Automated Text Annotation with AI

Matteo Nebbiai

PhD Candidate, Department of Political Economy, King’s College London

matteo.nebbiai@kcl.ac.uk

X/bsky: @MatteoNebbiai

1. What is (automated) text annotation?

Text annotation

- Assigning **labels** to **textual items** on the basis of a **coding scheme.**

- Examples:
  - determining if a political party manifesto is left-wing/right-wing on a 1-10 ideology scale (Young & Soroka 2012)
  - determining if a NYT News Article tone is positive/negative/neutral (Benoit et al. 2016)

Examples… from your research?

- How could text annotation be used in your research? Have you already used it?
  - types of **corpus** (manifestos, documents, social media posts, web pages…)
  - types of **feature** (ideology, tone, topic…)
  - (write your name!)

Click on https://www.menti.com/al6ixpdtya43

Or type the code **7695 4655** on www.menti.com

Text annotation with humans

1. Create a codebook with instructions
2. Recruit and train coders (i.e., research assistants or crowd-workers on platforms)
3. Coders annotate the corpus

Text annotation with classification model

1. Create a codebook with instructions
2. Use an already coded sample of items to train a classification model
3. Classification model annotates the corpus

Text annotation with Large Language Models

1. Create a codebook with instructions
2. Create a prompt explaining the task
3. Query the prompt and the corpus to be analysed to an LLM

LLMs allow for «zero-shot» classifications (without any additional training) (Gilardi 2023)

LLMs vs. humans

Pros

- higher accuracy and inter-coder agreement (Törnberg 2023)
- no need to train coders
- significantly lower monetary and time cost
  - 0.003$ (LLM) v. 0.10$ per annotation (Mturk) (Gilardi 2023)
  - few hours (LLM) v. 4 years (human) (Leek et al. 2024)

Cons

- lower replicability

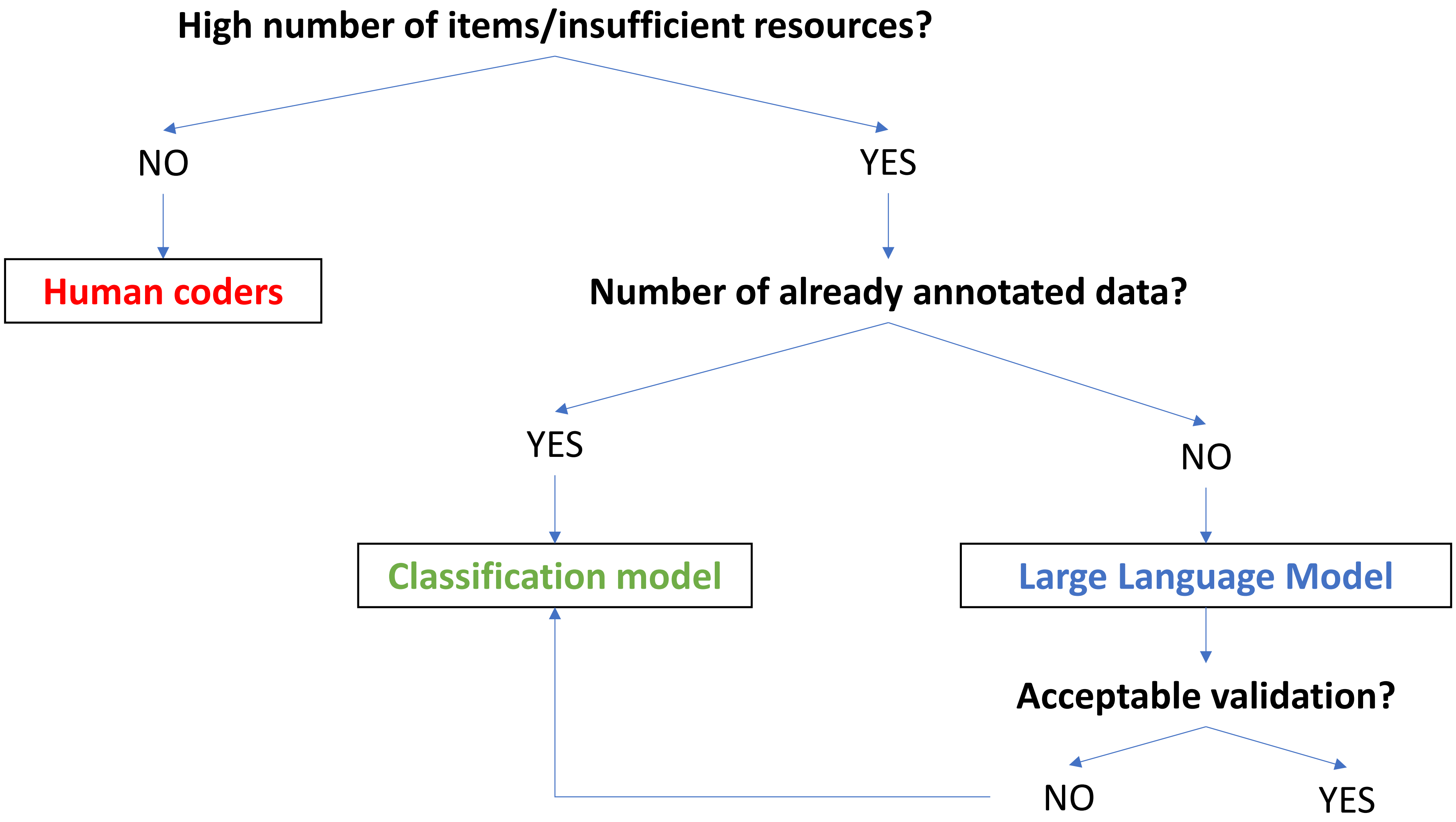
LLMs vs. classification models

Pros

- no need to train the model with pre-coded data

Cons

- lower replicability
- lower accuracy, especially in comparison with zero-shot & for complex tasks (Thalken et al. 2023, Plaza-del-arco et al. 2024)



Adapted from Weber and Reichardt (2023)

Ethical issues

- Copyright
- Job replacement
- Energy consumption

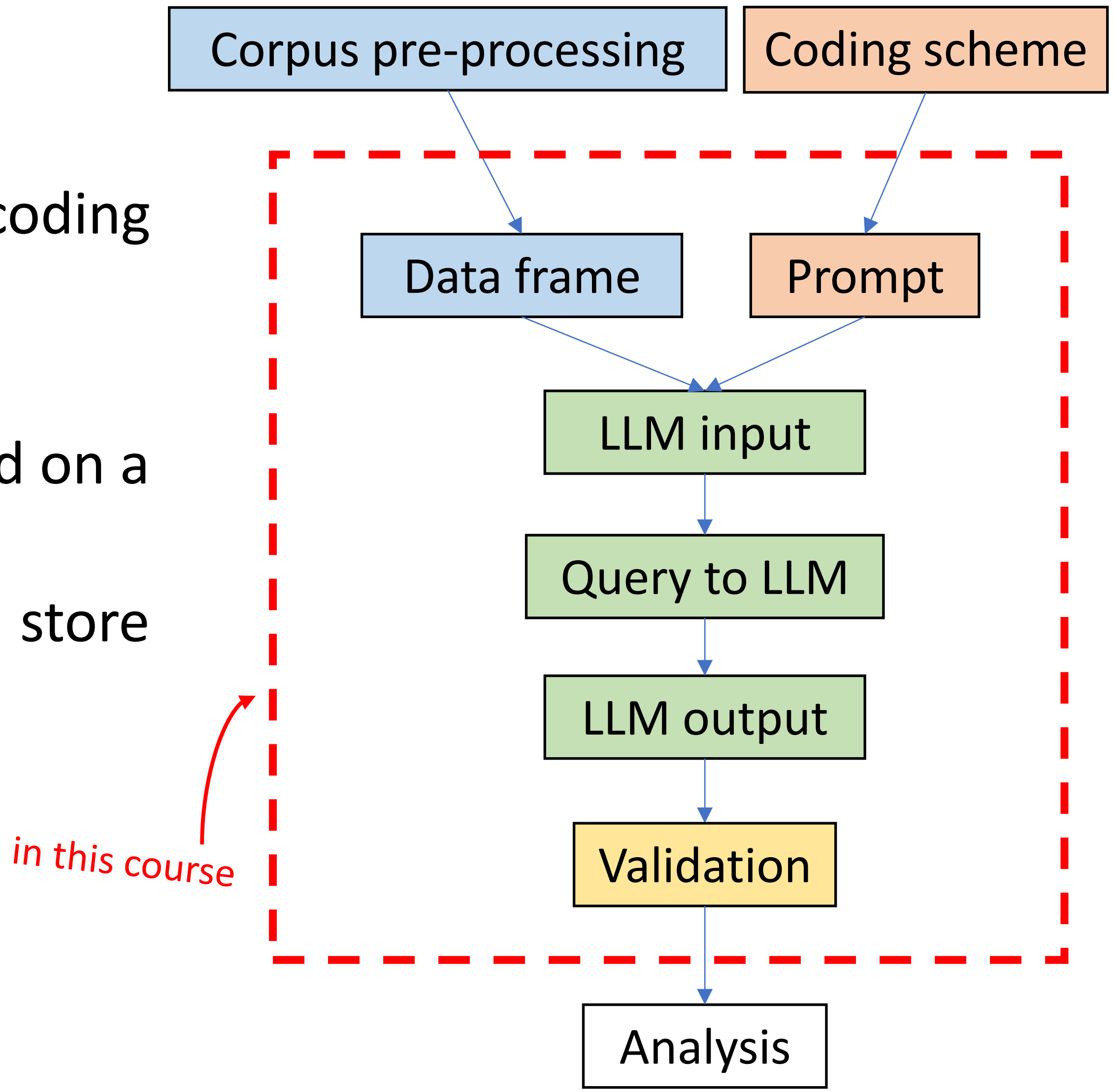
Questions?

2. Implementation

Workflow

1. Get a machine-readable corpus and a coding scheme*
2. Convert corpus into a data frame
3. Combine text items with a prompt based on a coding scheme, to obtain the LLM input
4. Query the input to the model's API, and store the LLM output
5. Validation
6. Analysis*

*not in this course



Corpus pre-processing

- delete text unnecessary for your analysis (i.e., sentences extraction)

→ **shorter text = lower cost-per-prompt**

- R Libraries:
  - Natural Language Processing: quanteda, tm
  - extracting text: pdftools, Rvest, Rselenium

R + tidyllms

Why **R**?

- Widely adopted
- Many statistics libraries
- Open source
- Free

Why **tidyllms**?

- Multiple models, single library
- Open source models (Gemini, Mistral)

break

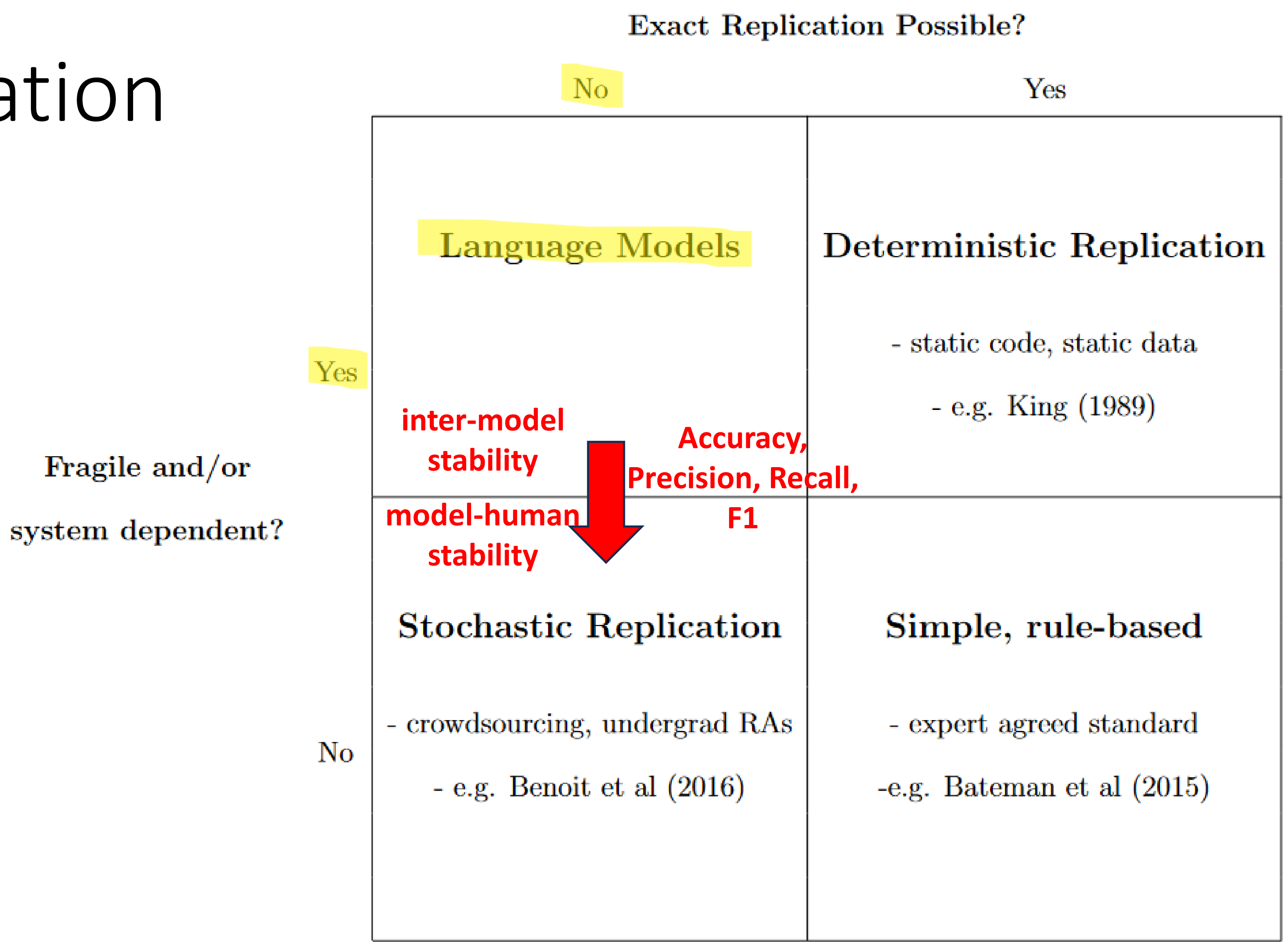
3. Making AI annotation replicable

Replication

Fragile and/or system dependent?	Exact Replication Possible? No	Exact Replication Possible? Yes
Yes	**Language Models**	**Deterministic Replication** - static code, static data - e.g. King (1989)
No	**Stochastic Replication** - crowdsourcing, undergrad RAs - e.g. Benoit et al (2016)	**Simple, rule-based** - expert agreed standard -e.g. Bateman et al (2015)

Barrie et al. forthcoming

Replication



Barrie et al. [forthcoming]()

Replication

Exact Replication Possible?

Fragile and/or system dependent?		No	Yes
	Yes	**Language Models** **inter-prompt stability** **intra-prompt stability** **intra-model stability**	**Deterministic Replication** - static code, static data - e.g. King (1989)
	No	**Stochastic Replication** - crowdsourcing, undergrad RAs - e.g. Benoit et al (2016)	**Simple, rule-based** - expert agreed standard -e.g. Bateman et al (2015)

Barrie et al. [forthcoming]

Accuracy, Precision, Recall

- Agreement with a benchmark of "True" annotations
  - Can be a rule-based classification (i.e. "Republican" vs. "Democrat")
  - Can be a classification assumed as "true" (i.e. compiled by experts)
- Used to compare different models/annotation techniques (classification models, humans) (Cova and Schmitz 2024)

Predicted class	True class	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Cova and Schmitz (2024)

Accuracy

$$\text{Accuracy} = \frac{TN + TP}{TN + FP + TP + FN} = \frac{\text{Total correct predictions}}{\text{Total predictions}}$$

- compare the number of agreements between true codification and model prediction with total number of predictions
- poor job with unbalanced data (data unevenly distributed across categories)

Precision

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{\text{correctly classified actual positive}}{\text{everything classified as positive}}$$

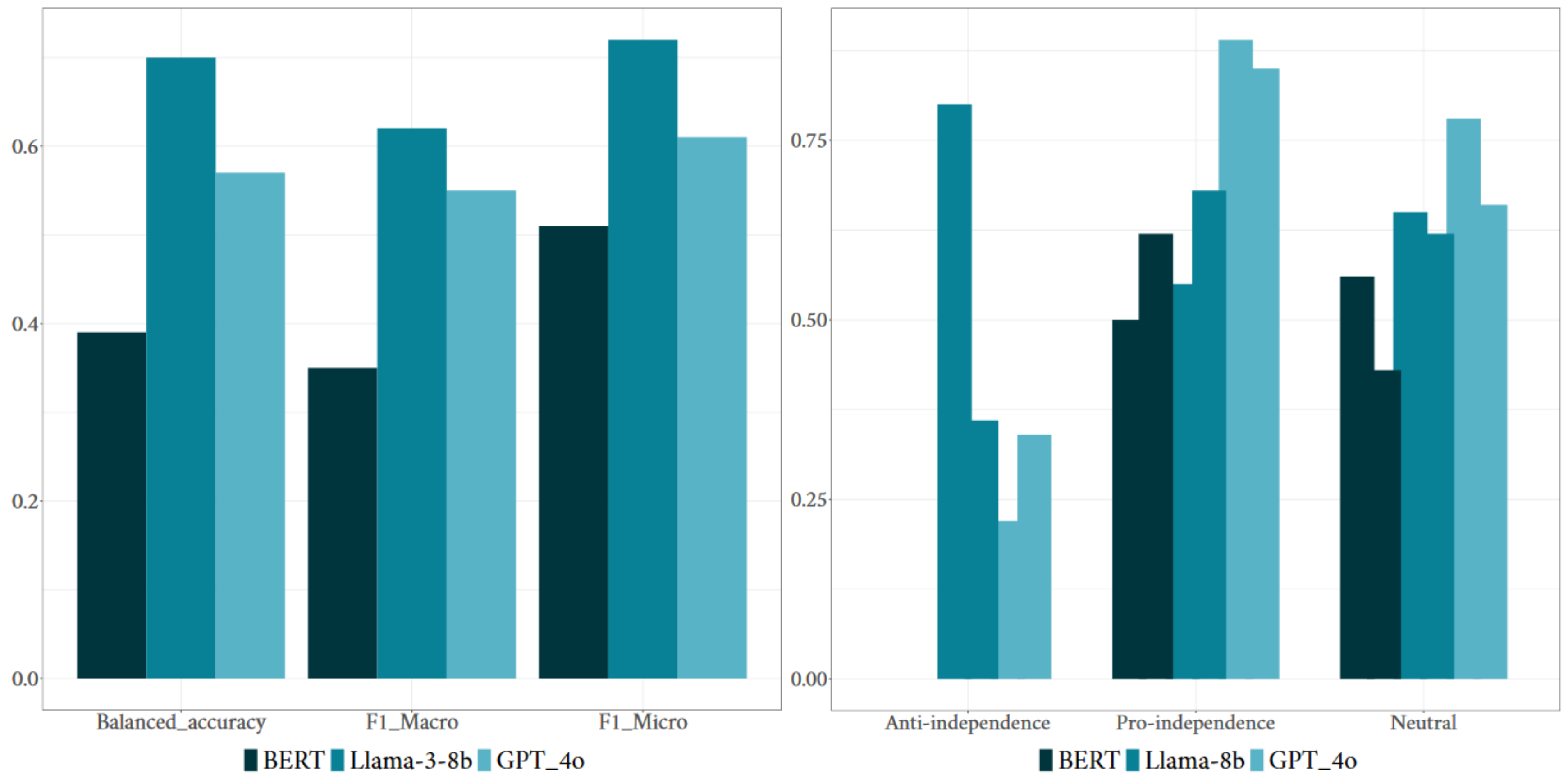
Recall

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{correctly classified actual positive}}{\text{all actual positives}}$$

F1 Score

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

- Harmonic mean of **precision** and **recall**, balancing the two
- Widely used to benchmark the performance (i) among models; (ii) between different annotation techniques; (iii) on different subsets of data



(a) *Model-level statistics* (b) *Precision and F1 per model*

(Cova and Schmitz 2024)

Inter-coder reliability

- Agreement between (somehow imperfect) independent coders classifying the same text

- Measures (Lombard et al. 2002):
  - Krippendorff-alpha, Cohen's kappa, Scott's pi
- Common thresholds for intercoder reliability (Lombard et al. 2002):
  - Conservative: >0.9/0.8
  - Less conservative: > 0,7

Measures of inter-coder reliability

Type	Output stability between runs with…	Literature
inter-prompt stability	semantically similar prompts	Barrie et al. 2024
intra-prompt stability	multiple runs of the same prompt (short span of time)	Barrie et al. 2024
intra-model stability	multiple runs of the same prompt (long span of time)	Barrie et al. forthcoming
inter-model stability	multiple runs of the same prompt, on different models	Barrie et al. forthcoming
model-human stability	human and LLM annotations	Gilardi 2023

Model tuning

- **System message**: the «role» assigned to the model, i.e.: "You are a skilled research assistant who will help to classify newspaper headlines."
- **Temperature**: randomness of the output.
- **Nucleus sampling** (top_p): sample of words considered by the model according to their probability.

Lower values: more deterministic output

- <u>No clear good practice at the moment</u>
  - Benchmarking on the basis of **accuracy, precision, recall and F1 scores**
  - Cova and Schmitz 2024 use the role to better specify the coding rules
  - Barrie et al. 2024 and Törnberg 2023 suggest a **low temperature** (i.e., 0.1)

Model choice

- Open models > closed models, because:
  - Closed models may cease to be runnable + privacy concerns, while open models can run offline/through a third party
  - Higher transparency* (i.e. updates, weights)
- Consider price & performance
  - HuggingFace LLM leaderboard
  - Free/trial APIs: GitHub

*debate on how open models might be not open enough

Prompting beyond zero-shot

- To improve accuracy: few-shot /fine-tuning LLMs (Alizadeh et al. 2024, Bucher & Martini 2024), Generated-knowledge prompting (Liu et al. 2022)
- For complex tasks: Chain-of-thought (Wei et al. 2022), Tree-of-thought (Long 2023)
- Etc. (Weber and Reichardt 2023)

Final recommendations

- Language matters: prevalence of English-centric models, impact on performance (Kuzman et al. 2023)
- Rapidly evolving landscape: keep updated on applications and academic standards
- Be tranparent and motivate every step (no limits in the Appendix!)
- Consider running a local LLM (no fees, but requires computational power)

Questions?