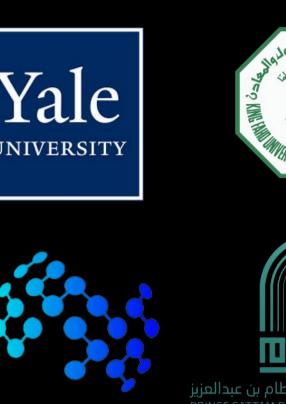
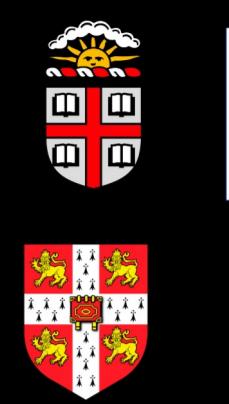
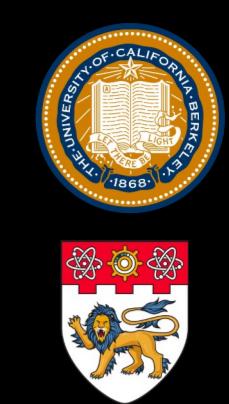
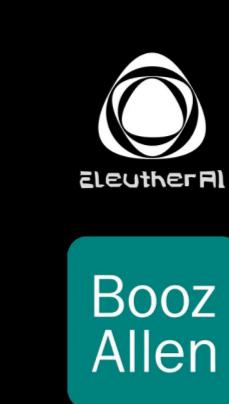


Crosslingual Generalization through Multitask Finetuning

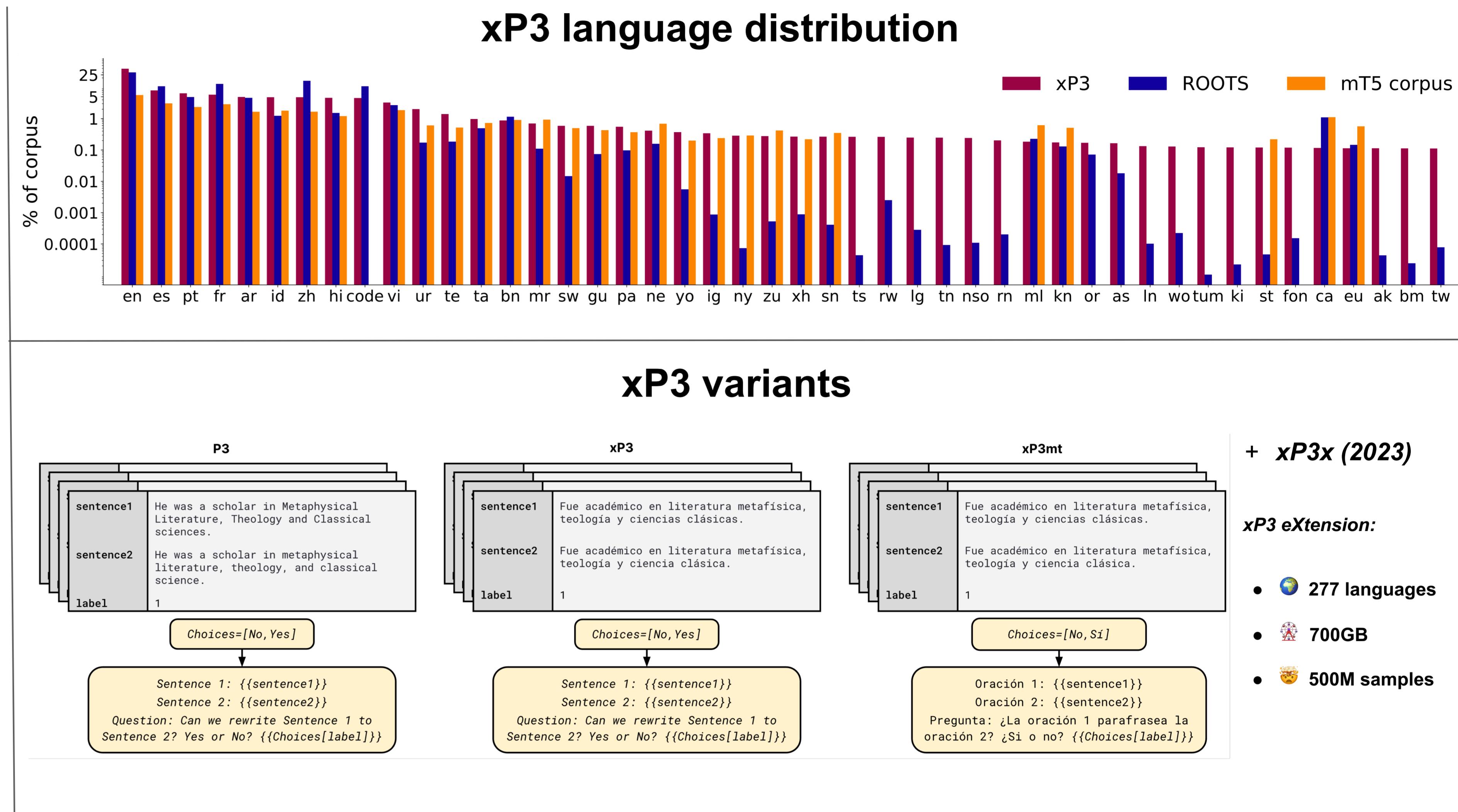
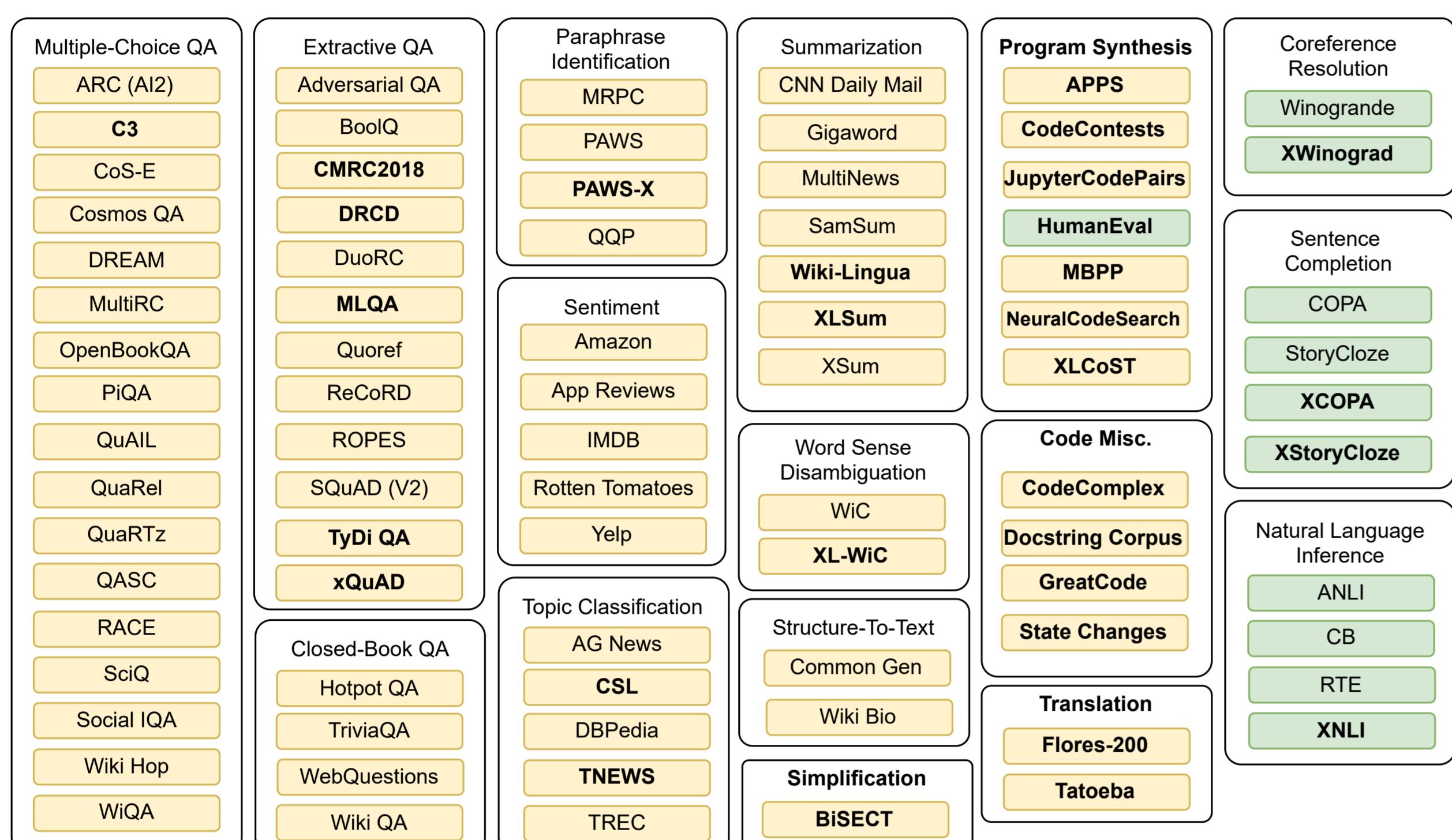
Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, Colin Raffel



PAPER

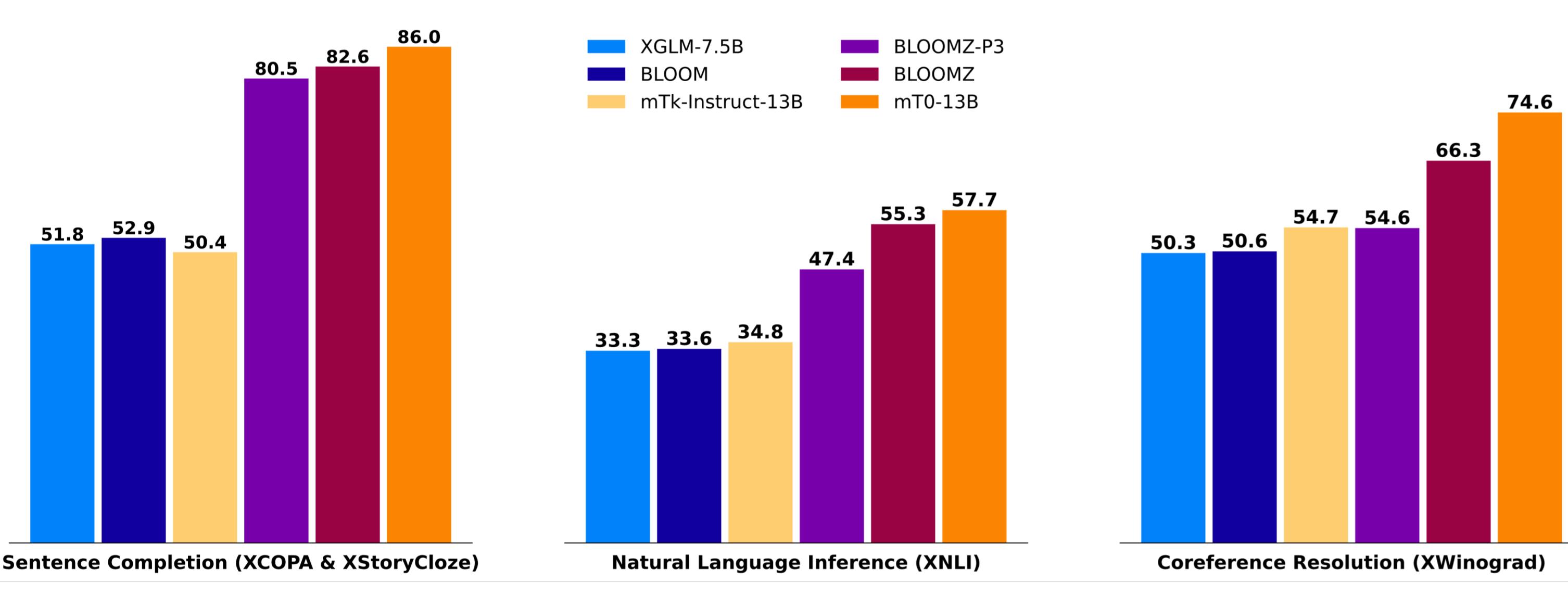
I. xP3: Crosslingual Corpus of prompted tasks in up to 277 languages

xP3 datasets & tasks divided into training (yellow) & evaluation (green)

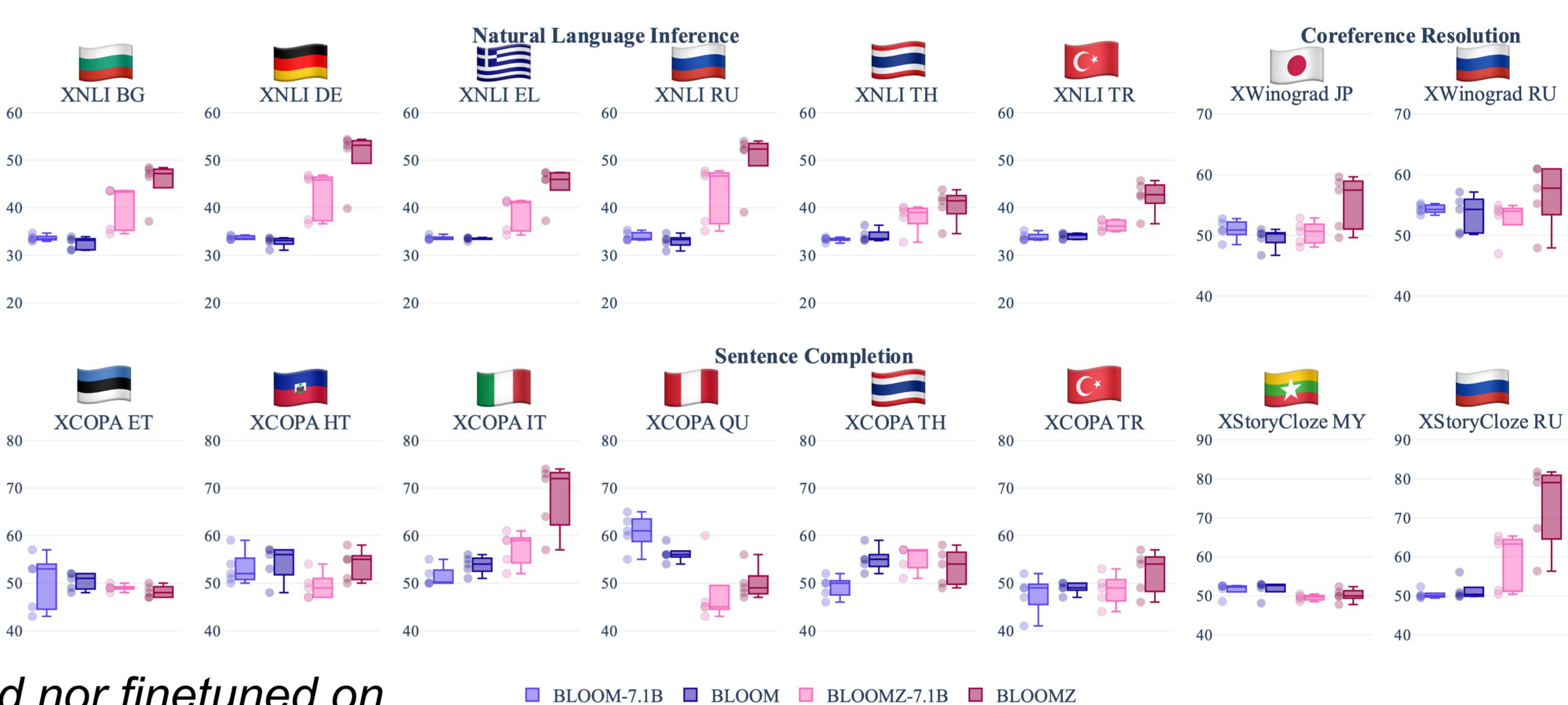


II. BLOOMZ & mT0: Instruction-tuned multilingual LLMs with up to 176B parameters

Generalization to unseen tasks

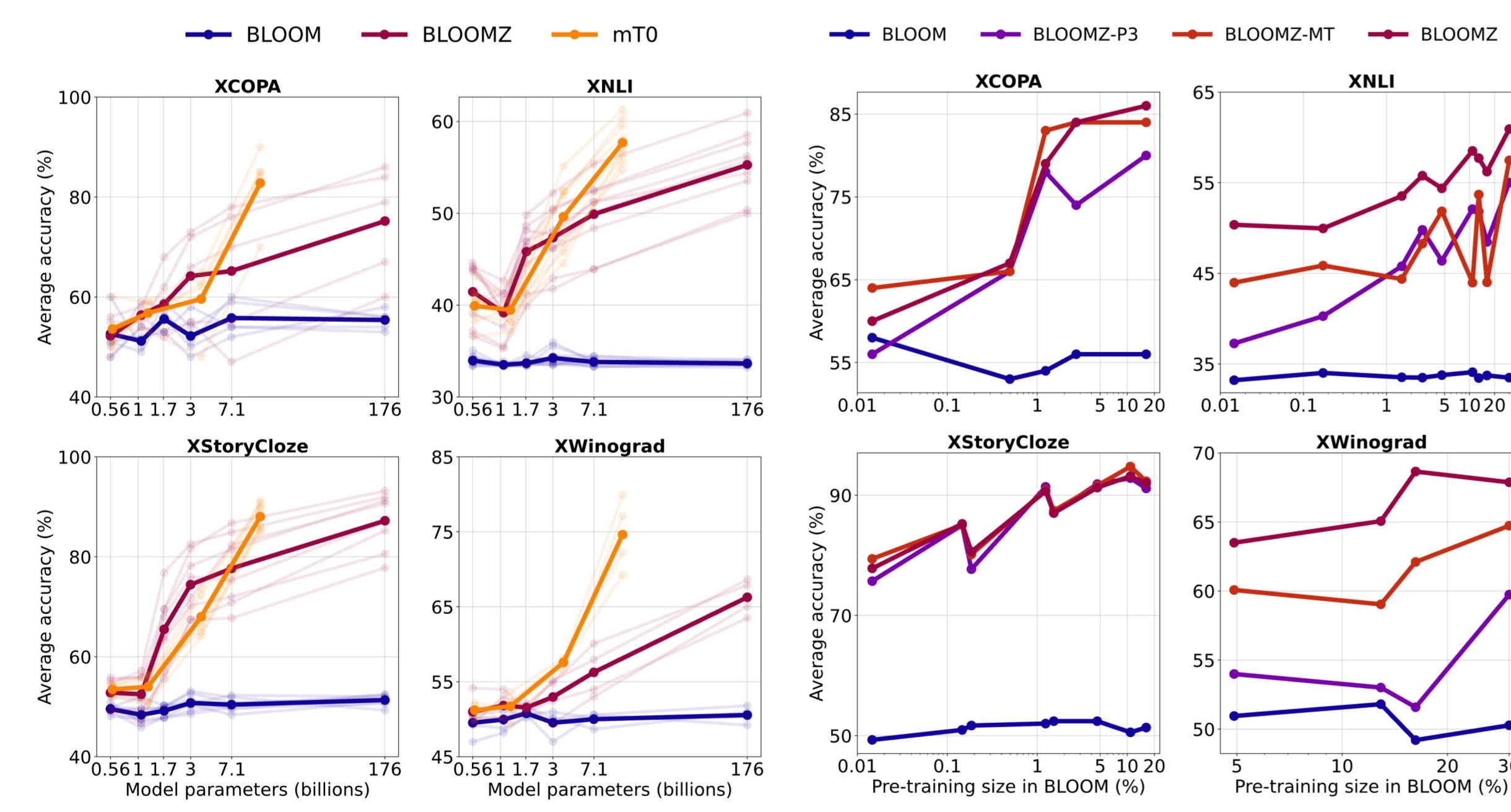


Generalization to unseen tasks in “unseen” languages

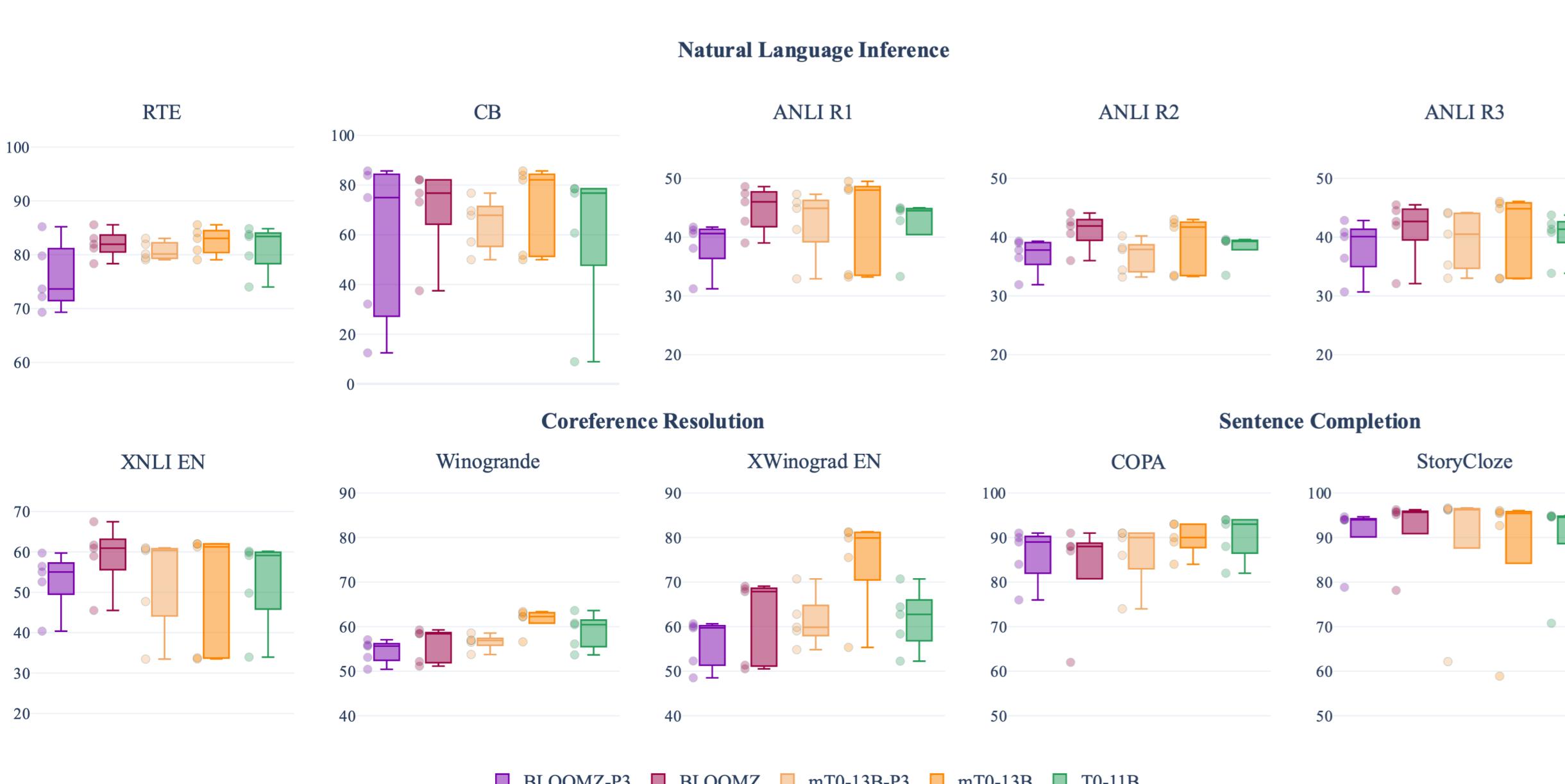


unseen = not pretrained nor finetuned on

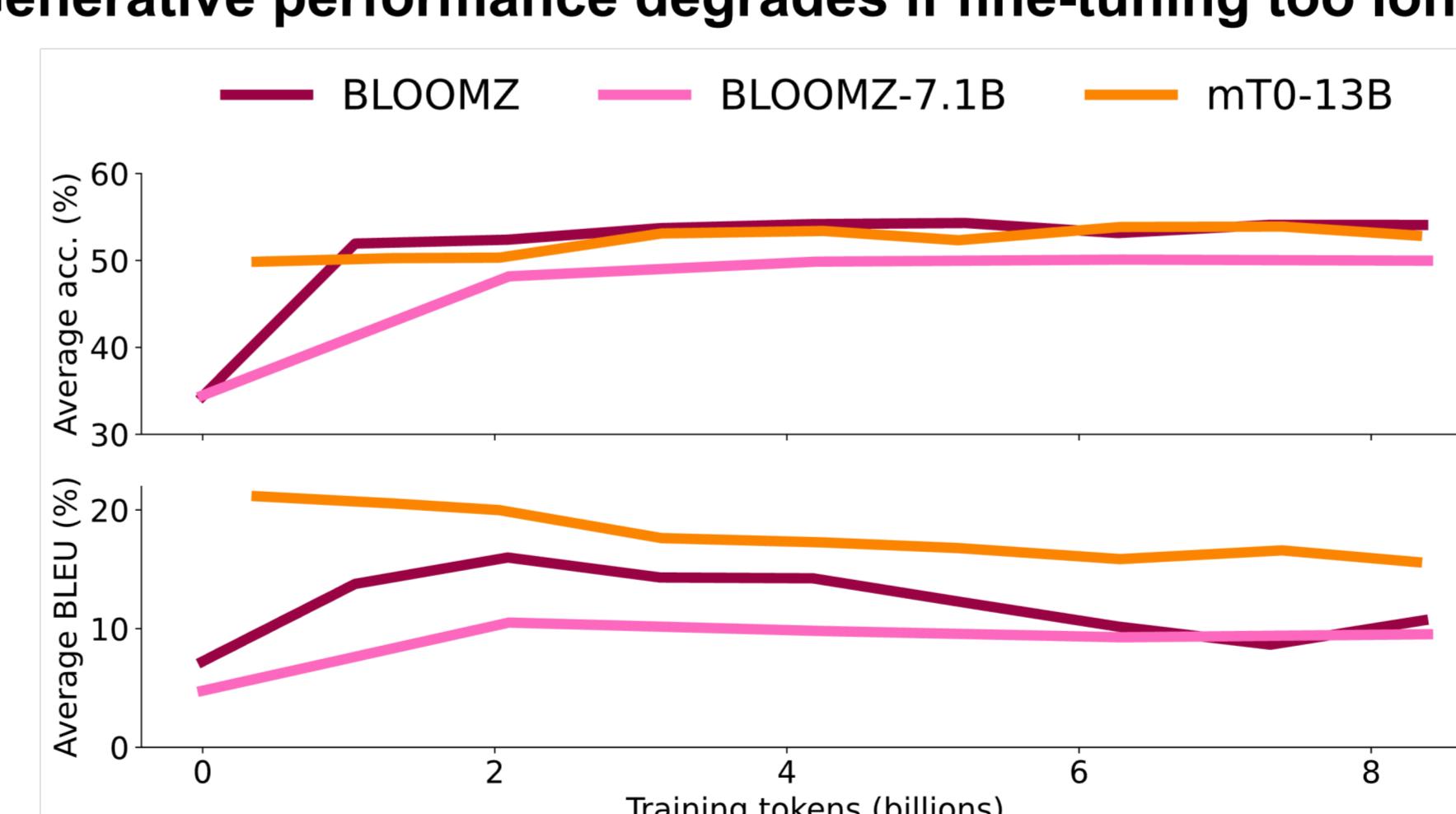
Scaling from 560M to 176B parameters & importance of language distributions



mT0 beats T0 on English Tasks



Generative performance degrades if fine-tuning too long!



Generation

Code Generation on HumanEval

```
from typing import List

BLOOM:
def has_close_elements(numbers: List[float], threshold: float) -> bool:
    """ Check if in given list of numbers, are any two numbers closer to each other than given threshold.
    >>> has_close_elements([1.0, 2.0, 3.0], 0.5)
    False
    >>> has_close_elements([1.0, 2.8, 3.0, 4.0, 5.0, 2.0], 0.3)
    True
    """
    if len(numbers) <= 1:
        return False
    numbers = sorted(numbers)
    i = 0
    while i < len(numbers) - 1:
        # Get distance between current and next elements
        distance = numbers[i] - numbers[i + 1]
        # Check if distance is closer than threshold
        if distance < threshold:
            return True
        i += 1
    return False

from typing import List

BLOOMZ:
def has_close_elements(numbers: List[float], threshold: float) -> bool:
    """ Check if in given list of numbers, are any two numbers closer to each other than given threshold.
    >>> has_close_elements([1.0, 2.0, 3.0], 0.5)
    False
    >>> has_close_elements([1.0, 2.8, 3.0, 4.0, 5.0, 2.0], 0.3)
    True
    """
    numbers = sorted(numbers)
    return any(numbers[i + 1] - numbers[i] < threshold for i in range(len(numbers) - 1))
```

BLOOMZ:

```
def has_close_elements(numbers: List[float], threshold: float) -> bool:
    """ Check if in given list of numbers, are any two numbers closer to each other than given threshold.
    >>> has_close_elements([1.0, 2.0, 3.0], 0.5)
    False
    >>> has_close_elements([1.0, 2.8, 3.0, 4.0, 5.0, 2.0], 0.3)
    True
    """
    numbers = sorted(numbers)
    return any(numbers[i + 1] - numbers[i] < threshold for i in range(len(numbers) - 1))
```

BLOOMZ is biased towards short answers as most training examples are short

Workaround: Force minimum generation length

Training on xP3mt influences generation language 🇫🇷🇬🇧🇫🇷

BLOOMZ capable of creative writing following instructions, e.g. generating fables 🎭

Models still display some failure modes like not clearly following instructions 🚫

Input:
Proposez au moins cinq mots clés concernant «réseau de neurones artificiels».

BLOOMZ: artificial neural network, artificial intelligence, machine learning, deep learning, neural network

BLOOMZ-MT: réseau de neurones artificiels, réseau de neurones, apprentissage automatique, apprentissage en profondeur

mT0-13B: apprentissage, réseau, intelligence artificielle

Input:
Write a fairy tale about a troll saving a princess from a dangerous dragon. The fairy tale is a masterpiece that has achieved praise worldwide and its moral is "Heroes Come in All Shapes and Sizes". Story in Spanish:

BLOOMZ: Un reino en el bosque estaba bajo un terrible dragón. Todos los que intentaron derrotar al dragón murieron. El rey y la reina le pedían al príncipe que les diera un niño, que les guiaría salvando al reino de la mano del dragón. Un día, la reina dio a luz a un niño con una cabeza enorme y ojos que brillaban como estrellas. El rey y la reina lloraron al niño Elliot. Elliot crecería y se convertiría en un joven gigante. Él salvaría al pueblo.

mT0: En la noche de la primera tempestad, un gigante dragón, llamado Odin, golpeó a una princesa al dedo. Así que un troll, llamado Odin, fue a salvar a la princesa. Luego, este fue quemado por todos.

Input:
Why is the sky blue?

BLOOM: "Why is the sky blue?" "Why is the sky blue?" "Why is the sky blue?..."

BLOOMZ: Rayleigh scattering.

BLOOMZ with 10 tokens min length: Rayleigh scattering. The blue color of the sky is due to Rayleigh scattering of sunlight by atmospheric molecules.

mT0-13B: because of the sun

mT0-13B with 10 tokens min length: because of the sun's radiation sensitivity to light