CO*IR: A Greedy and Individually Fair Re-ranker

Seán Healy
ADAPT Centre
<sean.healy@adaptcentre.ie>

Terminology

Cóir: the Gaelic term for justice or equity

Terminology

Cóir: the Gaelic term for justice or equity

CO*IR: Fairness (*in IR/Information retrieval)

Terminology

Cóir: the Gaelic term for justice or equity

CO*IR: Fairness (*in IR/Information retrieval)

- A nod to background work: **FA*IR**: Fair Top-k Ranking (Zehlike et al., 2017)

What’s the point of this work?

- **Fairness has an adoption issue in industry.**

What’s the point of this work?

- Fairness has an adoption issue in industry.
  - Muted response to FairSearch tool from Zehlike et al. (2020).

What’s the point of this work?

- Fairness has an adoption issue in industry.
  - Muted response to FairSearch tool from Zehlike et al. (2020).
- Clear need for “bottom-up” approach to fairness.

What’s the point of this work?

- Fairness has an adoption issue in industry.
  - Muted response to FairSearch tool from Zehlike et al. (2020).
- Clear need for a “bottom-up” approach to fairness.
  - Assume low resources.

What's the point of this work?

- Fairness has an adoption issue in industry.
  - Muted response to FairSearch tool from Zehlike et al. (2020).
- Clear need for a "bottom-up" approach to fairness.
  - Assume low resources.
    - Low computational resource.
    - Low human resources.
    - Low data resources.

What’s the point of this work?

- Low computational resources.
- Low human resources.
- Low data resources.

What’s the point of this work?

- ➔ Low computational resources.
- ➔ Low human resources.
- ➔ Low data resources.

Does your solution work on a cheap server, or is a powerful GPU needed?

Does your solution require special expertise to deploy, or can it be done by an admin?

Do you assume known group labels?

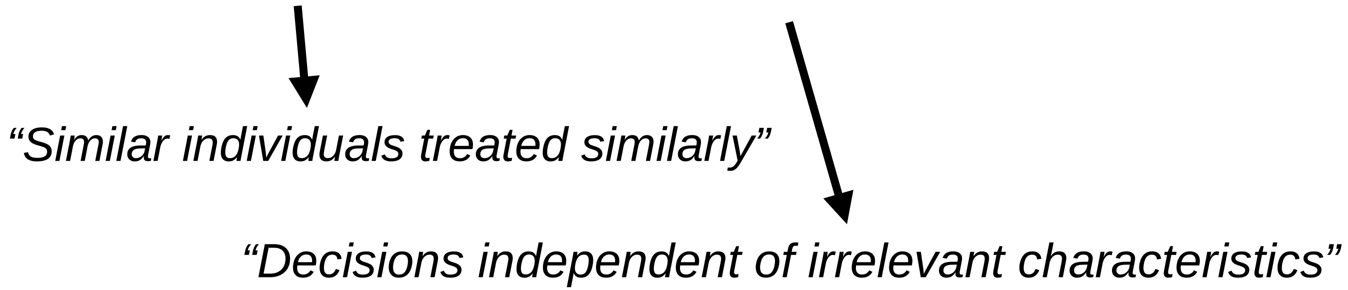
Also. . .

Fairness needs to work on an open web

> “ Information systems start small and grow. They also start isolated and then merge. A new system must allow existing systems to be linked together without requiring any central control or coordination." (Berners-Lee, 1989) ”

Background Work

- Laying the ground work: <u>Individual fairness</u> vs. <u>Group fairness</u> (Dwork et al., 2012)



“Similar individuals treated similarly”

“Decisions independent of irrelevant characteristics”

Background Work

- Laying the ground work: Individual fairness vs. Group fairness (Dwork et al., 2012)

“Similar individuals treated similarly”

“Decisions independent of irrelevant characteristics”

- Amortized fairness (Biega, Gummandi, Weikum, 2018) (Singh and Joachims, 2018)

“It is impossible to grant everyone the attention they deserve in a single ranking.” (Biega, 2018)

Individually fair rankings $\sqsubseteq$ *Group-fair rankings*

Background Work (Caveats)

- Meritocratic vs. Egalitarian notions of fairness (Zehlike, Yang and Stoyanovich, 2021)
  - WYSIWYG (*What you see is what you get*) worldview

    vs.
  - WAE (*We are all equal*) worldview.
  - Construct space vs. observed space.

Let’s rewind…

Does your solution work on a cheap server, or is a powerful GPU needed?

- Low computational resource.
- Low human resources.
- Low data resources.

Does your solution require special expertise to deploy, or can it be done by an admin?

Do you assume known group labels?

Proposed solutions

- Low computational resource.

Use a “greedy” method (sacrifice some optimality for speed)

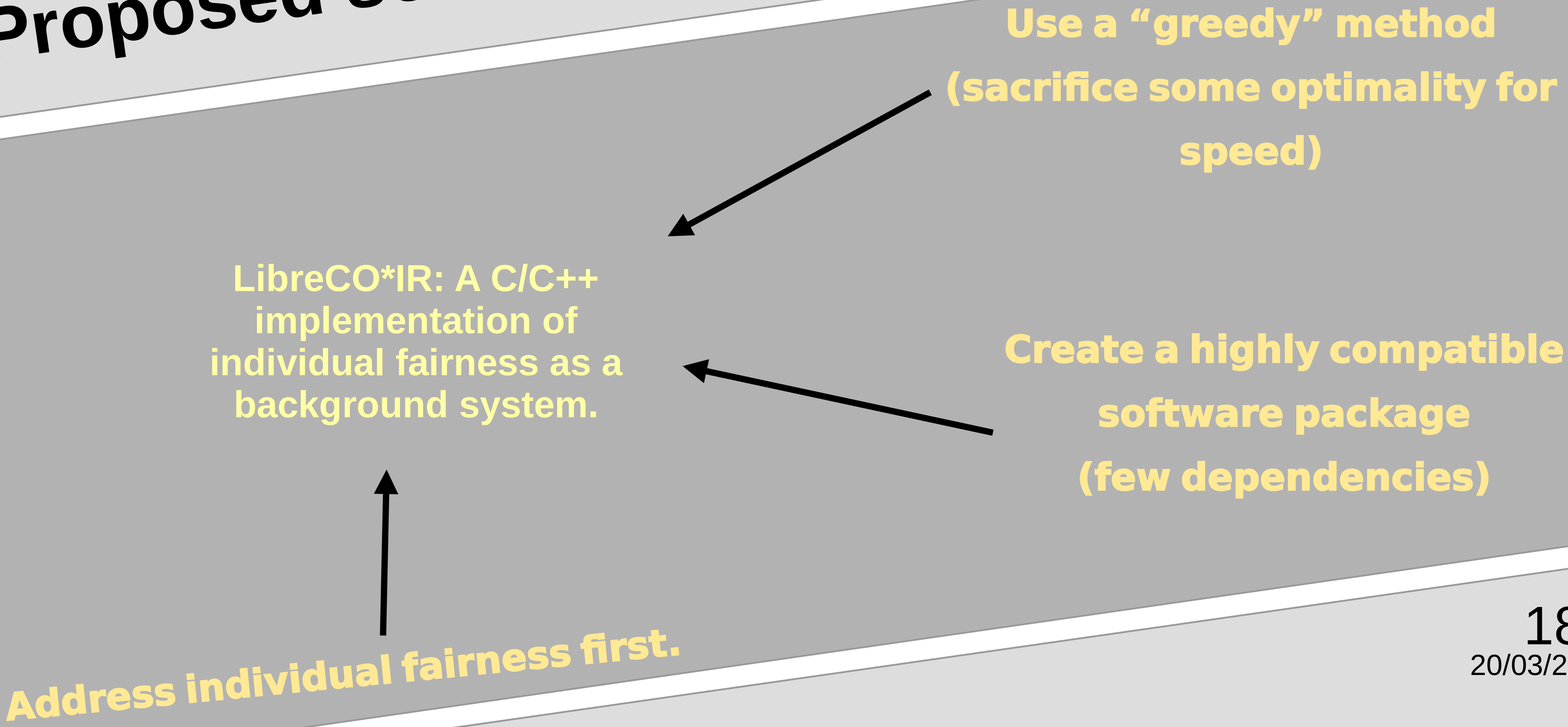
- Low human resources.

Create a highly compatible software package (few dependencies)

- Low data resources.

Address individual fairness first.

Proposed solutions



Use a “greedy” method
(sacrifice some optimality for
speed)

LibreCO*IR: A C/C++ implementation of individual fairness as a background system.

Create a highly compatible
software package
(few dependencies)

Address individual fairness first.

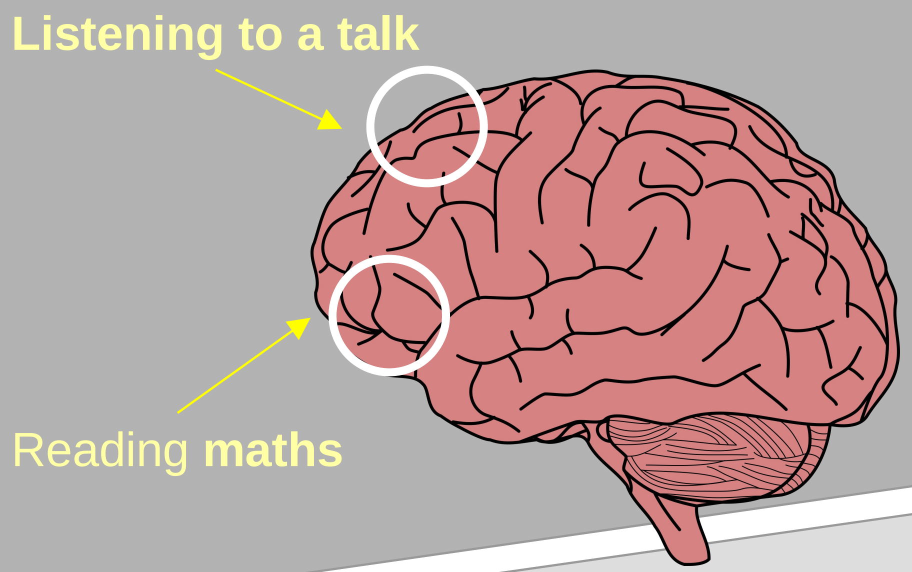
The greedy approach…

The greedy approach…

Are you ready for some maths?

The greedy approach…

Are you ready for some maths?



… I never am.

The greedy approach…

So let’s skip the technical stuff

… this is how it works:

The greedy approach…

So let’s skip the technical stuff

… this is how it works:

- Rather than re-ranking search results en-masse…

The greedy approach…

So let’s skip the technical stuff

… this is how it works:

- ➔ Rather than re-ranking search results en-masse…
- ➔ Generate one (re-ranked) result at a time.

The greedy approach…

So let’s skip the technical stuff

… this is how it works:

- Rather than re-ranking search results en-masse…
- Generate one (re-ranked) result at a time.
- This is much faster.

(albeit ‘sub-optimal’)

The greedy approach…

So let’s skip the technical stuff

(technical stuff in the white paper.)

Qualitative Research

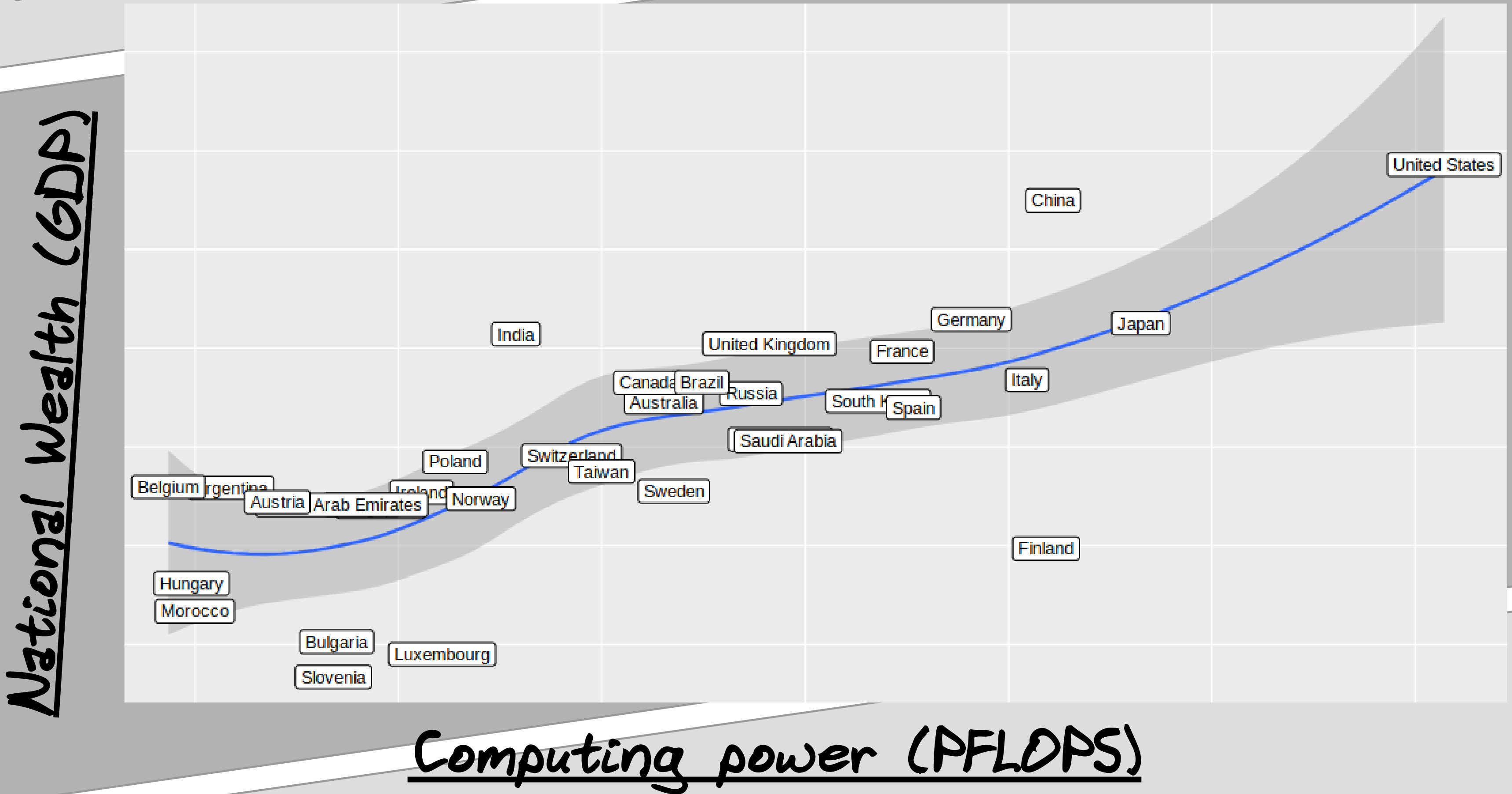
Examining GitHub “most engaged” software packages in data processing and information management (up to the top 300 “most popular”)

Qualitative Research

Examining GitHub “most engaged” software packages in data processing and information management (up to the top 300 “most popular”)

- The majority are written in native languages (C, Rust, Go).
- The majority have low external dependencies.
- Case studies: Redis, duckdb, sqlite, git

Qualitative Research



Insights from Software Eng.

- Ranked lists of people and groups are highly common in human resource systems (not just in search).
- Implementation sprints are often short (2 weeks).
- Those in power often reject fairness considerations.

Insights from Software Eng.

- Ranked lists of people and groups are highly common in human resource systems (not just in search).
- Implementation sprints are often short (2 weeks).
- Those in power often reject fairness considerations.

  The profit motive makes it difficult to obtain time and resources for fairness considerations.

Insights from Software Eng.

- Practically all operating systems have a C/C++ compiler.
- Almost all of the internet (big and small) runs on Linux.
- LibreCO*IR is A C/C++ program designed primarily for Linux, targetting wider adoption of fairness practices within industry.

Insights from Software Eng.

Changing the conversation via working software…

Common gripes:

- “Fairness might be a performance bottleneck.”
- “Fairness assumes surveillance capitalism (group labels)”
- “We don’t have the expertise to deploy it.”
- “The code worked well 4 years ago, but the dependencies are broken now.”

Insights from Software Eng.

Changing the conversation via working software…

Common gripes:

- ~~“Fairness might be a performance bottleneck.”~~
- ~~“Fairness assumes surveillance capitalism (group labels)”~~
- ~~“We don’t have the expertise to deploy it.”~~
- ~~“The code worked well 4 years ago, but the dependencies are broken now.”~~

Links

https://librecoir.com/

sean.healy@adaptcentre.ie