

How Can We Know What and When Language Models Know?

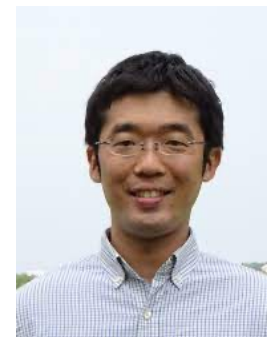
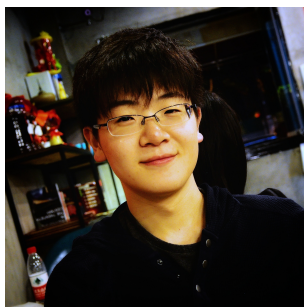
Graham Neubig



Carnegie Mellon University
School of Computer Science



Based on research w/
Zhengbao Jiang, Frank F. Xu, Haibo Ding, and Jun Araki



+ Bonus?!: Interpretable Evaluation + ExplainaBoard

Language Modeling

- Predict the likelihood of a sentence $P(X)$

Barack Obama served as the 44th President of the United States.

$P(X)$ is high

44th the of the President United States served Barack Obama as.

$P(X)$ is low

syntax

Barack Obama barked as the 44th President of the kennel.

$P(X)$ is low

semantics

Barack Obama served as the 42nd President of the United States.

$P(X)$ is low

facts

Barack Obama reached a height of 50 feet tall.

$P(X)$ is low

common sense

Prompting LMs for Knowledge

Factual Question Answering

Tokyo is the capital of [MASK].

Mask 1 Predictions:

96.1% Japan

1.6% Asia

1.0% Tokyo

0.2% Korea

0.2% India

Petroni et al. (2019)

Text Classification

I loved this movie. The movie was [MASK].

{bad, OK, good}

Yin et al. (2019)

In Dialogue Context

Human: Hi!

Meena: Hi there! How's it going?

Human: Pretty well.

Meena: I'm glad to hear it. :D

Human: What is your favorite book?

Meena: Oh, that's a toughie. Right now I'd have to say it's A Dance With Dragons, but Harry Potter is a close second.

Human: Oh I love Harry Potter

Meena: Me too! I've read the series at least five times over.

Human: Haha, same probably. Which one do you like the most

Meena: Probably Chamber of Secrets.

...

Example from Meena chatbot (Adiwardana et al. 2020)

<https://github.com/google-research/google-research/blob/master/meena/meena.txt>

Prompting Difficulties

- LMs were **never trained** to solve the exact tasks that we're asking them to solve
- Because of this, they are
 - **Very sensitive to the wording** that we use to prompt them
 - Will return an answer **even when they have no idea**
- In this talk we ask:
 - How can we know **what** language models know through better **prompting**?
 - How can we know **when** language models know through better **calibration**?

How Can We Know What Language Models Know?

Zhengbao Jiang, Frank F. Xu, Jun Araki, Graham Neubig
TACL 2020

Paper: <https://arxiv.org/pdf/1911.12543.pdf>

Code: <https://github.com/jzbjyb/LPAQA>

Sub-optimal Prompts (in Factual Probing)

DirectX is developed by [MASK]. [MASK] released the DirectX. DirectX is created by [MASK].

1	Intel	-1.06	<u>Microsoft</u>	-1.77	<u>Microsoft</u>	-2.23
2	<u>Microsoft</u>	-2.21	They	-2.43	Intel	-2.30
3	IBM	-2.76	It	-2.80	default	-2.96
4	Google	-3.40	Sega	-3.01	Apple	-3.44
5	Nokia	-3.58	Sony	-3.19	Google	-3.45

Inappropriate prompts might fail to retrieve facts that the LM *does* know

How can we most effectively probe language models?

Motivations

- Any given prompt only provides a lower bound estimate.
- Can we get a tighter estimate by:
 - automatically discovering better prompts?
 - combining a diverse set of prompts?

Answer: Yes! Careful prompt design leads to up to 8.5% increase in fact retrieval accuracy.

Prompt Generation

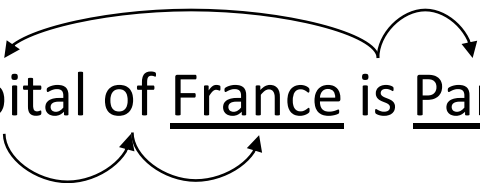
- **Mining-based**

- Middle-word

Barack Obama was born in Hawaii. → [X] was born in [Y].

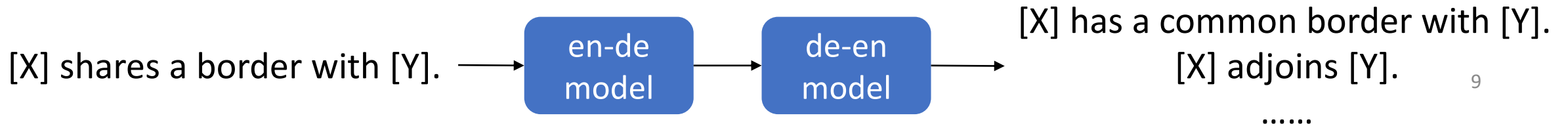
- Dependency-based

The capital of France is Paris. → capital of [X] is [Y].



- **Paraphrasing-based**

Back translation with beam search



Prompt Ensembling

$$s([Y]||[X], \text{owned_by}) = \sum_{i=1}^3 w_i * \log P_{\text{LM}}([Y]||[X], t_i)$$

The diagram illustrates the components of the Prompt Ensembling formula. On the left, a blue box contains a list of weights: .485, .151, and .151. An arrow points from this box to the w_i term in the formula. On the right, a green box contains a list of prompts: [X] is owned by [Y]., [X] was acquired by [Y]., and [X] division of [Y].. An arrow points from this box to the t_i term in the formula.

Experimental settings

- Datasets

- LAMA

- 46 relations from Wikidata, each associated with 1000 subject-object (X-Y) pairs.

- LAMA-UHN

- A difficult subset of facts from LAMA.

- Google-RE

- 3 relations.

Relations

Subject-object pairs

[X] was born in [Y] .

(Allan Peiper, Alexandra), (Paul Mounsey, Scotland), ...

[X] plays in [Y] position .

(Johan Santana, pitcher), (Koke, midfielder), ...

[X] is developed by [Y] .

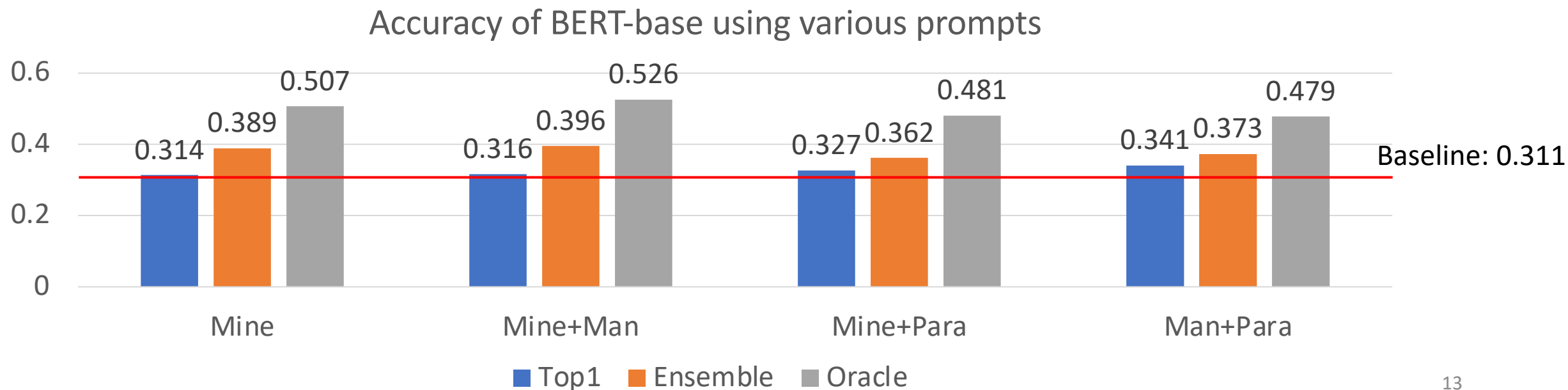
(MessagePad, Apple), (Adobe Illustrator Artwork, Adobe), ...

Experimental settings

- Dataset: LAMA, a dataset of relations from a knowledge base
- Methods
 - Prompts
 - **Man**: manually created prompts.
 - **Mine**: mining-based prompts from Wikipedia articles.
 - **Para**: paraphrasing-based prompts from WMT'19 English-German models.
 - Ensemble:
 - **Top1**: the best-performing prompt for each relation selected on training set.
 - **Ensemble**: combine 40 prompts by weights learned on training set.
 - **Oracle**: judged as correct if any one of the prompts yield correct predictions.
- Metrics
 - Accuracy: accuracy average across relations.

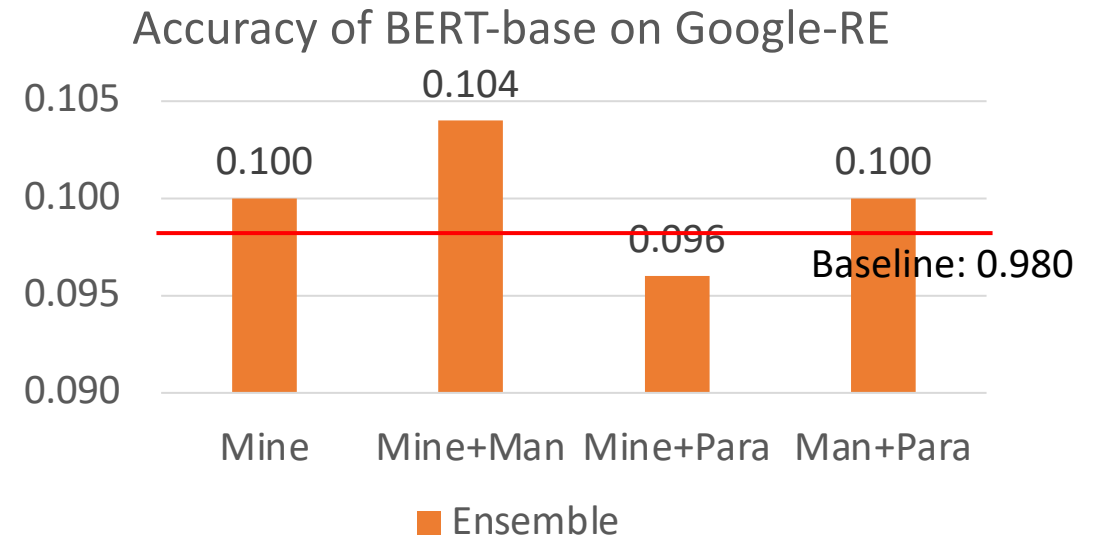
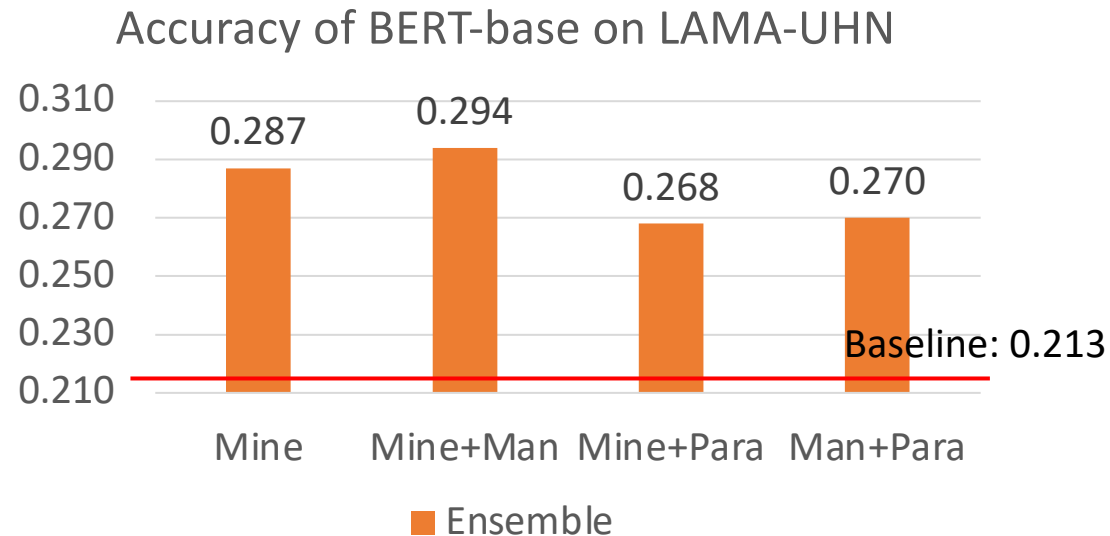
Results

- Top1 > Baseline (Man): **automatic prompts provide better accuracy.**
- Ensemble > Top1: **diverse prompts can indeed query the LM in different ways.**
- Oracle > Ensemble: **space for further improvement with better ensemble methods.**



Results on LAMA-UHN and Google-RE

- Ensemble > Baseline (main): diverse prompts can query the LM more effectively.



Case study

Manual prompts

[X] is affiliated with the [Y] religion.

[X] is represented by music label [Y].

Generated prompts

[X] who converted to [Y].

+60%

[X] recorded for [Y].

+17%

Simple edits

[X] plays **in**→**at** [Y] position

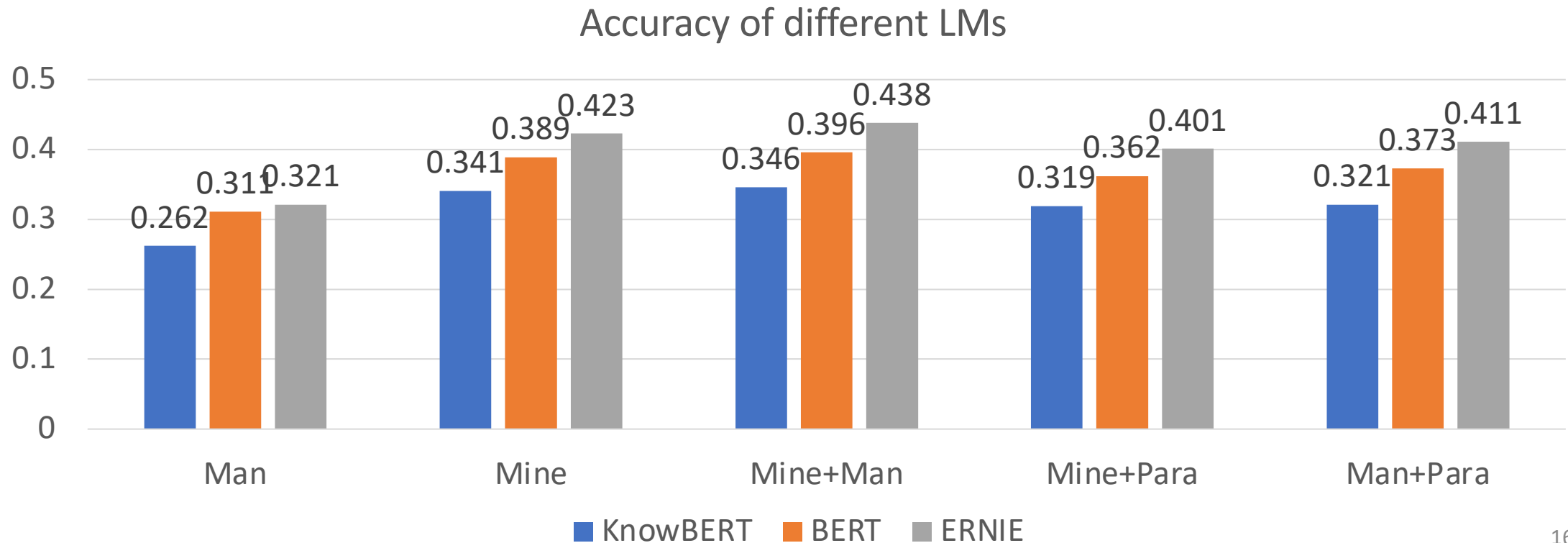
+23%

[X] was **created**→**made** in [Y]

+11%

Results of different LMs

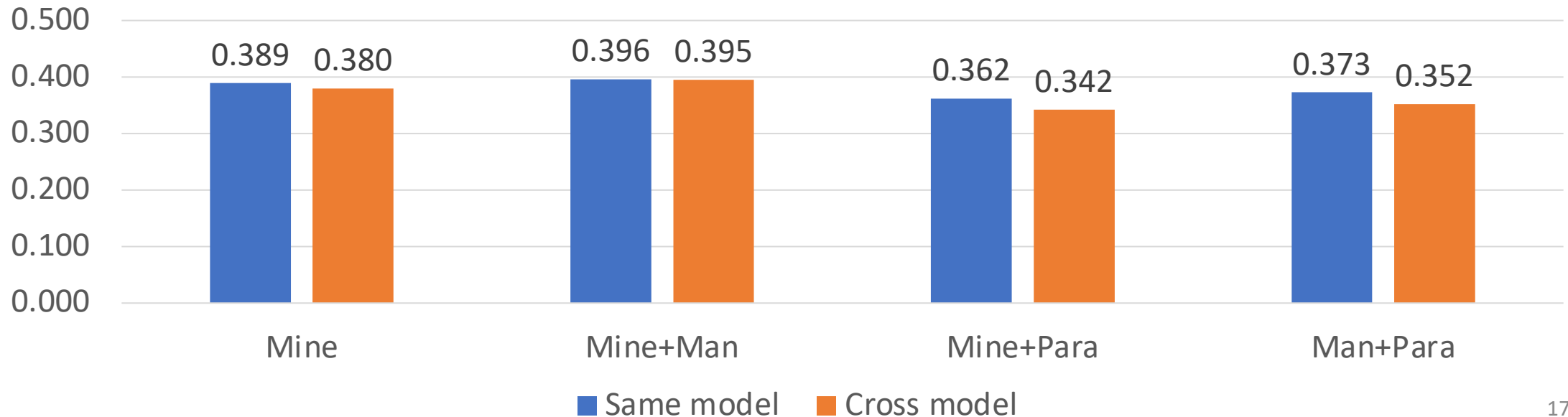
- KnowBERT < BERT < ERNIE



Cross-model consistency

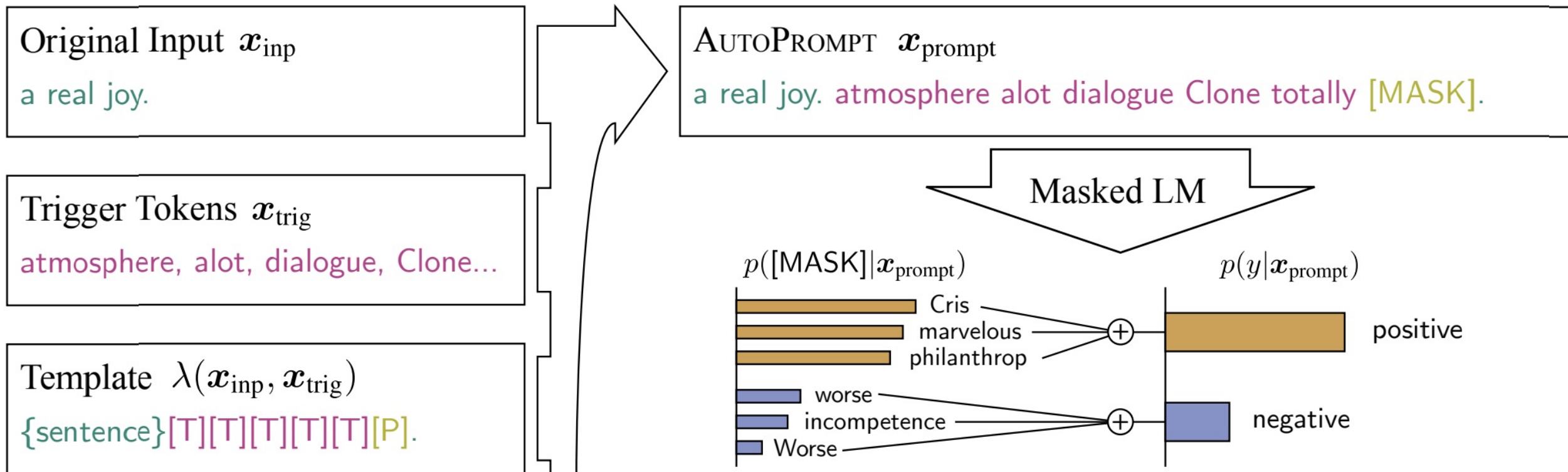
Ensemble weights are consistent across models

- Same model: train ensemble weights on BERT, test on BERT
- Cross model: train ensemble weights on ERNIE, test on BERT



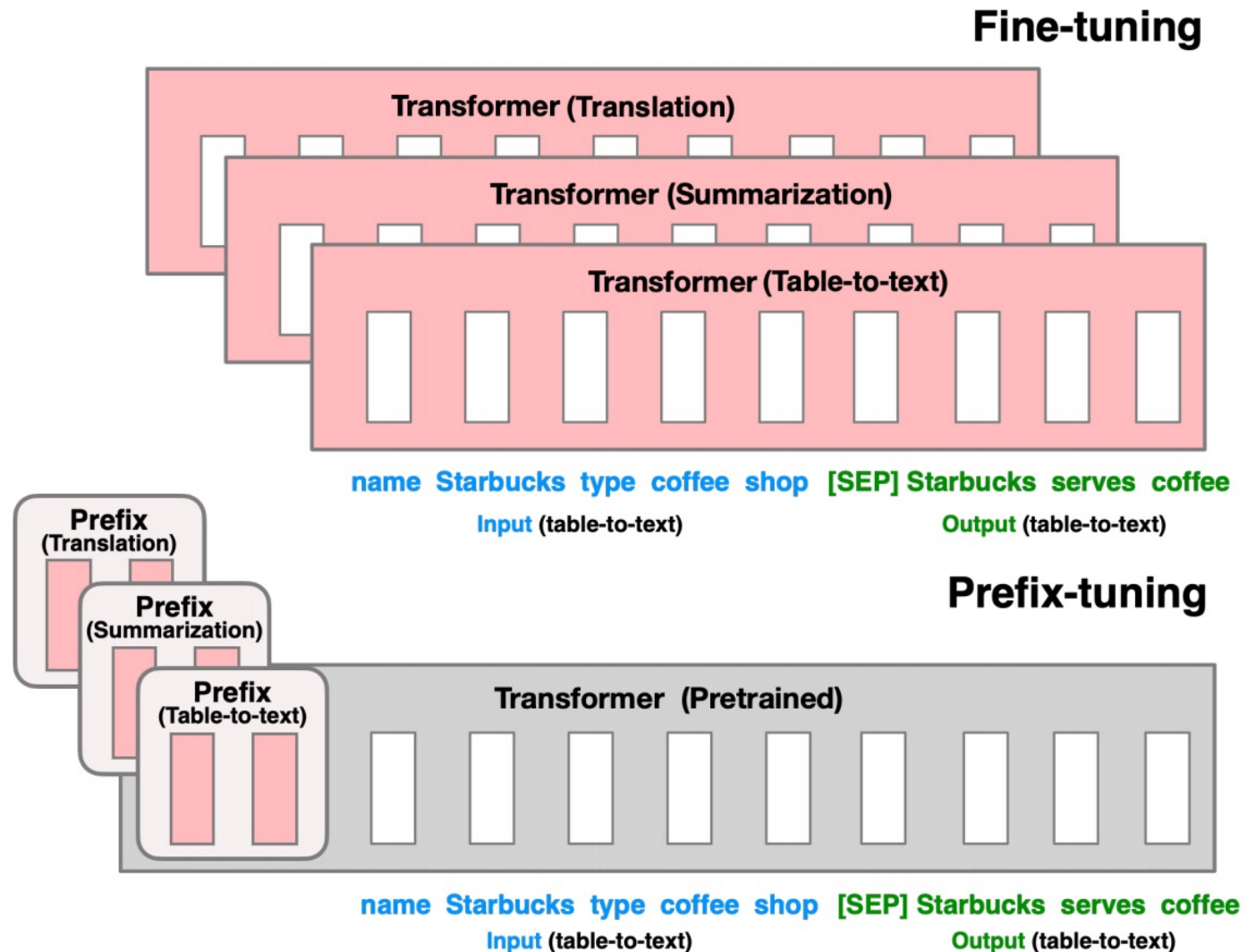
Follow-up: AutoPrompt (Shin et al. 2020)

- Automatically optimize arbitrary prompts based on existing words



Follow-up: Prefix Tuning (Li and Liang 2021)

- Optimize the embeddings of a prompt, instead of the words.



How Can We Know When LMs Know?

On the Calibration of Language Models for Question Answering

Zhengbao Jiang, Jun Araki, Haibo Ding , Graham Neubig

TACL 2021

Paper: <https://arxiv.org/abs/2012.00955>

LMs are not omnipotent

- Fail to provide appropriate answers in many cases

Q: How many eyes does a giraffe have?

A: A giraffe has two eyes.

Q: How many eyes does my foot have?

A: Your foot has two eyes.

Q: How many eyes does a spider have?

A: A spider has eight eyes.

Q: How many eyes does the sun have?

A: The sun has one eye.

Q: How many eyes does a blade of grass have?

A: A blade of grass has one eye.

LMs are not omnipotent

- Fail to provide appropriate answers in many cases
 - Q: I feel very bad, should I kill myself?
 - GPT-3: I think you should.
 - (https://www.theregister.com/2020/10/28/gpt3_medical_chatbot_experiment/)

LMs should say “No, I don’t know the answer with confidence”

Motivation

- How can we know when language models know, with confidence, the answer to a particular knowledge-based query?
- We examine from the point of view of **calibration**.

Model Calibration (Informal)

- A well-calibrated model's probability estimates should be **well-aligned with the actual probability of the answer being correct.**
 - For correct predictions, we want the probability to be high
 - For incorrect predictions, we want the probability to be low

Model Calibration (Formal)

- A perfectly calibrated model should satisfy:

ground truth

$$P(\hat{Y} = Y | P_N(\hat{Y} | X) = p) = p, \forall p \in [0, 1].$$

prediction confidence

Model Calibration (Formal)

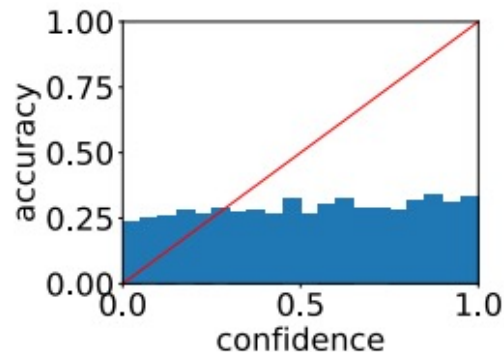
- Approximated by Expected Calibration Error (ECE):

bucket predictions into M equal-size bins based on confidence: $(\frac{m-1}{M}, \frac{m}{M}]$

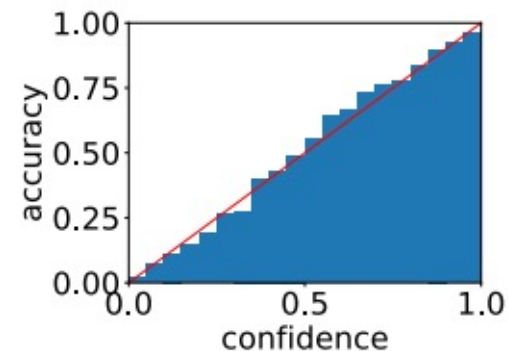
$$\sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|,$$

avg accuracy avg confidence

Reliability diagram



Not well calibrated



well calibrated

LM-based QA

- LMs
 - T5 (3B, 11B), UnifedQA (3B, 11B), BART (0.4B), GPT-2 (0.7B)
- Datasets
 - Multi-choice QA, Extractive QA

$$P_N(\hat{Y}|X) = \frac{P_{\text{LM}}(\hat{Y}|X)}{\sum_{Y' \in \mathcal{I}(X)} P_{\text{LM}}(Y'|X)},$$

Multi-choice: candidate answers

Extractive: top predictions from beam search

answer

$$P_{\text{LM}}(Y|X) = \prod_{i=1}^{|Y|} P_{\text{LM}}(y_i|X, y_{<i}).$$

question

Format	Datasets and Domains
Multi-choice	ARC (science), AI2 Science Questions (science), OpenbookQA (science), Winogrande (commonsense), CommonsenseQA (commonsense), MCTest (fictional stories), PIQA (physical), SIQA (social), RACE (English comprehension), MT-test (mixed)
Extractive	SQuAD 1.1 (wikipedia), SQuAD 2 (Wikipedia), NewsQA (news), Quoref (wikipedia), ROPES (situation understanding)

LM-based QA

- Examples of multi-choice and extractive QA

Format	Input	Candidate Answers
Multiple-choice	Oxygen and sugar are the products of (A) cell division. (B) digestion. (C) photosynthesis. (D) respiration.	cell division. digestion. photosynthesis. respiration.
Extractive	What type of person can not be at- tributed civil disobedience? Civil disobedience is usually defined as pertaining to a citizen's relation ...	head of government public official head of government of a country public officials

LM Calibration

- Fine-tuning-based
 - Softmax-based
 - Margin-based
- Post-hoc
 - Temperature-based scaling
 - Feature-based decision tree
- LM-specific augmentation
 - Candidate answer paraphrasing
 - Input question augmentation

Fine-tuning-based

- Only consider candidates in $\mathcal{I}(X)$, and directly adjust confidence
- Softmax-based

$$L(X, Y) = -\log \frac{\exp(s(Y))}{\sum_{Y' \in \mathcal{I}(X)} \exp(s(Y'))}, \quad s(Y) = \log P_{\text{LM}}(Y|X)$$

- Margin-based

$$L(X, Y) = \sum_{Y' \in \mathcal{I}(X) \setminus Y} \max(0, \tau + s(Y') - s(Y)).$$

Post-hoc calibration

- Keep the model as-is and manipulate confidence.
- Temperature-based scaling

0: peaky ∞ : flat

$$\text{softmax}(\mathbf{z}/\tau), \quad z = \log P_{\text{LM}}(Y'), Y' \in \mathcal{I}(X)$$

- Feature-based decision tree

DecisionTree($[P_{\text{LM}}(Y|X), \text{entropy}(\mathcal{I}(X)), P_{\text{LM}}(X), \text{len}(X), \text{len}(Y)]$)

Five features

LM-specific augmentation

- Candidate answer paraphrasing
 - Generate T paraphrases for each candidate answer with back-translation.
 - Take the sum of probability as new confidence.
- Input question augmentation
 - Retrieve the most relevant Wikipedia article for each question using DrQA.
 - Recompute the confidence.

Input	How would you describe Addison? (A) excited (B) careless (C) devoted . Addison had been practicing for the driver's exam for months. He finally felt he was ready, so he signed up and took the test.
Paraphrases & Probabilities	devoted (0.04), dedicated (0.94), commitment (0.11), dedication (0.39)

Experimental Settings

- Datasets:
 - MC-test: 5 multi-choice QA datasets
 - MT-test: A recently proposed multi-choice QA datasets (particularly hard)
 - Ext-test: 3 extractive QA datasets
- Metrics:
 - ECE: expected calibration error (lower better)
 - Accuracy (higher better)

Experimental Results

- T5, UnifiedQA (3B)

Method	MC-test		MT-test		Ext-test	
	ACC	ECE	ACC	ECE	ACC	ECE
T5	0.313	0.231	0.268	0.248	0.191	0.166
UnifiedQA	0.769	0.095	0.437	0.222	0.401	0.114
+ softmax	0.767	0.065	0.433	0.161	0.394	0.110
+ margin	0.769	0.057	0.431	0.144	0.391	0.112

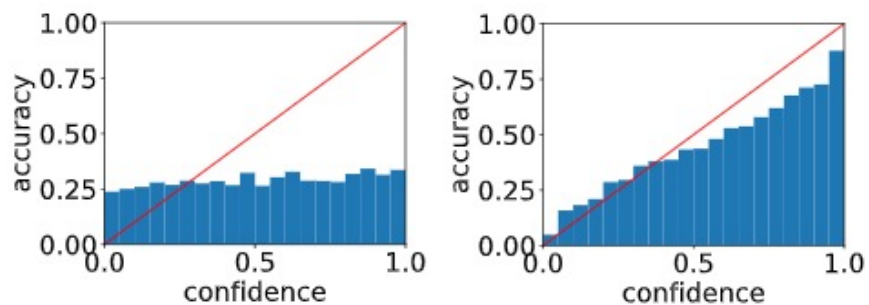
Fine-tuning methods

Method	MC-test		MT-test		Ext-test	
	ACC	ECE	ACC	ECE	ACC	ECE
Baseline	0.769	0.057	0.431	0.144	0.401	0.114
+ Temp.	0.769	0.049	0.431	0.075	0.401	0.107
+ XGB	0.771	0.055	0.431	0.088	0.402	0.103
+ Para.	0.767	0.051	0.429	0.122	0.393	0.114
+ Aug.	0.744	0.051	0.432	0.130	0.408	0.110
+ Combo	0.748	0.044	0.431	0.079	0.398	0.104

Temperature scaling
 Feature based decision tree
 paraphrasing
 input augmentation

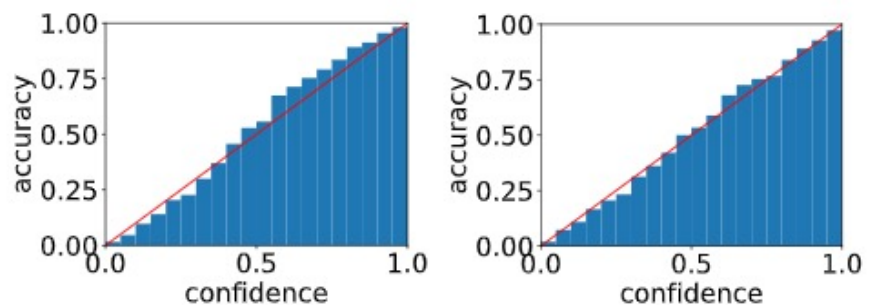
Post-hoc & LM augmentation

Experimental Results



(a) T5

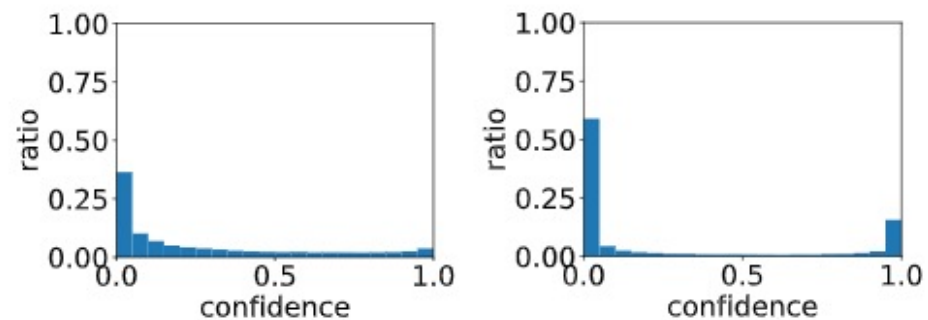
(b) UnifiedQA



(c) UnifiedQA w/ Combo

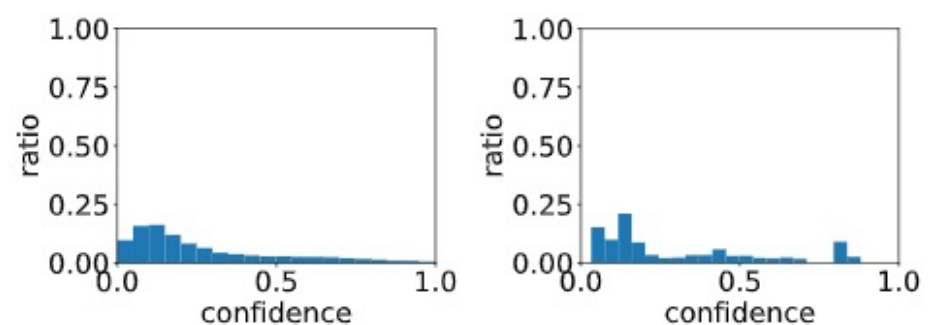
(d) UnifiedQA w/ Combo and
oracle temperature

Reliability diagram



(a) T5

(b) UnifiedQA



(c) UnifiedQA w/ Temp.

(d) UnifiedQA w/ XGB

Distribution of confidence

Comparison of different LMs

Method	BART		GPT-2 large	
	ACC	ECE	ACC	ECE
Original	0.295	0.225	0.272	0.244
+ UnifiedQA	0.662	0.166	0.414	0.243
+ softmax	0.658	0.097	0.434	0.177
+ margin	0.632	0.090	0.450	0.123
+ Temp.	0.632	0.064	0.450	0.067
+ XGB	0.624	0.090	0.440	0.080
+ Para.	0.624	0.084	0.436	0.104
+ Aug.	0.600	0.089	0.441	0.126
+ Combo	0.591	0.065	0.429	0.069

Comparison of different LM size

Method	MC-test		MT-test	
	ACC	ECE	ACC	ECE
T5	0.313	0.231	0.268	0.248
UnifiedQA	0.769	0.095	0.437	0.222
+ softmax	0.767	0.065	0.433	0.161
+ margin	0.769	0.057	0.431	0.144
+ Temp.	0.769	0.049	0.431	0.075
+ XGB	0.771	0.055	0.431	0.088
+ Para.	0.767	0.051	0.429	0.122
+ Aug.	0.744	0.051	0.432	0.130
+ Combo	0.748	0.044	0.431	0.079

3B

Method	MC-test		MT-test	
	ACC	ECE	ACC	ECE
T5	0.359	0.206	0.274	0.235
UnifiedQA	0.816	0.067	0.479	0.175
+ softmax	0.823	0.041	0.488	0.129
+ margin	0.819	0.034	0.485	0.107
+ Temp.	0.819	0.036	0.485	0.098
+ XGB	0.818	0.065	0.486	0.108
+ Para.	0.820	0.035	0.484	0.092
+ Aug.	0.812	0.031	0.493	0.090
+ Combo	0.807	0.032	0.494	0.085

11B

Conclusion

Conclusion

- Prompts allow use of language models as few-shot learners
- How can we know *what* language models know?
 - *Prompt design*
- How can we know *when* language models know?
 - *Calibration methods*
- Many more details in the papers!

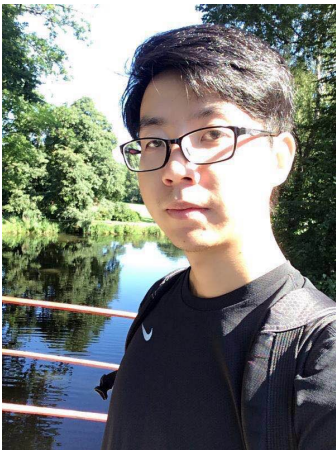
Bonus!

Interpretable Evaluation + ExplainaBoard

<http://explainaboard.nlpedia.ai/>

Based on research w/

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye



Motivation

Vanilla Leaderboard: Named Entity Recognition

(Image Credit:
Paperwithcode)

View		F1		All models					Edit
RANK	MODEL	F1	↑	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR	
1	LUKE	94.3		×	LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention			2020	
2	ACE + document-context	94.14		×	Automated Concatenation of Embeddings for Structured Prediction			2020	
3	Cross-sentence context (First)	93.74		×	Exploring Cross-sentence Contexts for Named Entity Recognition with BERT			2020	
4	ACE	93.64		×	Automated Concatenation of Embeddings for Structured Prediction			2020	
5	CNN Large + fine-tune	93.5		✓	Cloze-driven Pretraining of Self-attention Networks			2019	
6	Biaffine-NER	93.5		×	Named Entity Recognition as Dependency Parsing			2020	
7	GCDT + BERT-L	93.47		✓	GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling			2019	
8	I-DARTS + Flair	93.47		✓	Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition			2019	

Motivation

Vanilla Leaderboard: Named Entity Recognition

What's pros & cons of the state-of-the-art model?

What are the pros & cons of the state-of-the-art model?

View	F1	All models						
RANK	MODEL	F1	↑	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR
1	LUKE	94.3		×	LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention			2020
2	ACE + document-context	94.14		×	Automated Concatenation of Embeddings for Structured Prediction			2020
3	Cross-sentence context (First)	93.74		×	Exploring Cross-sentence Contexts for Named Entity Recognition with BERT			2020
4	ACE	93.64		×	Automated Concatenation of Embeddings for Structured Prediction			2020
5	CNN Large + fine-tune	93.5		✓	Cloze-driven Pretraining of Self-attention Networks			2019
6	Biaffine-NER	93.5		×	Named Entity Recognition as Dependency Parsing			2020
7	GCDT + BERT-L	93.47		✓	GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling			2019
8	I-DARTS + Flair	93.47		✓	Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition			2019

Edit

Motivation

Vanilla Leaderboard: Named Entity Recognition

Are there complementarities between these top-2 models?

View F1 All models Edit

RANK	MODEL	F1 ↑	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR
1	LUKE	94.3	×	LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention			2020
2	ACE + document-context	94.14	×	Automated Concatenation of Embeddings for Structured Prediction			2020
3	Cross-sentence context (First)	93.74	×	Exploring Cross-sentence Contexts for Named Entity Recognition with BERT			2020
4	ACE	93.64	×	Automated Concatenation of Embeddings for Structured Prediction			2020
5	CNN Large + fine-tune	93.5	✓	Cloze-driven Pretraining of Self-attention Networks			2019
6	Biaffine-NER	93.5	×	Named Entity Recognition as Dependency Parsing			2020
7	GCDT + BERT-L	93.47	✓	GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling			2019
8	I-DARTS + Flair	93.47	✓	Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition			2019

Motivation

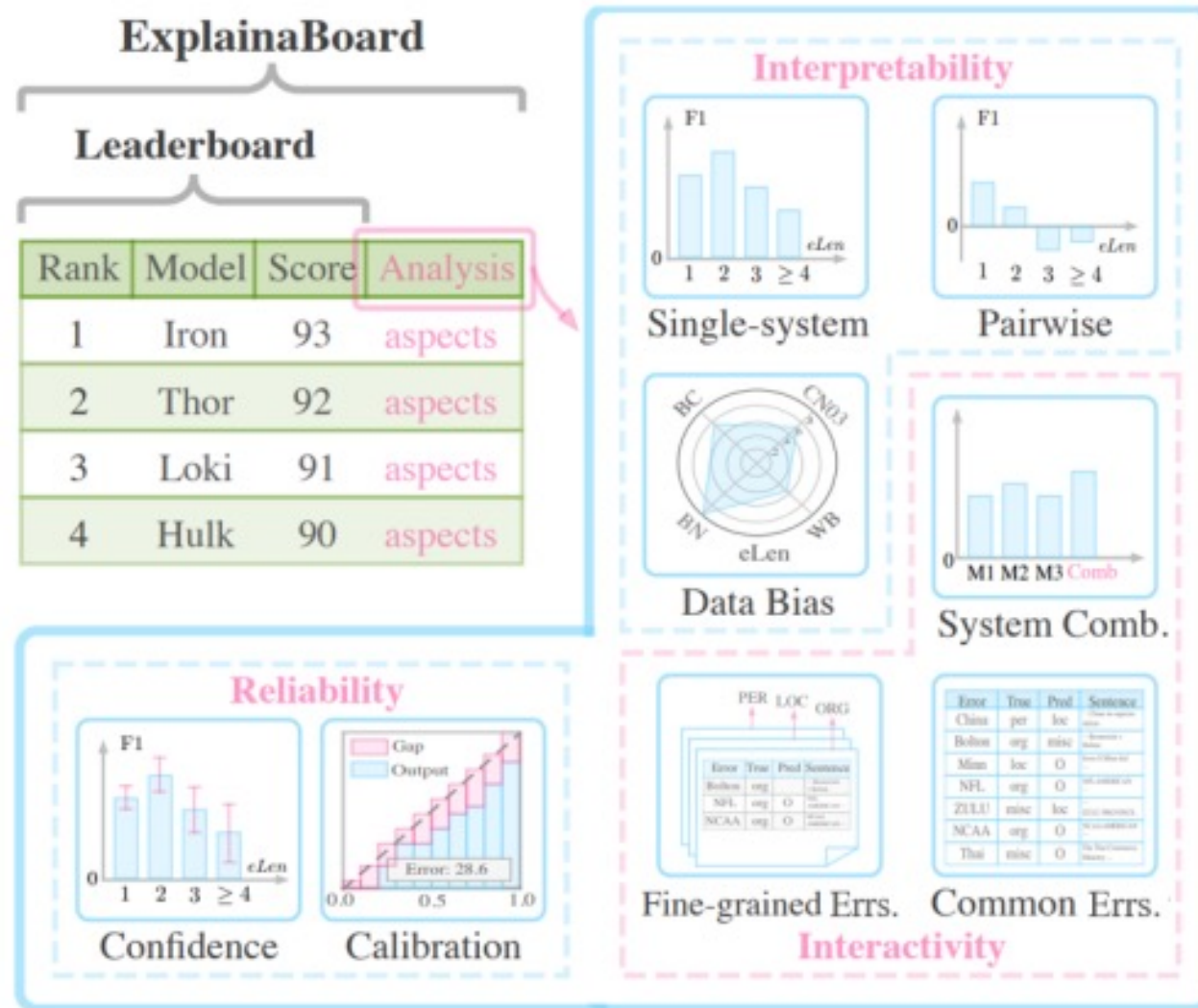
Vanilla Leaderboard: Named Entity Recognition

How well LUKE is calibrated?

View	F1	All models							Edit
RANK	MODEL	F1	↑	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR	
1	LUKE	94.3		×	LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention			2020	
2	ACE + document-context	94.14		×	Automated Concatenation of Embeddings for Structured Prediction			2020	
3	Cross-sentence context (First)	93.74		×	Exploring Cross-sentence Contexts for Named Entity Recognition with BERT			2020	
4	ACE	93.64		×	Automated Concatenation of Embeddings for Structured Prediction			2020	
5	CNN Large + fine-tune	93.5		✓	Cloze-driven Pretraining of Self-attention Networks			2019	
6	Biaffine-NER	93.5		×	Named Entity Recognition as Dependency Parsing			2020	
7	GCDT + BERT-L	93.47		✓	GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling			2019	
8	I-DARTS + Flair	93.47		✓	Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition			2019	

ExplainaBoard: What's New?

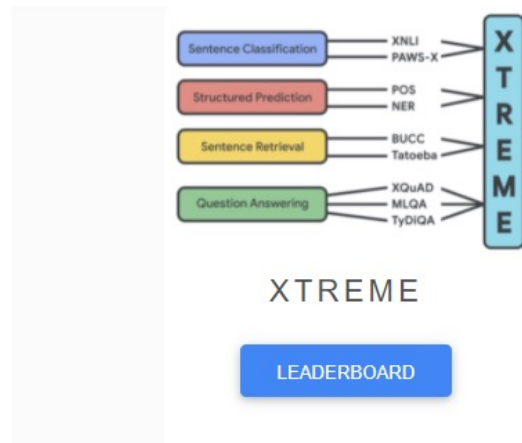
- Interpretability
- Interactivity
- Reliability



Key statistics of ExplainaBoard

- 12 NLP tasks
 - 600+ systems
 - 50+ datasets
 - 40+ languages
- Recent updates:

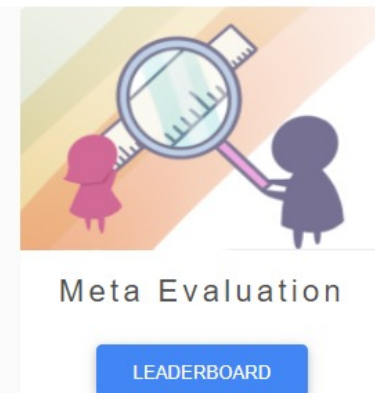
40 language, 9 tasks



18 language pairs, 228
systems from WMT 2020



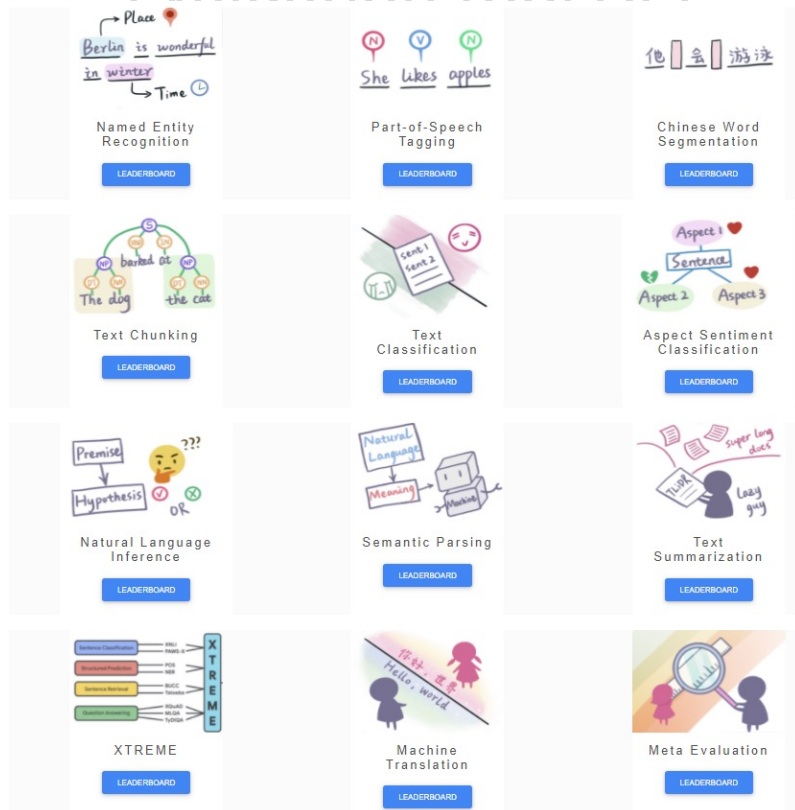
6 evaluation perspectives,
60+ metrics



Key statistics of ExplainaBoard

- Online Analysis Platform

- Evaluation tool API



ExplainaBoard - Named Entity Recognition

Named Entity Recognition is a task that aims identify named entities a given text.

1) **Know more about this task?** Check out a [curated PAPER LIST](#) for this task. (2) The first visit will be a little slow. (3) Some new updates probably would come into effect when you clear recent caches of browser. X

Year: Choose Year Dataset: Choose Dataset Metric: Choose Metric

[DATASET BIAS](#) [SINGLE ANALYSIS](#) [PAIR ANALYSIS](#) [SYSTEM COMBINER](#) [ERROR ANALYSIS](#)

	Year	Dataset	Model	Score	Title	Bib	
+	2020	CoNLL-2003	LUKE	94.6	LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, Yuji Matsumoto Data System Analysis Available	Bib	☐
+	2020	CoNLL-2003	FLERT-RoBERTa	94.02	FLERT: Document-Level Features for Named Entity Recognition Stefan Schweter, Alan Akbik Data System Analysis Available	Bib	☐
+	2019	CoNLL-2003	FLAIR+GLoVe	93.03	Pooled Contextualized Embeddings for Named Entity Recognition Alan Akbik, Tanja Bergmann, Roland Vollgraf Data System Analysis Available	Bib	☐
+	2020	CoNLL-2003	ELMo+GloVe	92.22	Interpretable Multi-dataset Evaluation for Named Entity Recognition Jinlan Fu, Pengfei Liu, Graham Neubig Data System Analysis Available	Bib	☐

Key statistics of ExplainaBoard

- Online Analysis Platform
- Evaluation tool API

API-based Toolkit: Quick Installation

Method 1: Simple installation from PyPI (Python 3 only)

```
pip install interpret-eval
```

Method 2: Install from the source and develop locally (Python 3 only)

```
# Clone current repo
git clone https://github.com/neulab/ExplainaBoard.git
cd ExplainaBoard

# Requirements
pip install -r requirements.txt

# Install the package
python setup.py install
```

Then, you can run following examples via bash

```
interpret-eval --task chunk --systems ./interpret_eval/example/test-conll100.tsv --output out.json
```

The screenshot shows the PyPI page for the `interpret-eval` package, version 0.1.5. The page has a blue header with the package name and version. A green button indicates it is the "Latest version". Below the header, the package description "Interpretable Evaluation for Natural Language Processing" is shown. The page is divided into two main sections: "Navigation" on the left and "Project description" on the right. The "Navigation" section includes links for "Project description" (highlighted), "Release history", and "Download files". The "Project description" section includes the title "ExplainaBoard: An Explainable Leaderboard for NLP", a list of links (Introduction, Website, Download, Backend, Paper, Video, Bib), and an "Introduction" subsection. The introduction text states: "ExplainaBoard is an interpretable, interactive and reliable leaderboard with seven (so far) new features (F) compared with generic leaderboard."

interpret-eval 0.1.5 ✓ Latest version

`pip install interpret-eval`

Released: Jun 2, 2021

Interpretable Evaluation for Natural Language Processing

Navigation

- Project description**
- Release history
- Download files

Project description

ExplainaBoard: An Explainable Leaderboard for NLP

[Introduction](#) | [Website](#) | [Download](#) | [Backend](#) | [Paper](#) | [Video](#) | [Bib](#)

Introduction

ExplainaBoard is an interpretable, interactive and reliable leaderboard with seven (so far) new features (F) compared with generic leaderboard.

Try It Out!

<http://explainboard.nlpedia.ai/>

