Selective-LAMA: Selective Prediction for Confidence-Aware Evaluation of Language Models

Hiyori Yoshikawa[†‡] **Naoaki Okazaki**[†]
[†]Tokyo Institute of Technology, Japan [‡]Fujitsu Limited, Japan
{hiyori.yoshikawa@nlp., okazaki@}c.titech.ac.jp

Abstract

Recent studies have suggested that neural language models learn and store a large amount of facts and commonsense knowledge from training data. The ability of language models to restore such knowledge is often evaluated via zero-shot cloze-style QA tasks. However, such evaluations rely only on prediction accuracy without punishing the systems for their mistakes, e.g., simply guessing or hallucinating likely answers. Selective prediction is a more informative evaluation framework that takes the confidence of predictions into account. Under the selective prediction setting, a model is evaluated not only by the number of correct predictions, but also by the ability to filter out dubious predictions by estimating the confidence of individual predictions. Such confidence-aware evaluation is crucial for determining whether to trust zero-shot predictions of language models. In this paper, we apply the selective prediction setting to an existing benchmark, LAMA probe, and conduct extensive experiments with recent neural language models and different confidence functions. We empirically show that our Selective-LAMA evaluation is more robust to the effect of simple guesses than the conventional accuracy-based evaluation. Our evaluation reveals the importance of the choice of confidence functions by showing that simply relying on token probabilities is not always the best choice. Further analysis shows that various confidence functions exhibit different preferences over predicted tokens for a given context.

1 Introduction

Recently, knowledge stored in pre-trained language models has been intensively investigated. Many studies have suggested that language models trained on a large amount of textual corpora, such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2019; Brown et al., 2020), store both linguistic knowledge (Warstadt et al., 2019; Miaschi et al., 2020) and factual and commonsense knowledge (Bosselut et al., 2019; Roberts et al., 2020) during training. However, this knowledge is embedded in the parameters of these language models and thus is difficult to interpret, in contrast to symbolic knowledge bases, which allows us to inspect and edit stored facts explicitly.

Petroni et al. (2019) proposed a benchmark task, the LAMA probe, that aims at evaluating the amount of relational knowledge, such as commonsense knowledge and facts, which is stored in a language model. In LAMA probe, a relational fact is converted into a cloze statement (*query*) and then given to a language model as a fill-in-the-blank question. If the language model fills in the blank with the correct answer, the model is considered to possess "knowledge" of the relation. According to Petroni et al.'s experiments, the BERT language model (Devlin et al., 2019) has a comparable performance to a supervised relation extraction baseline, with precision ranging from 10.5 to 32.3 depending on the dataset type.

However, in many applications, we are concerned not only with the amount of the knowledge extracted from a language model, but also with its reliability. This is because large pre-trained language models are known to fluently generate "facts" that they have never seen (Cao et al., 2018; Rohrbach et al., 2018; Müller et al., 2020). Therefore, it is crucial to know when we can trust the output of a language model. The LAMA probe framework does not cover this issue, as it always forces the model to output an answer for all instances, regardless of whether the model really "knows" the answer to a query. This means that it implicitly trusts all outputs of a language model to the same degree.

Figure 1 shows an example suggesting that a pre-trained language model is not always using its knowledge for prediction. The figure shows the distribution of predicted tokens for a particular relation in the original LAMA probe benchmark

Findings of the Association for Computational Linguistics: EACL 2023, pages 2017–2028

(`place-of-birth`). We can see that three tokens account for more than half of the wrong predictions. This indicates that the model has a bias which it acquired during training, probably due to the input template used, rather than using actual question-specific knowledge about individual facts.

To address this issue, we apply the selective prediction (El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017) setting to the LAMA probe and propose a new evaluation framework, *Selective-LAMA*, to evaluate both the amount of knowledge in a pre-trained language model and the model's ability to estimate the reliability of its prediction. Selective prediction is a framework by which a system can choose whether to output the individual predictions of a model based on the prediction results. Specifically, we consider the selection with guaranteed risk control setting (Geifman and El-Yaniv, 2017), where the system computes confidence scores of individual predictions to determine whether it outputs the predictions. A system is evaluated by the number of predictions it can make while maintaining a risk of error below a certain level. To achieve high performance, a system is required not only to answer many questions correctly, but also to accurately estimate the model's confidence about individual facts and determine when the system should not answer a question.

In this paper, we focus on masked language models and address the following research questions: (1) whether the pre-trained language model has the ability to estimate the confidence of individual predictions and (2) how various confidence metrics affect the ability of a system to do that. With our proposed Selective-LAMA framework, we examine several basic confidence functions that can be computed using only language model predictions and do not require additional datasets or external knowledge sources. We empirically verify that the selective prediction evaluation is less likely to overestimate predictions with template-related biases than the conventional accuracy-based evaluation. The results of the experiments suggest that the choice of confidence functions also influences the results, showing that simply using token probability is a strong baseline but not always the best choice, and that the optimal confidence function depends on both the model and the dataset. We hope that the selective prediction framework facilitates an under-explored research direction of utilizing predictions of language models in a more reliable way.

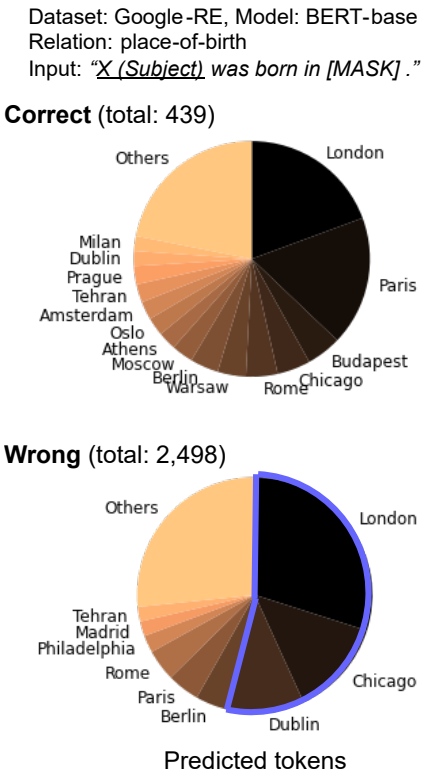


Figure 1: Composition of predicted tokens in each of the correct (top) and wrong (bottom) predictions by BERT-base for the `place-of-birth` relation in the Google-RE dataset (size: 2,937). Just three tokens account for more than half of the wrong predictions, implying that the model has a template-dependent bias.

2 Selective Prediction

Under the selective prediction setting (El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017), a *selective classifier* determines whether a system should output the prediction of the model. We consider a classification problem from an input space $\mathcal{X}$ to a set of labels $\mathcal{Y}$. A selective classifier (f, g) consists of an original classification model $f : \mathcal{X} \to \mathcal{Y}$ and a *selection function* $g : \mathcal{X} \to \{0, 1\}$. Given an input example $x \in \mathcal{X}$, a selection function determines whether the system outputs the prediction $f(x) \in \mathcal{Y}$:

$$(f, g)(x) := \begin{cases} f(x) & \text{if } g(x) = 1 \\ \textit{don't know} & \text{if } g(x) = 0 \end{cases}. \quad (1)$$

Geifman and El-Yaniv (2017) introduced the selection with guaranteed risk (SGR) setting, which uses a confidence-based selection function:

$$g(x) = \begin{cases} 1 & \text{if } \phi(x) \geq \beta \\ 0 & \text{if } \phi(x) < \beta \end{cases}, \quad (2)$$

where $\phi(x) : \mathcal{X} \to \mathbb{R}$ is the *confidence score function* of f. The system outputs the prediction if the confidence score exceeds the threshold $\beta \in \mathbb{R}$. This setting allows a user to adjust the error risk generated by the system by appropriately setting the value of β. Specifically, increasing β decreases the number of cases predicted by the system while reducing the risk of making a wrong prediction.

Under the SGR setting, there is a risk-coverage trade-off between the risk $(N_{\text{pred}} - N_{\text{corr}})/N_{\text{pred}}$ that a selective classifier will make a wrong prediction and the coverage N_{pred}/N of the predictions made by the system. Here, N, N_{pred}, and N_{corr} denote the number of all examples, predicted examples, and correct predictions, respectively. The performance of a selective classifier is evaluated based on the AUC of the risk-coverage curve (RC-AUC) obtained by changing β in the selection function (2). A smaller RC-AUC value indicates a lower risk of making a wrong prediction. In practice, the threshold will be determined by the level of risk acceptable to the users.

3 Selective-LAMA

3.1 LAMA Probe and Model Prediction

In the original LAMA probe, a relational fact is converted into a natural sentence using templates and input into the language model. For example, when querying about an entity that has a relationship of `born-in` with "*Dante*," the input to the language model will be "*Dante was born in* `[MASK]`.," where `[MASK]` is a special token that represents the mask token. The model output for the masked position is considered the answer to the query.[1] The templates are manually designed for each relation type.

Following the original study (Petroni et al., 2019), we focus on bi-directional language models. Given the input sentence with a mask token at the t-th position $x = W_{\setminus t} := (w_1, \ldots, w_{t-1}, \texttt{[MASK]}, w_{t+1}, \ldots, w_{|W|})$, the language model predicts the probability distribution of the t-th token $P_{\text{LM}}(w_t|W_{\setminus t})$. The model prediction is the token w' with the highest probability:

$$f(x) = w' := \arg\max_{w_t} P_{\text{LM}}(w_t|W_{\setminus t}). \quad (3)$$

We denote the sentence in which the masked position is filled with w' by W'.

[1] For simplicity, the target is limited to entities comprising a single word.

3.2 Confidence Functions

As the task is to evaluate the knowledge present in a pre-trained language model, we select confidence functions that use only the prediction of the language model and do not require additional training or external knowledge sources. The following is the list of confidence functions that we investigate.

`Token` (T) The simplest confidence function is to use the log probability of the predicted token w' (3) directly:

$$\phi_{\text{T}}(x) = \log P_{\text{LM}}(w'|W_{\setminus t}). \quad (4)$$

`Sent` (S) `Sentence`-level likelihood is widely used in the context of sentence acceptability and fact-checking (Lau et al., 2020; Lee et al., 2021). This reflects how natural the entire sentence is when the predicted token is substituted into the mask position. Here, we adopt the pseudo-log likelihood (Salazar et al., 2020) for masked language models normalized by sentence length:

$$\phi_{\text{S}}(x) = \frac{1}{|W'|} \sum_{u=1}^{|W'|} \log P_{\text{LM}}(w_u|W'_{\setminus u}). \quad (5)$$

`Gap` (G) Let w'' be the token with the second-largest probability by the model. The confidence score is then calculated as follows:

$$\phi_{\text{G}}(x) = \log P_{\text{LM}}(w'|W_{\setminus t}) - \log P_{\text{LM}}(w''|W_{\setminus t}). \quad (6)$$

This function is based on the assumption that a model makes a confident prediction when the probability of the predicted token is significantly larger than that of other tokens.

`Reranking` (R) The following function is based on the assumption that, if the confidence of the prediction is high, the score for the prediction is consistently higher than those of other candidates even when different metrics are used. First, we obtain top-K predictions $\mathcal{W}$ based on the token log probability (3). Then, we re-rank those candidates using another score function ψ. Let $\text{rank}_\psi(w')$ be the rank of w' after the reranking. The confidence score is subsequently computed as follows:

$$\phi_{\text{R}}(x) = \log_2 \frac{K}{\text{rank}_\psi(w')} = \log_2 K - \log_2 \text{rank}_\psi(w'). \quad (7)$$

The above score function is essentially a measure based only on the new rank after the reranking and has been used to assess the risk of language models to memorize privacy information (Carlini et al., 2019). In the experiments, we apply $K = 100$ and use the `Sent` score $\phi_S(x)$ for ψ.

`DropoutMean` (DM) Dropout-based metrics have been widely used to estimate uncertainty of deep neural network models (Gal and Ghahramani, 2016). The basic concept is to use dropout to sample slightly different model parameters that yield different predictions and to use stochastic information to estimate the model uncertainty. Following (Kamath et al., 2020), we adopt two dropout-based measures. We apply M different dropout masks to the language model's layers and obtain different probability distributions. Let $P_{\text{LM}}^{(m)}(w'|W_{\setminus t})$ denote the m-th output ($m \in \{1, \ldots, M\}$). `DropoutMean` takes the mean of the M outputs:

$$\phi_{\text{DM}}(x) = \frac{1}{M} \sum_{m=1}^{M} P_{\text{LM}}^{(m)}(w'|W_{\setminus t}), \quad (8)$$

which can be considered an ensemble of the M model predictions.

`DropoutVar` (DV) Similarly, `DropoutVar` utilizes the variance of the outputs. As large variance implies high model uncertainty, we take the negative variance of the outputs:

$$\phi_{\text{DV}}(x) = -\frac{1}{M} \sum_{m=1}^{M} (P_{\text{LM}}^{(m)}(w'|W_{\setminus t}) - \phi_{\text{DM}}(x))^2. \quad (9)$$

In our experiments, we apply $M = 30$ different dropout masks for each input, using the same dropout ratios as those used to train the models.

`TemplateDiff` (TD) A large portion of the LAMA probe benchmark consists of instances based on subject-relation-object triples. These instances share relation-specific templates, such as "`<subj>` *was born in* `[MASK]`.", where the subject of each triple is substituted for `<subj>`. Cao et al. (2021) found that predictions of language models are highly biased by templates and the impact of subject entities are limited. Inspired by this observation, we define a confidence measure that assesses the impact of subject entities to predictions. Let W_{temp} be a template-only input sentence where the subject of the input $W_{\setminus t}$ is replaced by the mask token, e.g. "`[MASK]` *was born in* `[MASK]`." Then, we calculate the confidence by comparing the log probabilities of the prediction with and without the subject entity mention:

$$\phi_{\text{TD}}(x) = P_{\text{LM}}(w'|W_{\setminus t}) - P_{\text{LM}}(w'|W_{\text{temp}}). \quad (10)$$

4 Experiments

The proposed Selective-LAMA framework allows us to evaluate the ability of language models to recognize questions for which they *do not know* the answer. To see how the proposed framework affects the evaluation of language models, in Section 4.2, we first compare the evaluation based on the Selective-LAMA framework with the conventional accuracy-based evaluation, focusing on the sensitivity to biased predictions. Then, in Section 4.3, we present a comprehensive study of the performance of three masked language models on different datasets using the confidence functions introduced in Section 3.2.

4.1 Experimental Settings

We used the same data set as the original LAMA benchmark for our experiment and evaluated it with our proposed Selective-LAMA framework. The benchmark consists of four datasets: Google-RE, T-REx, ConceptNet, and SQuAD. The Google-RE and T-REx datasets contain relational facts extracted from Wikipedia. The ConceptNet dataset contains relational knowledge about commonsense extracted from the ConceptNet dataset (Speer and Havasi, 2012). The SQuAD dataset (Rajpurkar et al., 2016) is based on a question answering dataset of the same name, but the questions are rewritten in cloze style. As all these datasets, except for ConceptNet, use Wikipedia as the knowledge source, evidence for the correct answer should be found in Wikipedia. For language models, we use BERT-base (110 M parameters), BERT-large (340 M parameters), and RoBERTa-base (Liu et al., 2019). Because these models are trained using Wikipedia, it is expected that the models have seen the correct answers for the queries during training.

4.2 Template Bias Robustness

In the selective prediction framework, the performance of language models is evaluated by RC-AUC (Section 2), while the original LAMA benchmark uses the accuracy of the top-1 predictions as the evaluation metric. A disadvantage of accuracy-based evaluation is that the amount of knowledge of a language model can be overestimated by counting lucky guesses. Such lucky guesses can affect the evaluation results, especially in cases where the model's predictions are biased by relation-specific templates (Figure 1).

We investigate how these evaluation metrics are

		BERT-base		BERT-large		RoBERTa-base	
		$\mathrm{Cov}^{\mathrm{A}}$	$\mathrm{Cov}^{\mathrm{P}}$	$\mathrm{Cov}^{\mathrm{A}}$	$\mathrm{Cov}^{\mathrm{P}}$	$\mathrm{Cov}^{\mathrm{A}}$	$\mathrm{Cov}^{\mathrm{P}}$
Accuracy		0.387	-0.247	0.469	-0.244	0.512	-0.224
RC-AUC	`Token`	0.344 ↓	-0.316 ↓	0.438 ↓	-0.292 ↓	0.484 ↓	-0.277 ↓
(negative)	`Sent`	0.355 ↓	-0.290 ↓	0.441 ↓	-0.285 ↓	0.499 ↓	-0.249 ↓
	`Gap`	0.351 ↓	-0.314 ↓	0.430 ↓	-0.294 ↓	0.474 ↓	-0.285 ↓
	`Reranking`	0.350 ↓	-0.286 ↓	0.452 ↓	-0.283 ↓	0.498 ↓	-0.266 ↓
	`DropoutMean`	0.338 ↓	-0.319 ↓	0.433 ↓	-0.293 ↓	0.486 ↓	-0.280 ↓
	`DropoutVar`	0.419 ↑	-0.125 ↑	0.470 ↑	-0.124 ↑	0.456 ↓	-0.166 ↑
	`TemplateDiff`	0.349 ↓	-0.317 ↓	0.427 ↓	-0.299 ↓	0.486 ↓	-0.271 ↓

Table 1: Correlation between evaluation metrics and template bias metrics: answer coverage ($\mathrm{Cov}^{\mathrm{A}}$) and prediction coverage ($\mathrm{Cov}^{\mathrm{P}}$) on the T-REx dataset. Here, we use the sign-reversed RC-AUC values for easier interpretation.

affected by template-related biases using the T-REx subset of the LAMA benchmark, which contains 34k facts about 41 different relations with their corresponding templates. To quantify template-related biases, we introduce two indicators: *prediction coverage* and *answer coverage*.

Prediction coverage quantifies biases in model predictions for a given template. If a model often predicts the same answers for a template, it is likely that the predictions are heavily influenced by the template, rather than using knowledge of individual subject entities. Let $\mathcal{D}_r = (\{(s_i, o_i)\}_{i=1}^{N_r}, t_r)$ denote a relation subset containing N_r fact triples (s_i, r, o_i) of relation r and a template t_r. We represent the input sentence corresponding to the i-th fact by $t_r(s_i)$. For each relation subset D_r, we first identify five most frequent tokens $\mathcal{W}^{\mathrm{freq}}(r)$ predicted by a model. Prediction coverage is the proportion of predicted tokens covered by these tokens:

$$\mathrm{Cov}^{\mathrm{P}}(r) = \frac{|\{i \mid f(t_r(s_i)) \in \mathcal{W}^{\mathrm{freq}}(r)\}|}{N_r}. \quad (11)$$

Answer coverage quantifies biases in a relation subset in the test set. If the distribution of the correct answers for a relation subset is skewed towards a few particular entities, the subset can be easily answered by exploiting the bias without using the knowledge of individual subject entities. Answer coverage is calculated as the proportion of gold answers covered by the frequently predicted tokens:

$$\mathrm{Cov}^{\mathrm{A}}(r) = \frac{|\{i \mid o_i \in \mathcal{W}^{\mathrm{freq}}(r)\}|}{N_r}. \quad (12)$$

Table 1 shows the correlation between the bias indicators and the evaluation metrics including accuracy and (negative) RC-AUC calculated with different confidence functions. Compared to the conventional accuracy metric, all RC-AUC metrics except `DropoutVar` show a weaker positive correlation with answer coverage and a stronger negative correlation with prediction coverage, indicating that the RC-AUC metrics are less likely to overestimate template-biased predictions and results from intrinsically biased test examples.

Figure 2 shows the output of the BERT-base model for two relation subsets `P36` and `P1412`. Although the accuracy scores for both subsets are around 0.6, for `P1412`, both the prediction and answer distributions are biased towards a small number of entities, leading to high prediction and answer coverage. The `Token` confidence scoring fails to discriminate between correct and incorrect predictions in this subset, resulting in high risk at a low coverage point. Evaluation based on the RC-AUC score successfully captures the difference between these two cases and avoids overestimating the results from biased predictions.

4.3 Selective-LAMA Evaluation and Analysis

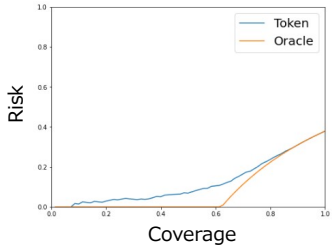
Overall performance on different datasets

Table 2 shows the RC-AUC scores achieved by different confidence functions on various datasets. We also calculate the performance lower bound based on an oracle confidence function that gives 1 to all correct predictions and 0 to incorrect ones. While the simple `Token` metric constantly performs well, the best confidence function depends on the model and dataset. Notably, `Gap` and `TemplateDiff` perform better on the datasets of Wikipedia fact triples, Google-RE and T-REx, than on ConceptNet and SQuAD, outperforming the `Token` metric in some cases. The breakdown of the results on the T-REx dataset indicates that the performance of confidence functions also depend on relation templates. We further investigate this phenomenon below.

r = **P36 ("The capital of X (Subject) is [MASK] .")** **Accuracy = 0.621, RC-AUC = 0.121**

$\mathcal{W}^{\text{freq}}(r)$: Rome (1.9%), Baghdad (1.7%), Paris (1.7%), Bangor (1.7%), Kabul (1.4%)

Prediction coverage: 0.084, Answer coverage: 0.047

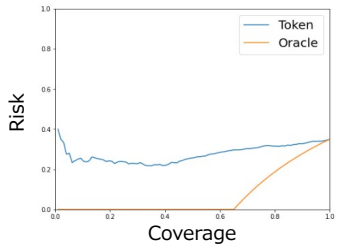


Subject	Gold	Predict	ϕ_T
Sri Lanka	Colombo	Colombo	-0.001
Bratislava Region	Bratislava	Bratislava	-0.001
Albania	Tirana	Tirana	-0.002
Tirana District	Tirana	Tirana	-0.002
Hiroshima Prefecture	Hiroshima	Hiroshima	-0.002
Brest Region	Brest	Brest	-0.003
South Korea	Seoul	Seoul	-0.003
Afghanistan	Kabul	Kabul	-0.003
Bosnia and Herzegovina	Sarajevo	Sarajevo	-0.003
Democratic Republic of Afghanistan	Kabul	Kabul	-0.003

r = **P1412 ("X (Subject) used to communicate in [MASK] .")** **Accuracy = 0.650, RC-AUC = 0.278**

$\mathcal{W}^{\text{freq}}(r)$: English (38.6%), French (15.9%), Spanish (10.0%), Italian (9.4%), Russian (4.5%)

Prediction coverage: 0.784, Answer coverage: 0.687



Subject	Gold	Predict	ϕ_T
Adrianus Valerius	Dutch	Latin	-0.490
Muhammad Ali	English	Arabic	-0.575
Gloria Estefan	Spanish	Spanish	-0.587
Imre Nagy	Hungarian	Hungarian	-0.610
Sextus Pompeius Festus	Latin	Latin	-0.619
Hieronymus Fabricius	Latin	Latin	-0.635
Infante Juan, Count of Barcelona	Spanish	Spanish	-0.637
Ramon Llull	Catalan	Spanish	-0.665
Lau Kar-leung	Chinese	Cantonese	-0.724
Juan Bautista Villalpando	Spanish	Spanish	-0.749

Figure 2: BERT-base results for relation subsets $r = \texttt{P36}$ and $r = \texttt{P1412}$. While the model performance is similar in terms of accuracy, the RC-AUC scores exhibit a large difference. Left: Risk-coverage curves of Token and the Oracle confidence scores. Right: Top 20 predictions sorted by the Token confidence score ϕ_{T}. The gray-shaded rows indicate incorrect predictions. Many incorrect predictions for P1412 indicate that the model suffers from high risk even at a low coverage point.

When does a confidence function beat another?

For the T-REx dataset in Table 2, Gap and TemplateDiff outperform the Token metric for BERT-base and RoBERTa-base, respectively. We choose these two cases and perform a pairwise comparison for each relation type to identify the properties that determine the preference for one confidence function over the other. The results in Table 3 show that Gap is preferred over Token for easier relations with high accuracy and low RC-AUC for BERT-base, whereas TemplateDiff is preferred over Token for more difficult relations for RoBERTa-base. The subset where Gap is preferred over Token also shows lower prediction coverage, which might be because the Gap function is not good at handling overconfident predictions by definition.

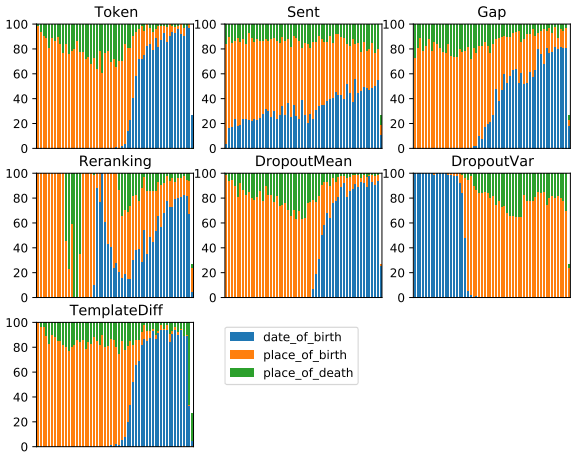


Figure 3: Breakdown of relation types of BERT-base predictions on Google-RE, sorted by confidence scores (left is the largest).

Confidence functions and relation templates

To understand whether and how different confidence functions prioritize one relation over another, we visualize in Figure 3 the composition of the relation types of input examples sorted by the confidence scores in the Google-RE dataset predicted by the BERT-base model. The Google-RE dataset contains three relation types, date-of-birth, place-of-birth, and place-of-death. Evidently, the BERT-base language model tend to pro-

Model	Conf.	Google-RE	T-REx				ConceptNet	SQuAD	All
			1-1	N-1	N-M	All			
BERT-base	Token	**.775**	**.118**	.434	.611	.478	**.686**	**.755**	.545
	Sent	.834	.163	.549	.776	.594	.797	.815	.652
	Gap	.798	.133	**.422**	.604	**.470**	.714	.794	.548
	Reranking	.835	.248	.580	.623	.597	.834	.798	.633
	DropoutMean	**.775**	.123	.425	.609	.473	.690	.762	**.543**
	DropoutVar	.962	.525	.834	.883	.850	.918	.912	.886
	TemplateDiff	.778	.119	.427	**.603**	.472	.782	-	-
	Oracle	.663	.070	.301	.456	.344	.551	.583	.413
BERT-large	Token	.763	.085	.409	.575	.445	**.616**	**.669**	.506
	Sent	.815	.119	.520	.740	.560	.738	.768	.614
	Gap	.801	.092	.412	.597	.456	.650	.712	.525
	Reranking	.826	.170	.552	.610	.576	.792	.785	.609
	DropoutMean	**.762**	.086	**.402**	**.572**	**.441**	**.616**	.670	**.504**
	DropoutVar	.960	.370	.775	.894	.817	.881	.907	.858
	TemplateDiff	.763	**.084**	.406	.574	.444	.730	-	-
	Oracle	.648	.048	.277	.459	.327	.489	.522	.388
RoBERTa-base	Token	.818	.191	.540	.635	.562	.618	**.741**	**.599**
	Sent	.876	.267	.631	.761	.657	.754	.780	.716
	Gap	.827	.197	.545	.632	.565	.657	.782	.610
	Reranking	.865	.276	.637	.627	.636	.804	.828	.669
	DropoutMean	.815	.201	**.536**	.633	.562	**.615**	.744	**.599**
	DropoutVar	.979	.643	.924	.920	.920	.896	.907	.923
	TemplateDiff	**.813**	**.189**	.537	**.626**	**.558**	.744	-	-
	Oracle	.730	.106	.416	.492	.432	.503	.571	.474

Table 2: RC-AUC calculated on each dataset (lower is better). For T-REx, the results on three splits divided by the property of the relations are also provided: one-to-one relations (1-1), many-to-one relations (N-1) and many-to-many relations (N-M). "Oracle" represents the best possible performance that could be achieved by an oracle confidence function that gives 1 to all correct predictions and 0 to incorrect ones. TemplateDiff cannot be calculated for SQuAD as the instances do not contain subject entities.

duce high probability outputs for a certain relation type, namely, place-of-birth. While Gap, DropoutMean, and TemplateDiff follow the same trend as that of Token, Sent and Reranking are less sensitive to relation types. DropoutVar shows the opposite trend. While the Token metric is effective in many cases, one should be aware of the potential bias this confidence function may introduce.

Table 4 compares the most frequent predictions of BERT-base on the Google-RE dataset ranked top by two different metrics: Token and Reranking. We can observe similar distributions for the date-of-birth relation type. This indicates that the model is strongly biased toward a limited vocabulary for this particular relation type. For the other two relation types, the frequent words in the top predictions are clearly different between Token and Reranking. However, while the overlap of the top-ranked predictions between them are small, both results have strong preference toward a few particular tokens for each relation type. For place-of-birth, five tokens account for more than 50% of the top-ranked predictions for both Token and Reranking. In place-of-death, just one token occupies around 40% of the top predictions. The results indicate that these confidence functions produce different template biases rather than that one is more robust to template biases than the other.

Using confidence functions for prediction

In the experiments above, model predictions are always determined by the token log probability as in (3). However, some of the confidence functions introduced in Section 3.2 can also be used directly to determine the prediction as an alternative to (3). Therefore, we investigate whether effective confidence functions are also effective in improving prediction accuracy (P @ 1) when used directly for token prediction. For Gap, we extend the original definition (6) so that we can apply the function to token candidates that are not ranked first in terms of token probability. Let $w^{(k)}$ denote the k-th best prediction based on the model's predicted token probability. Then, the extended Gap function is

	BERT-base			
	All	Token-win	Gap-win	Δ
Accuracy	0.311	0.283	0.413	-0.130
RC-AUC Token	0.558	0.577	0.466	0.111
RC-AUC Gap	0.566	0.597	0.443	0.154
Answer Cov.	0.285	0.276	0.334	-0.058
Prediction Cov.	0.547	0.579	0.464	0.115
	RoBERTa-base			
	All	Token-win	TD-win	Δ
Accuracy	0.242	0.315	0.231	0.085
RC-AUC Token	0.643	0.545	0.657	-0.112
RC-AUC TD	0.638	0.546	0.650	-0.103
Answer Cov.	0.237	0.285	0.235	0.050
Prediction Cov.	0.562	0.586	0.541	0.045

Table 3: Comparison of two confidence functions on the T-REx dataset (Token–Gap for BERT-base and Token–TemplateDiff (TD) for RoBERTa-base). The average value of each metric is displayed for the entire T-REx dataset (All) and the subset for which the confidence function X outperforms the other (X-win). Δ stands for the difference between the two subsets.

defined as follows:

$$\phi_{\mathrm{G}}(x) = \frac{1}{k}(\log P_{\mathrm{LM}}(w^{(k)}|W_{\setminus t}) - \log P_{\mathrm{LM}}(w^{(k+1)}|W_{\setminus t})). \quad (13)$$

The Gap score for the lowest ranked prediction is defined as zero. The computation of the Sent score requires $\mathcal{O}(|W'|\cdot V)$ forward computations for each instance, where V is the vocabulary size. To save computational cost, we approximate the prediction results by limiting the token candidates to the top 100 results based on the Token score (3).

Table 5 shows the results. For all models, the best performance on all data is achieved by DropoutMean. However, all functions, except for DropoutVar, show a quite competitive performance in terms of precision. Unlike for confidence estimation, no advantage is observed for Gap and TemplateDiff on the T-REx dataset. Overall, the performance of these confidence functions is flat when they are used directly for token prediction. Furthermore, there is no strong correlation between the performance of each confidence function as a predictor and a confidence estimator. The results suggest that effective metrics for inference and confidence estimation should be designed based on different strategies.

5 Related Work

In NLP, the reliability of the model responses has been discussed mainly in the field of question answering. Estimating the confidence of an answer is critical in quiz competitions, such as Jeopardy, since the system has to decide when to answer the questions (Ferrucci et al., 2010). Kamath et al. (2020) recently introduced a selective prediction setting to question answering tasks and then evaluated the performance of the models on out-of-domain questions. Jiang et al. (2021) addressed a similar problem, but focused on a calibration of the model prediction on QA tasks. While they focused on extractive or multiple-choice QA tasks where a limited number of candidate answers are available, our focus is on the knowledge probing of language models where the candidate answer is the entire vocabulary and, thus, false positives are more frequent.

Several studies have addressed the reliability issue of pre-trained language models as a calibration problem; the goal of these studies is to train a well-calibrated language model that makes accurate confidence estimation. Desai and Durrett (2020) investigate the calibration level of pre-trained language models, focusing on BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). They evaluate the "out-of-the-box" performance of these models without post-processing, as well as the performance of post-hoc calibration methods (e.g., temperature scaling and label smoothing). Kong et al. (2020) proposed regularization methods to better calibrate pre-trained language models. Both studies assume access to (at least in-domain) training data of the target tasks on which parameterized calibration models can be trained. In contrast, our study primarily aims to explore better signals in pre-trained language models to estimate the knowledge they store. Thus, we focus on methods that do not require additional training data or an external knowledge source. Although training-based methods (e.g., temperature scaling) have the potential to achieve better performance in terms of calibration, optimal parameters vary depending on models and tasks, especially when evaluated in out-of-domain datasets (Desai and Durrett, 2020).

In our experiments, all queries have at least one correct answer. Therefore, when a model cannot answer a question correctly, this implies that it did not acquire the correct knowledge during training or that its knowledge was not elicited by the natural language query because of a sub-optimal prompt (Jiang et al., 2020). However, there are also cases where the question is essentially impossible to answer due to ambiguity (Zhang and Choi, 2021) or false presupposition (Kim et al., 2021).

Relation	Confidence	Top predictions
date-of-birth	Token	1979 (47), 1944 (33), 1988 (10), 1990 (8)
	Reranking	1979 (44), 1944 (32), 1953 (13), 1970 (3), 1949 (2)
place-of-birth	Token	Budapest (18), Prague (10), Istanbul (8), Athens (8), Paris (7), Moscow (7), Helsinki (6), Bucharest (6), Tehran (5), Stockholm (4)
	Reranking	London (30), Dublin (12), Paris (12), Moscow (5), Madrid (4), Philadelphia (4), Chicago (4), Warsaw (3), Tehran (3), Berlin (2)
place-of-death	Token	Paris (38), Rome (32), Moscow (6), Madrid (4), infancy (4), office (3), Athens (2), Helsinki (2), Warsaw (2), Amsterdam (2)
	Reranking	London (46), Paris (14), Rome (7), office (6), Moscow (4), Munich (3), Amsterdam (3), infancy (2), prison (2), Stockholm (2)

Table 4: Comparison of the most frequent tokens among the top-100 predictions based on different confidence scores. Based on the results on the Google-RE dataset with the BERT-base model. The numbers in parentheses represent the frequency of the predictions.

Model	Pred.	GRE	TREx	CNet	SQ	All
BERT-base	T	10.3	29.6	**15.8**	14.1	24.3
	S	**10.5**	29.6	14.6	14.4	24.1
	G	9.7	28.6	15.3	**15.1**	23.5
	DM	10.3	**29.8**	15.4	14.1	**24.4**
	DV	0.2	0.1	0.1	0.0	0.1
	TD	9.6	29.4	14.2	-	-
BERT-large	T	11.0	31.0	19.3	17.4	26.1
	S	**11.2**	31.5	17.6	15.7	26.1
	G	10.4	29.6	18.6	17.4	25.0
	DM	10.9	**31.7**	**19.6**	**17.7**	**26.7**
	DV	0.2	0.0	0.0	0.0	0.1
	TD	10.6	30.5	17.0	-	-
RoBERTa-base	T	7.5	23.0	**18.5**	14.7	20.2
	S	**8.2**	24.3	17.0	12.2	20.7
	G	7.6	22.0	17.4	14.7	19.3
	DM	8.0	**24.4**	18.3	**15.7**	**21.1**
	DV	0.1	0.1	0.1	0.0	0.1
	TD	7.5	23.2	16.4	-	-

Table 5: P@1 based on different prediction scores for each dataset. Bb: BERT-base, Bl: BERT-large, T: `Token`, S: `Sent`, G: `Gap`, DM: `DropoutMean`, DV: `DropoutVar`, TD: `TemplateDiff`, GRE: Google-RE, CNet: ConceptNet, SQ: SQuAD. We omit the result of using the `Reranking` score because the results are the same as those of `Sent` by definition.

An investigation of such cases remains a direction of future research.

6 Conclusion

In this paper, we introduced the selective prediction setting to the LAMA probe benchmark to evaluate both the amount of relational knowledge stored in a language model and the ability of the models to effectively filter out unconfident predictions. We compared different confidence functions that can be calculated using only the model parameters and the output information. The experimental results are summarized as follows:

- The selective prediction evaluation is more robust to template-related biases than the conventional accuracy-based evaluation (Table 1).
- The token log probability is not always the best choice, and the best confidence function depends on the language model and the dataset (Table 2).
- Different confidence functions have different preferences over relation types and predicted tokens, even though all functions are based solely on the model output (Figure 3, Table 4).
- There is no strong correlation between the performance of each confidence function as a predictor and a confidence estimator (Table 5).

Future studies will include a detailed analysis of the relationship between tasks, models, and confidence scores. Moreover, more sophisticated methods will be explored to ensure the reliability of language model predictions under various tasks. The code for our work is attached as supplementary material.

Limitations

In this paper, we focused on evaluating the predictions of masked language models on the LAMA probe bencmhark. Although our proposed framework is easily applicable to other kinds of language model with small adjustments, some of the confidence functions we investigated require properties specific to particular language models and datasets. For instance, `Token` and `Gap` functions require the prediction to be a single token, and `TemplateDiff` requires templates for subject-relation-object triples.

Ethics Statement

Data and code

In our experiments, we use the original LAMA benchmark dataset from Petroni et al. (2019) as is. All data are based on publicly available data sources and data statistics can be found in the original paper. Parts of the code are based on LAMA[2]. The license of the code can be found in the supplementary material.

Details of experiments

The experiments were conducted using a 2.4GHz CPU and an NVIDIA TESLA P100 GPU. Inference time was 1–1.5 s per instance for BERT-base and 2–3 s per instance for BERT-large.

Potential risks

This study evaluates the knowledge stored in language models considering the reliability of model predictions. However, it should be emphasized that the outputs of the selective classifier constructed by the proposed method do not guarantee the correctness of the model predictions. For the validation of each fact, this method should only be used as an aid, and the final decision should be made by the user.

Acknowledgements

We thank Prof. Simone Teufel, Marco Cognetta and anonymous reviewers for their valuable feedback. This study was carried out using the TSUBAME3.0 supercomputer at Tokyo Institute of Technology. This work was partially supported by JSPS KAKENHI Grant Number 19H01118.

References

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact-aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pages 267–284, USA. USENIX Association.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

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.

Ran El-Yaniv and Yair Wiener. 2010. On the Foundations of Noise-free Selective Classification. *Journal of Machine Learning Research*, 11(53):1605–1641.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd*

[2] https://github.com/facebookresearch/LAMA

International Conference on International Conference on Machine Learning - Volume 48, ICML'16, page 1050–1059. JMLR.org.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective Classification for Deep Neural Networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 4885–4894. Curran Associates Inc.

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.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. Which linguist invented the lightbulb? presupposition verification for question-answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3932–3945, Online. Association for Computational Linguistics.

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. Calibrated language model fine-tuning for in- and out-of-distribution data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1326–1340, Online. Association for Computational Linguistics.

Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How furiously can colorless green ideas sleep? sentence acceptability in context. *Transactions of the Association for Computational Linguistics*, 8:296–310.

Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. Towards few-shot fact-checking via perplexity. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Alessio Miaschi, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2020. Linguistic profiling of a neural language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 745–756, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's knowledge of language: Five analysis methods with NPIs. 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 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Michael Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.