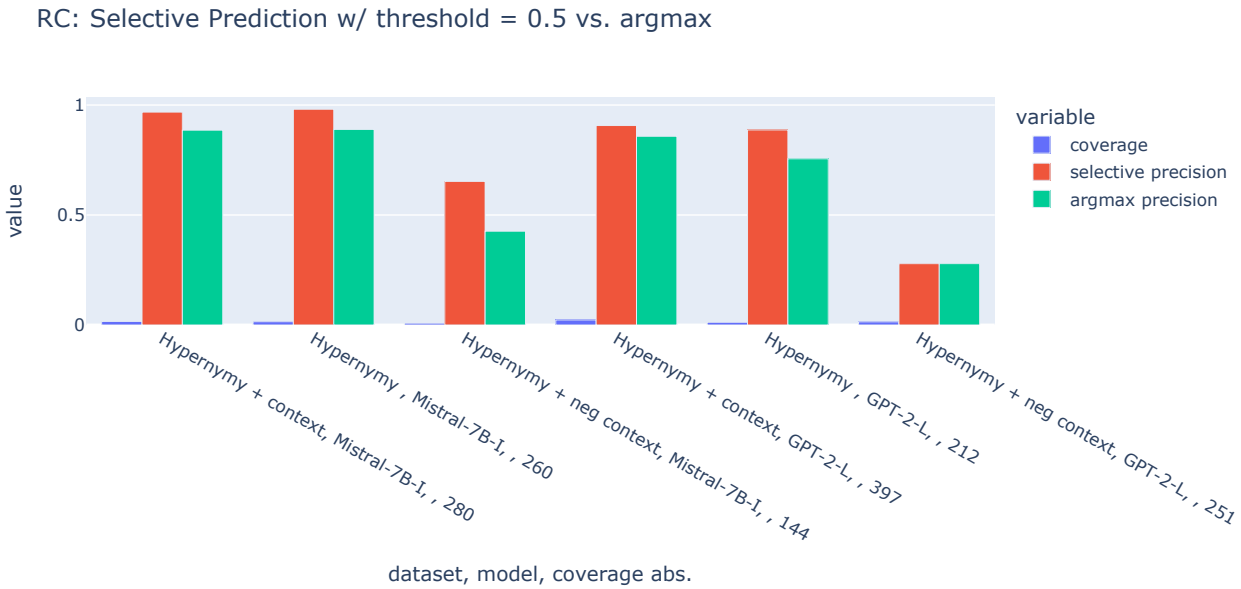


Figure 6.5: Comparison of selective precision results (at a threshold of 0.5) with context vs. without context. "neg context" denotes the control experiment, where unhelpful context is provided.

A threshold-independent analysis is provided using the area under the precision risk curves in Figure 6.6 and Table 6.4. Providing correct context about the sequence's subject led to an increase in precision AUC of around 15%, while it led to a decrease in precision AUC in the range of 8 to 18% when random contexts were provided. The context had more influence on the precision AUC results of the GPT-2-L model, whereas the Mistral-7B-Instruct-v0.2 results changed less.

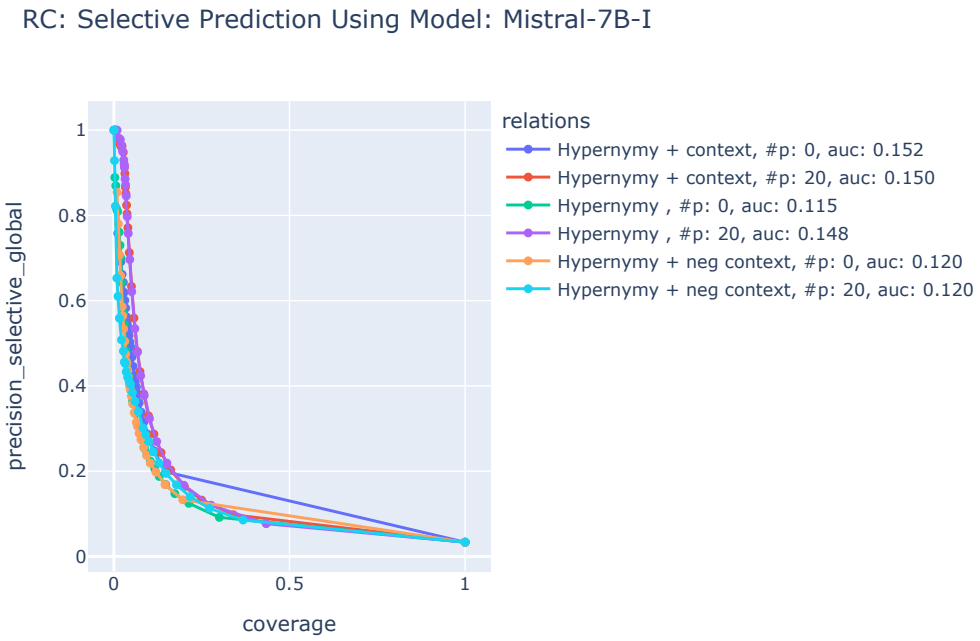


Figure 6.6: Precision of $P(o(r)|r, s(r))$ vs. coverage, for each configuration on the hypernym dataset: 0-shot, 3-shot and 3-shot with negative examples. A higher AUC score indicates that a configuration leads to an improvement in selective precision across thresholds. Each configuration is available with all available paraphrases (20) or without paraphrased templates.

		AUC			Δ	
	run	+ c	+ neg c	0-shot	+ c	+ neg c
dataset	model					
Hypernym	GPT-2-L	0.143	0.099	0.121	0.178	-0.183
	Mistral-7B-I	0.151	0.120	0.132	0.147	-0.089

Table 6.4: AUC values for selective prediction on the hypernym dataset in the standard setting (0-shot), with context (c) information about the subject, and with a random subject (neg c) provided as control run. Δ denotes the relative difference of AUC values between the 0-shot setting and the run attribute modification.

6.2.2 Changes on Subject Level

Table 6.5 provides a subject-level analysis of the changes in $P(o(r)|r, s(r))$ that occured as a result of the correct or incorrect contexts provided in the same way as in Section 6.1.4. Changes from incorrect to correct predictions or vice versa are measured using selective precision at a threshold of 0.5. In two thirds of the sequences, no change in selective precision occurred by adding relevant or irrelevant contexts. For GPT-2-L, adding negative contexts changed fewer predictions than for positive contexts, the opposite is observable for Mistral-7B-Instruct-v0.2.

More changes occur in the direction that the context supports. Thus, adding supportive context, generates more instances where selective precision increases above the threshold (and was

below it before adding context). Adding irrelevant context increases the cases where selective precision changed from above the threshold to below. These changes occur in $P(o(r)|r, s(r))$, and not necessarily on the ranking of the argmax object.

For both negative and positive contexts, more instances are corrected in the direction from correct to incorrect for Mistral-7B-Instruct-v0.2. GPT-2-L benefits more from correct contexts, as it is able to change previously wrong predictions to right predictions.

		to correct		to incorrect		no changes	
	run	+ c	+ neg c	+ c	+ neg c	+ c	+ neg c
dataset	model						
Hypernymy	GPT-2-L	0.315	0.028	0.017	0.232	0.667	0.740
	Mistral-7B-I	0.173	0.021	0.146	0.300	0.681	0.679

Table 6.5: Comparison 0-shot vs. run attibute modification (correct context (c) vs. negative context). Values denote relative amouont of changed sequences.

Examples of Sequences We analyze a subsection of sequences individually at a threshold of 0.5 and use all 20 paraphrase templates for selective prediction on the hypernym dataset. All interactive visualizations for all models and datasets are provided in the GitHub repository.

Examples of Sequences Where Context About the Subjects Lead to Correct Predictions occurred often for ambiguous subjects where the correct hypernym can be different depending on the object. Some of these examples can be found in Table 6.6. The examples above the horizontal line all are ambiguous nouns, where the model was able to benefit from context about the subject in order to resolve the ambiguity. In the bottom part of Table 6.6, non-ambiguous hyponyms are presented. For no obvious reason, many of the incorrect predictions without context include time. However, by considering context about the subject, the correct hypernym was found.

	hypernym	
hyponym	**prediction**	**reference**
drill	music	tool
quarter	time	money
club	sport	weapon
ferry	vehicle	boat
armoire	clothing	furniture
nail gun	weapon	tool
cabin	vehicle	home
yard	time	distance
nail	footwear	tool

short	time	clothing
meter	time	distance
sander	time	distance
private	time	soldier

Table 6.6: Examples where predictions are false without context, and then with the addition of more information about the hyponym changed to the reference hypernym. Hyponyms in the first part of the table are ambiguous, while the hyponyms below the horizontal line are not.

Examples of Sequences Where Context About the Subjects Lead to Incorrect Predictions are due to ambiguous hyponyms. Examples are presented in this format: subject: correct prediction according to labels, incorrectly changed prediction of the hyponym due to the context. All these changes are factually still correct, but the dataset fails to take into account ambiguities. E.g., country: country vs. music; condominium: music vs. home; zebra: music vs. animal. Condominium and Zebra are also the name of bands.

Chapter 7

Discussion

7.1 Limitations

No Full-scale Evaluation on T-REx and PopQA We restricted the upper bound of subjects per relation to 50, as it would not be computationally feasable to conduct our experiments on all available subjects, which range between 5 - 1,500 depending on relation and dataset, we instead opted to extend the sizes of $O(r)$ to 50 and number of paraphrase templates per relation ($|T(r)|$) to 20—which is more valuable and allows us to better investigate our research questions. However, due to the limitation in terms of subjects per relation we cannot make a valid comparison to the results of related work.

Random Processes in Sampling of Subjects and $O(r)$ As explained in more detail in Section 4.1.3 and Section 4.1.3, we sample 50 random subjects per relation, and also 50 negative objects per relation. This stochasticity can influence results, as the factual knowledge of a LM might not be universally equal for each subject and object contained in a relation. We counteract this issue to some degree by keeping these sets fixed for every experiment in this work. This still allows us to make deductions about how additional context influences the probabilities attributed to the permuted sequences. Nonetheless, a multitude of repeated sampling and evaluation are needed to obtain perfect results.

Different Semantic Distances of Objects in $O(r)$ In cases where there are fewer objects in a given relation of the data, we included samples from other relations. It can therefore be the case that for some relations (those that contain fewer than 50 objects) are less challenging for the LM, as the substitutes from the foreign relations could be more obviously wrong due to a greater semantic distance from the r, s pair.

Restrictions of the Datasets Facts are not universally valid, and they can change frequently. E.g., Obama is the president of the United States; this statement was valid for eight years, but not anymore. It can be that such statements are not updated in the datasets, and especially also not in the LMs parametric knowledge. Other limitations include noisy facts contained in LAMA T-REx that we discussed in Section 5.2.3. Due to the nature of automatically compiled data, cases like these will always be a limitation.

7.1.1 Qualitative Evaluation of Filled-in Templates

A considerable source of error was identified to be that the grammar post-processing tool modified the subject and object. However, subject and object needed to be filled in and could not just be placeholders, as the overall sequence is dependent on them and the grammar checker

requires context as well. E.g. determiners need to be adapted to the specific subject: A car, not an car. This would require a hybrid approach, specifically accepting modifications to the original sequence only if the change has no effect on subject nor object. The exact amount of limitations included by this is examined in more detail on the full-scale datasets in Section 7.1.1.

10 random paraphrase templates per relation for T-REx and PopQA, 10 subjects per unique object for the hypernym dataset, leads to a total of 200 sequences for hypernymy, 344 for LAMA T-REx, and 300 for PopQA. These were judged by whether the resulting sequences were clear and grammatically correct. Instances where subjects or objects were modified by the grammar tool are not annotated here, as this is done automatically on the full scale dataset. For the subsets of PopQA 91% of the sequences were judged to be good, 88% for LAMA T-REx, and 87% for the hypernym dataset.

Other sources of error include sequences that are not perfectly correct grammar wise, although their influence is limited, as the general sense is still clear. E.g., "*Spain is affiliated with Free Will Baptist as their religion.*". Omitted determiners and verb-noun agreement are an issue in cases where the subject is in the plural form, but the verb is 3rd person singular. Also problematic are relations that have persons as subject and e.g. companies, in these cases there is a which / who confusion, e.g. *Kris Kristofferson, which is represented by Monument.*.

Ambiguous paraphrases, e.g., "*Rob Cohen is the individual behind the production of Reporter.*" are a further issue. Additionally, punctuation, a missing subject placeholder (in case of one paraphrase template), and generally artificial sounding paraphrases, appear as issues. These are all induced by the automatic paraphrase generation method used in this work.

Post-processing can affect Subject and Object In some cases, subjects or objects in the filled-in templates were modified by the grammar checking tool. The relative effects denoted here are over all permutations. The least observed effect was in the hypernym dataset, where 1.5% of the sequences' subjects were slightly modified. No objects were affected. The effects were more relevant in both PopQA and T-REx data: Approximately 30% of the subjects were affected, and 25% of PopQA's objects, respectively approximately 1% of objects for T-REx. This occurs mostly when the subject or object was in a foreign language and resembled English words closely, e.g. *Ozren Perić* was modified to *Often Eric*.

Chapter 8

Conclusion

This work has provided an overview of methods to probe encoder-only LMs, such as BERT on task of masked word prediction to probe the LM for its factual knowledge, and has highlighted the challenges of probing decoder-only LMs such as GPT-2-L on open-ended tasks such as question answering. Lastly, probing methods have been studied that can be applied on the internal representations of LMs, but these also do not allow a reliable estimation whether a representation of a fact exists in LMs, as the assigned sequence probabilities are not indicative of the LM's knowledge.

The omnipresent challenge of prompting is that paraphrased prompts can be understood differently by the LM and therefore, many prompts need to be tested in order to make deductions about the knowledge of LMs.

We have reframed the factual probing task as a probabilistic inference problem which allows us to address both challenges, by normalizing the sequence probabilities over paraphrases and equivalence sets consisting of objects per relation template. Evaluation of a subset of T-REx and PopQA allowed us to investigate how two LMs estimate their factual confidence, which we then evaluated using selective prediction. We provided a high level overview, per dataset, and more detailed analysis on the relation level and on the level of individual facts answering our four research questions, most notably whether we can use the probabilities assigned to the reference objects as an indication of factual certainty (Chapter 1).

Overall, we can say that a high confidence for an object is not indicative of factuality in both LMs (GPT-2-L, and Mistral-7B-Instruct-v0.2). Depending on the dataset an attribution of a probability of an object larger than 0.5 only occurred in $0.8 - 1.7\%$ of all facts. In these examples, where the model attributed the largest part of the probability to one specific object, the sequence was factually true in $73-98\%$ of the cases for Mistral-7B-Instruct-v0.2 independent of whether paraphrases were used or not.

Additionally to parametric knowledge, we have applied probabilistic prompting to in-context understanding tasks such as reading comprehension and few-shot prompting. Additional demonstrations allowed Mistral-7B-Instruct-v0.2 to perform better in terms of selective precision, GPT-2-L performed worse. On the reading comprehension task, both models benefited from additional context. These results indicate that additional context, in addition to the prompt, can lead to a better 'understanding' of what is asked of a LM.

Future Work As we have discussed in the limitations (Chapter 7), a larger scale investigation is required, in order to reliably estimate factual precision over entire datasets and also be able to make comparisons to related work.
What could be interesting to further evaluate is how the creation of equivalence sets, $O(r)$, influence a LMs confidence. For example, by also considering the individual subject of a fact, specifically challenging sets of objects could be generated. E.g., for *Rome* and the *capital-of*

relation, it would be interesting to assess how the probabilities would be distributed among other Italian cities. Such cases were not present in our object sets.
Secondly, as the sensitivity to paraphrases is large, and the selection of the true object in our generated equivalence classes is rarely attributed a probability significantly larger than to other objects in this set, it could be interesting to investigate the following two aspects further. (1), how can an adapted pre-training objective that accounts for consistency in terms of the predicted object over paraphrased templates improve the reliability of knowledge prediction, and (2) could a contrastive approach be used to better select the correct object from the equivalence sets?

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

William F. Battig and William Edward Montague. 1969. Category norms of verbal items in 56 categories a replication and extension of the connecticut category norms. *Journal of Experimental Psychology*, 80:1–46.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Hila Gonen, Srini Iyer, Terra Blevins, Noah Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148, Singapore. Association for Computational Linguistics.

Michael Hanna and David Mareček. 2021. Analyzing BERT's knowledge of hypernymy via prompting. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 275–282, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060, Singapore. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Linhao Luo, Trang Vu, Dinh Phung, and Reza Haf. 2023. Systematic assessment of factual knowledge in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13272–13286, Singapore. Association for Computational Linguistics.

Matéo Mahaut, Laura Aina, Paula Czarnowska, Momchil Hardalov, Thomas Müller, and Lluís Màrquez. 2024. Factual confidence of LLMs: On reliability and robustness of current estimators.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

OpenAI. 2024. GPT-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020a. How context affects language models' factual predictions.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020b. How context affects language models factual predictions.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Ella Rabinovich, Samuel Ackerman, Orna Raz, Eitan Farchi, and Ateret Anaby Tavor. 2023. Predicting question-answering performance of large language models through semantic consistency. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 138–154, Singapore. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Saket Sharma, Aviral Joshi, Yiyun Zhao, Namrata Mukhija, Hanoz Bhathena, Prateek Singh, and Sashank Santhanam. 2023. When and how to paraphrase for named entity recognition? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7052–7087, Toronto, Canada. Association for Computational Linguistics.

Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. 2024. A thorough examination of decoding methods in the era of LLMs.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV au2, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts.

Jakub M Tomczak. 2022. *Deep Generative Modeling*. Springer Nature.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

James P Van Overschelde, Katherine A Rawson, and John Dunlosky. 2004. Category norms: An updated and expanded version of the battig and montague (1969) norms. *Journal of Memory and Language*, 50(3):289–335.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

BigScience Workshop. 2023. BLOOM: A 176b-parameter open-access multilingual language model.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*.

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. 2023. Give me the facts! a survey on factual knowledge probing in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605, Singapore. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Appendix A

Appendix

A.1 Dataset Statistics

A.1.1 Full Datasets

relation	#o	#s	$\lvert O^- \rvert$	#p	#pos	#neg	total
S is located in O.	516	3679	0	0	3745	0	3745
S plays O music.	29	929	0	0	931	0	931
S was founded in O.	237	936	0	0	936	0	936
S is affiliated with the O religion.	12	467	0	0	473	0	473
The official language of S is O.	75	774	0	0	966	0	966
S plays in O position.	14	952	0	0	952	0	952
The headquarter of S is in O.	282	965	0	0	967	0	967
S was born in O.	325	944	0	0	944	0	944
S is a subclass of O.	434	962	0	0	964	0	964
S died in O.	236	953	0	0	953	0	953
The native language of S is O.	39	976	0	0	977	0	977
S is part of O.	331	888	0	0	932	0	932
S was created in O.	69	908	0	0	909	0	909
S is the capital of O.	218	225	0	0	234	0	234
S is represented by music label O.	16	421	0	0	429	0	429
S is named after O.	413	622	0	0	645	0	645
S is produced by O.	52	970	0	0	982	0	982
The original language of S is O.	51	850	0	0	856	0	856
S is owned by O.	237	687	0	0	687	0	687
S was originally aired on O.	24	876	0	0	881	0	881

S was written in O.	53	859	0	0	877	0	877
The capital of S is O.	380	692	0	0	703	0	703

Table A.1: Statistics for the original LAMA T-REx data. #o denotes the number of unique objects in a relation, #s the number of unique subjects, $|O(r)^-|$ the sampled number of unique objects in the reference set $O(r)^-$, which make the sequence factually false, #p the number of paraphrase templates per relation, #pos the resulting number of sequences that are factually correct, #neg the number of incorrect sequences, and total denotes the total amount of all factually false and true sequences per relation.

relation	#o	#s	$\lvert O^- \rvert$	#p	#pos	#neg	total
S's occupation is O.	84	531	0	0	532	0	532
S was born in O.	434	584	0	0	584	0	584
The band S plays O music.	283	1530	0	0	1619	0	1619
O is S's father.	524	562	0	0	570	0	570
S is located in O.	107	824	0	0	838	0	838
The producer of S is O.	1133	1320	0	0	1520	0	1520
The director of S is O.	1521	1784	0	0	1999	0	1999
O is the screenwriter of S	1468	1671	0	0	1999	0	1999
The composer of S is O.	624	880	0	0	978	0	978
The color of S is O.	5	34	0	0	34	0	34
The religion of S is O.	70	326	0	0	338	0	338
S play O.	49	547	0	0	547	0	547
O is the author of S.	1158	1408	0	0	1514	0	1514
S is the mother of O.	173	187	0	0	187	0	187
S is the capital of O.	584	622	0	0	645	0	645

Table A.2: Statistics for the original PopQA data. #o denotes the number of unique objects in a relation, #s the number of unique subjects, $|O(r)^-|$ the sampled number of unique objects in the reference set $O(r)^-$, which make the sequence factually false, #p the number of paraphrase templates per relation, #pos the resulting number of sequences that are factually correct, #neg the number of incorrect sequences, and total denotes the total amount of all factually false and true sequences per relation.

relation	#o	#s	$\lvert O^- \rvert$	#p	#pos	#neg	total
S is a O.	30	574	0	0	577	0	577

Table A.3: Statistics for the original hypernym data. #o denotes the number of unique objects in a relation, #s the number of unique subjects, $|O(r)^-|$ the sampled number of unique objects in the reference set $O(r)^-$, which make the sequence factually false, #p the number of paraphrase templates per relation, #pos the resulting number of sequences that are factually correct, #neg the number of incorrect sequences, and total denotes the total amount of all factually false and true sequences per relation.

A.1.2 Permutation Stats

relation	#o	#s	$\lvert O^- \rvert$	#p	#pos	#neg	total
S is located in O.	86	50	50	22	1100	54890	55990
S plays O music.	49	50	49	22	1100	53900	55000
S was founded in O.	78	50	50	22	1100	54604	55704
S is affiliated with the O religion.	49	50	49	22	1100	53900	55000
The official language of S is O.	59	47	50	22	1100	54384	55484
S plays in O position.	48	50	49	22	1100	53746	54846
The headquarter of S is in O.	79	50	50	22	1100	54670	55770
S was born in O.	84	50	50	22	1100	54802	55902
S is a subclass of O.	91	50	50	22	1100	54978	56078
S died in O.	81	50	50	22	1100	54890	55990
The native language of S is O.	50	50	49	22	1100	53900	55000
S is part of O.	79	49	50	22	1100	54934	56034
S was created in O.	55	50	50	22	1100	54076	55176
S is the capital of O.	90	49	50	22	1100	54846	55946
S is represented by music label O.	50	48	49	22	1100	53900	55000
S is named after O.	93	45	50	22	1100	54890	55990
S is produced by O.	52	50	50	22	1100	53988	55088
The original language of S is O.	50	50	49	22	1100	53900	55000
S is owned by O.	69	50	50	22	1100	54626	55726
S was originally aired on O.	49	50	49	22	1100	53900	55000
S was written in O.	52	50	50	22	1100	53944	55044
The capital of S is O.	92	50	50	22	1100	54846	55946

Table A.4: Statistics for the original LAMA T-REx data. #o denotes the number of unique objects in a relation, #s the number of unique subjects, $|O(r)^-|$ the sampled number of unique objects in the reference set $O(r)^-$, which make the sequence factually false, #p the number of paraphrase templates per relation, #pos the resulting number of sequences that are factually correct, #neg the number of incorrect sequences, and total denotes the total amount of all factually false and true sequences per relation.

relation	#o	#s	$\lvert O^- \rvert$	#p	#pos	#neg	total
S's occupation is O.	56	50	50	21	1050	51639	52689
S was born in O.	94	50	50	21	1050	52395	53445
The band S plays O music.	84	50	50	21	1050	52374	53424
O is S's father.	95	50	50	21	1050	52395	53445
S is located in O.	60	50	50	21	1050	52164	53214
The producer of S is O.	97	50	50	21	1050	52437	53487
The director of S is O.	98	50	50	21	1050	52458	53508
O is the screenwriter of S	96	50	50	21	1050	52437	53487
The composer of S is O.	93	50	50	21	1050	52395	53445
The color of S is O.	50	34	49	21	714	34986	35700
The religion of S is O.	58	49	50	21	1050	52080	53130
S play O.	50	50	49	21	1050	51450	52500
O is the author of S.	98	50	50	21	1050	52458	53508
S is the mother of O.	82	50	50	21	1050	52143	53193
S is the capital of O.	94	49	50	21	1029	51366	52395

Table A.5: Statistics for the original PopQA data. #o denotes the number of unique objects in a relation, #s the number of unique subjects, $|O(r)^-|$ the sampled number of unique objects in the reference set $O(r)^-$, which make the sequence factually false, #p the number of paraphrase templates per relation, #pos the resulting number of sequences that are factually correct, #neg the number of incorrect sequences, and total denotes the total amount of all factually false and true sequences per relation.

A.2 Results per Dataset

A.2.1 Classification of with Varying Thresholds

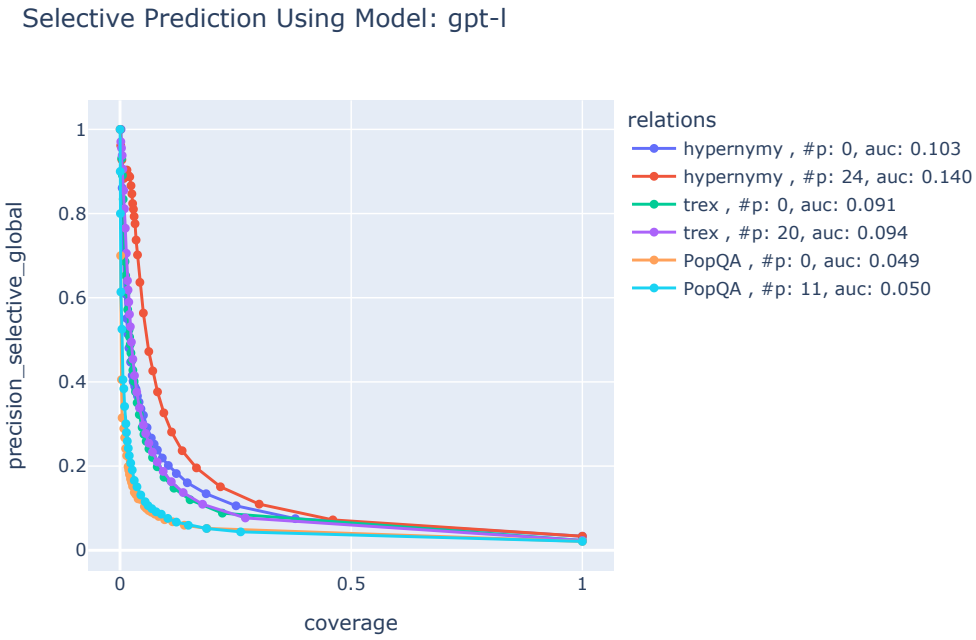


Figure A.1: Precision of P(O—T(r)) vs. coverage.

A.3 Class Separation

A.3.1 Ranks Assigned to True and False Sequences

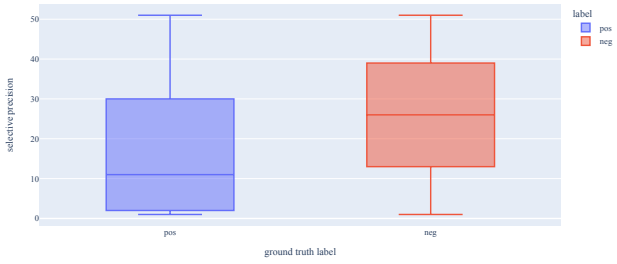


Figure A.2: Ranks assigned to P(o—t(r)_i) vs. ground truth label, evaluated over the entire PopQA dataset. Using Mistral-7B-Instruct-v0.2.

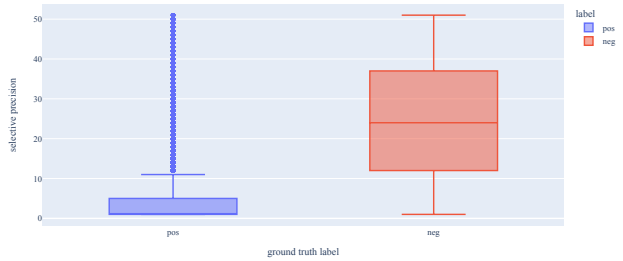


Figure A.3: Ranks assigned to P(o—t(r)_i) vs. ground truth label, evaluated over the entire T-REx dataset. Using Mistral-7B-Instruct-v0.2.

A.3.2 Hypernym Class Separation

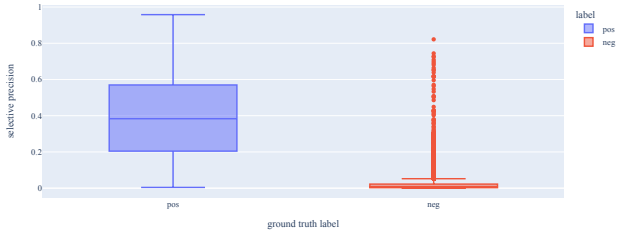


Figure A.4: Predicted values of P(o—T(r)) vs. ground truth label, evaluated over the entire dataset.

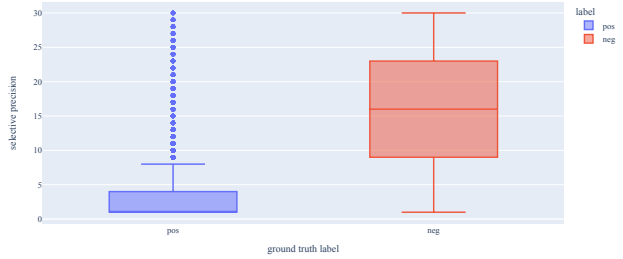


Figure A.5: Ranks assigned to P(o—t(r)_i) vs. ground truth label, evaluated over the entire dataset.

A.4 AUC Scores by Relation

A.4.1 Parametric Knowledge

					aucrel.	diff.	0 vs.	all p
model		GPT2		Mistral7B	GPT2	Mistral7B		
#p	0	20	0	20				

dataset	relation						
Hypernymy	all_relations	0.103	0.140	0.115	0.148	0.354	0.279
PopQA	O is S's father.	0.044	0.046	0.053	0.071	0.031	0.321
	O is the author of S.	0.029	0.034	0.092	0.068	0.142	-0.265
	O, screenwriter of S	0.036	0.036	0.059	0.055	0.002	-0.062
	S located in O.	0.073	0.073	0.096	0.080	0.002	-0.160
	S, capital of O.	0.091	0.080	0.101	0.091	-0.125	-0.095
	S, mother of O.	0.048	0.044	0.075	0.056	-0.078	-0.261
	S play O.	0.028	0.031	0.027	0.046	0.108	0.682
	S was born in O.	0.044	0.046	0.071	0.058	0.038	-0.184
	S's occupation, O.	0.038	0.039	0.028	0.044	0.040	0.587
	Band S play O.	0.034	0.036	0.044	0.042	0.064	-0.050
	The color of S, O.	0.148	0.151	0.197	0.154	0.020	-0.218
	Composer of S, O.	0.027	0.032	0.055	0.061	0.149	0.097
	Director of S, O.	0.026	0.018	0.044	0.043	-0.306	-0.014
	Producer of S, O.	0.021	0.025	0.043	0.050	0.233	0.177
	Religion of S, O.	0.053	0.049	0.054	0.056	-0.065	0.031
	all_relations	0.044	0.045	0.059	0.061	0.026	0.038
TRex	S died in O.	0.070	0.061	0.081	0.076	-0.124	-0.056
	S, subclass of O.	0.141	0.139	0.144	0.131	-0.015	-0.087
	S's religion, O.	0.102	0.204	0.128	0.238	0.994	0.861
	S is located in O.	0.084	0.087	0.099	0.100	0.039	0.009
	S named after O.	0.135	0.111	0.155	0.114	-0.179	-0.266
	S is owned by O.	0.095	0.093	0.162	0.115	-0.021	-0.291
	S is part of O.	0.104	0.105	0.123	0.122	0.005	-0.006
	S is produced by O.	0.113	0.109	0.210	0.144	-0.033	-0.314
	S repr. by label O.	0.144	0.117	0.208	0.183	-0.182	-0.121
	S, capital of O.	0.152	0.112	0.204	0.173	-0.264	-0.153
	S play O music.	0.114	0.113	0.133	0.127	-0.005	-0.045
	S plays in O position.	0.115	0.149	0.066	0.168	0.297	1.546
	S was born in O.	0.064	0.056	0.085	0.077	-0.128	-0.093
	S was created in O.	0.065	0.068	0.079	0.079	0.036	-0.005
	S was founded in O.	0.052	0.053	0.082	0.081	0.021	-0.009

S was aired on O.	0.124	0.106	0.154	0.143	-0.144	-0.070
S was written in O.	0.090	0.092	0.113	0.104	0.024	-0.081
Capital of S, O.	0.089	0.099	0.170	0.130	0.107	-0.234
Headquarter of S, in O.	0.090	0.088	0.129	0.111	-0.017	-0.143
Native lang. of S, O.	0.120	0.113	0.202	0.146	-0.057	-0.277
Official lang. of S, O.	0.100	0.087	0.115	0.101	-0.124	-0.116
Original lang. of S, O.	0.073	0.073	0.101	0.088	-0.006	-0.133
all relations	0.091	0.094	0.113	0.108	0.024	-0.044

Table A.6: AUC values for selective prediction using GPT-2-L and Mistral-7B-Instruct-v0.2 on all datasets and relations. The last two columns denote the relative difference of AUC between runs with zero paraphrases and 20 paraphrases.

A.4.2 In-context Understanding

						auc rel. diff. 0 vs. 3-shot	
	model	GPT-2-L		Mistral-7B-I		GPT-2-L	Mistral-7B-I
	run	0-shot	3-shot	0-shot	3-shot		
dataset	relation	#p					
PopQA	O is S's father.	0	0.044	0.028	0.053	0.070	-0.373	0.310
		20	0.046	0.037	0.071	0.071	-0.184	0.011
	O is the author of S.	0	0.029	0.020	0.092	0.111	-0.312	0.202
		20	0.034	0.029	0.068	0.081	-0.124	0.191
	O, screenwriter S	0	0.036	0.023	0.059	0.070	-0.374	0.186
		20	0.036	0.027	0.055	0.060	-0.257	0.091
	S is located in O.	0	0.073	0.050	0.096	0.101	-0.318	0.051
		20	0.073	0.073	0.080	0.086	-0.001	0.069
	S is the capital of O.	0	0.091	0.052	0.101	0.130	-0.428	0.290
		20	0.080	0.080	0.091	0.111	0.011	0.224
	S is the mother of O.	0	0.048	0.017	0.075	0.092	-0.639	0.224
		20	0.044	0.033	0.056	0.071	-0.259	0.267
	S play O.	0	0.028	0.028	0.027	0.079	-0.015	1.875
		20	0.031	0.069	0.046	0.077	1.217	0.681
	S was born in O.	0	0.044	0.033	0.071	0.058	-0.255	-0.183

		20	0.046	0.040	0.058	0.056	-0.118	-0.042
	S's occupation is O.	0	0.038	0.033	0.028	0.066	-0.122	1.378
		20	0.039	0.051	0.044	0.063	0.287	0.443
	Band S plays O.	0	0.034	0.021	0.044	0.053	-0.385	0.190
		20	0.036	0.038	0.042	0.054	0.046	0.285
	The color of S is O.	0	0.148	0.136	0.197	0.270	-0.083	0.375
		20	0.151	0.151	0.154	0.168	-0.001	0.091
	The composer of S is O.	0	0.027	0.023	0.055	0.064	-0.150	0.153
		20	0.032	0.030	0.061	0.067	-0.038	0.102
	The director of S is O.	0	0.026	0.019	0.044	0.044	-0.279	0.003
		20	0.018	0.017	0.043	0.042	-0.088	-0.028
	The producer of S is O.	0	0.021	0.013	0.043	0.037	-0.382	-0.125
		20	0.025	0.021	0.050	0.052	-0.165	0.028
	The religion of S is O.	0	0.053	0.037	0.054	0.083	-0.292	0.550
		20	0.049	0.064	0.056	0.070	0.302	0.255
TRex	S is located in O.	0	0.084	0.054	0.099	0.124	-0.359	0.251
		20	0.087	0.081	0.100	0.120	-0.063	0.201
	S is the capital of O.	0	0.152	0.085	0.204	0.222	-0.438	0.089
		20	0.112	0.133	0.173	0.196	0.189	0.135
	S was born in O.	0	0.064	0.040	0.085	0.083	-0.382	-0.026
		20	0.056	0.060	0.077	0.083	0.067	0.079
	S died in O.	0	0.070	0.039	0.081	0.089	-0.438	0.104
		20	0.061	0.064	0.076	0.089	0.049	0.159
	S is a subclass of O.	0	0.141	0.085	0.144	0.257	-0.402	0.785
		20	0.139	0.131	0.131	0.241	-0.057	0.834
	S, O religion.	0	0.102	0.129	0.128	0.267	0.259	1.090
		20	0.204	0.222	0.238	0.265	0.091	0.112
	S is named after O.	0	0.135	0.076	0.155	0.254	-0.442	0.634
		20	0.111	0.123	0.114	0.214	0.109	0.879
	S is owned by O.	0	0.095	0.055	0.162	0.163	-0.418	0.009
		20	0.093	0.083	0.115	0.129	-0.107	0.129
	S is part of O.	0	0.104	0.069	0.123	0.235	-0.334	0.918
		20	0.105	0.102	0.122	0.234	-0.022	0.920

S is produced by O.	0	0.113	0.072	0.210	0.355	-0.363	0.687
	20	0.109	0.112	0.144	0.205	0.026	0.419
S music label O.	0	0.144	0.096	0.208	0.221	-0.333	0.066
	20	0.117	0.146	0.183	0.217	0.246	0.190
S plays O music.	0	0.114	0.079	0.133	0.150	-0.306	0.127
	20	0.113	0.118	0.127	0.140	0.046	0.109
S plays in O position.	0	0.115	0.073	0.066	0.132	-0.363	0.998
	20	0.149	0.107	0.168	0.154	-0.283	-0.085
S was created in O.	0	0.065	0.038	0.079	0.083	-0.420	0.045
	20	0.068	0.059	0.079	0.083	-0.127	0.049
S was founded in O.	0	0.052	0.037	0.082	0.087	-0.289	0.063
	20	0.053	0.055	0.081	0.088	0.041	0.085
S was aired on O.	0	0.124	0.082	0.154	0.188	-0.336	0.224
	20	0.106	0.113	0.143	0.161	0.071	0.129
S was written in O.	0	0.090	0.053	0.113	0.152	-0.412	0.350
	20	0.092	0.091	0.104	0.122	-0.008	0.178
The capital of S is O.	0	0.089	0.066	0.170	0.184	-0.262	0.087
	20	0.099	0.103	0.130	0.144	0.038	0.109
HQ of S is in O.	0	0.090	0.048	0.129	0.135	-0.470	0.042
	20	0.088	0.081	0.111	0.116	-0.084	0.045
language of S is O.	0	0.120	0.087	0.202	0.194	-0.276	-0.040
	20	0.113	0.114	0.146	0.127	0.007	-0.128
official lang. of S, O.	0	0.100	0.060	0.115	0.111	-0.395	-0.029
	20	0.087	0.088	0.101	0.101	0.010	-0.000
original language S, O.	0	0.073	0.037	0.101	0.088	-0.488	-0.133
	20	0.073	0.063	0.088	0.085	-0.140	-0.036

Table A.7: AUC values for selective prediction using GPT-2-L and Mistral-7B-Instruct-v0.2 on all datasets and relations. The last two columns denote the relative difference of AUC between runs with 0-shot prompts and 3-shot prompts. #p denotes the number of paraphrases.

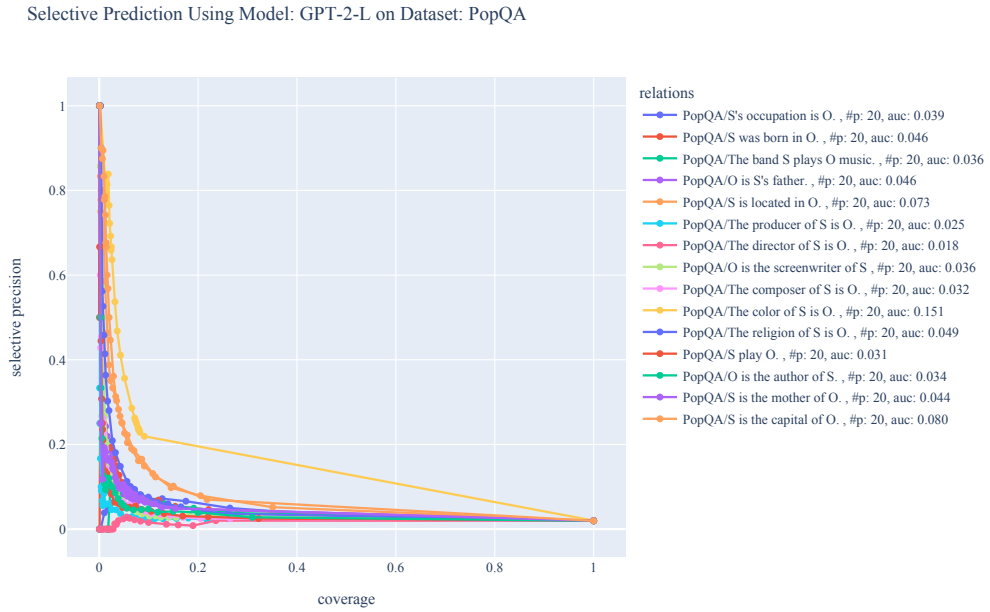


Figure A.6: Classification performance trade-off for datasets across all relations.

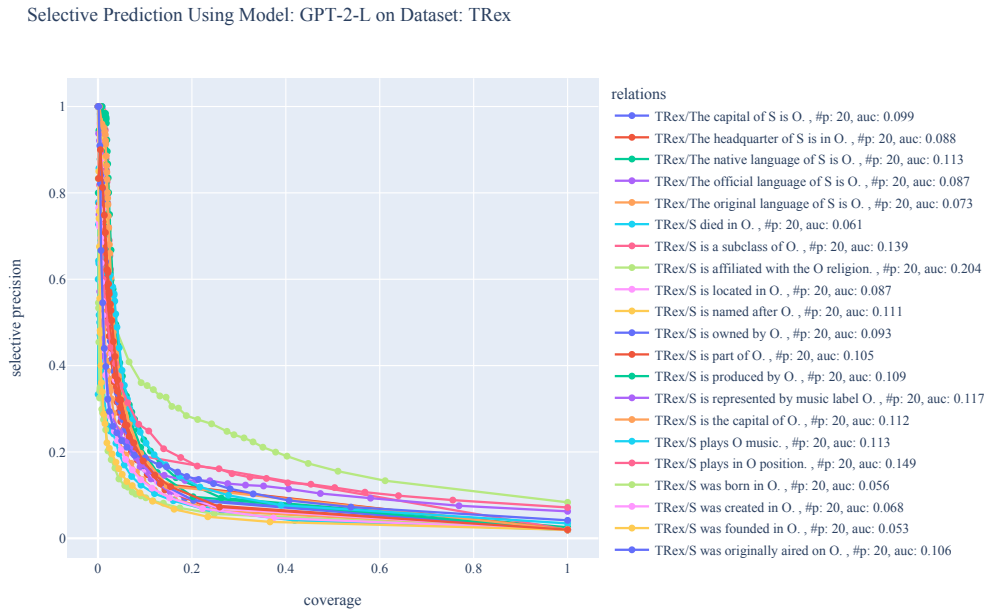


Figure A.7: Classification performance trade-off for datasets across all relations.

A.5 Full Example for $P(o(r)|r, s(r))$ and Aggregation over Paraphrase Templates

Figure A.8: Full illustration showing how the probabilities of different paraphrases are aggregated in the PopQA dataset. The remaining rows show how the probabilities per object and individual paraphrase template are distributed for one relation, one subject and 50 objects.