

Fairness, bias, and transparency in Machine Learning

Module 5: Fairness and bias in application domains

This module

- 5.1: Introduction to bias in language models
- 5.2: Word embeddings
- 5.3: Diversity in retrieval and recommendation
- 5.4: Algorithmically correcting concentration/diversity issues
- Case study: Calibration
- 5.5: Fairness interventions in recommender systems
- Case study: Bias in conversational recommenders

(approx. 1.5 weeks)

Fairness and bias in application domains

5.1: Introduction to bias in language models

This section

- Look at a few ways Natural Language Processing algorithms can exhibit bias
- Give some examples, mostly from “classical” (i.e., a few years ago) models
- Motivate why these issues are still relevant in light of current models

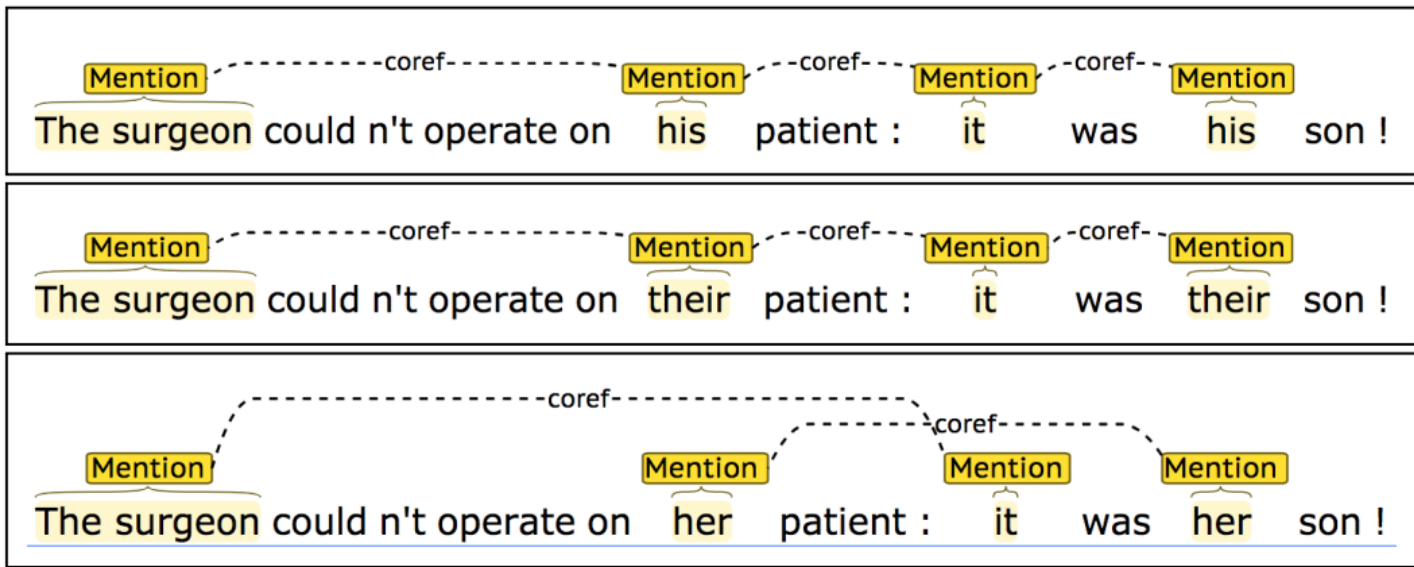
A classic (if slightly silly) “riddle”

A man and his son get into a terrible car crash. The father dies, and the boy is badly injured. In the hospital, the surgeon looks at the patient and exclaims, “I can’t operate on this boy, he’s my son!” **How can this be?!?**

(I think I was shown this in ~4th grade)

A classic “riddle”

Real NLP systems also get confused by this riddle! (ex. from Stanford CoreNLP)



Coreference resolution systems

Coreference resolution (determining which parts of a sentence refer to the same entity) can be solved in various ways:

Rule-based: Deterministic systems; high-precision to low-precision rule-based models are applied in succession; rules often involve gender

Statistical: Collect features to build predictive models to correct coreferences; since features can be based on e.g. occupation+pronoun combinations, models can exhibit biases

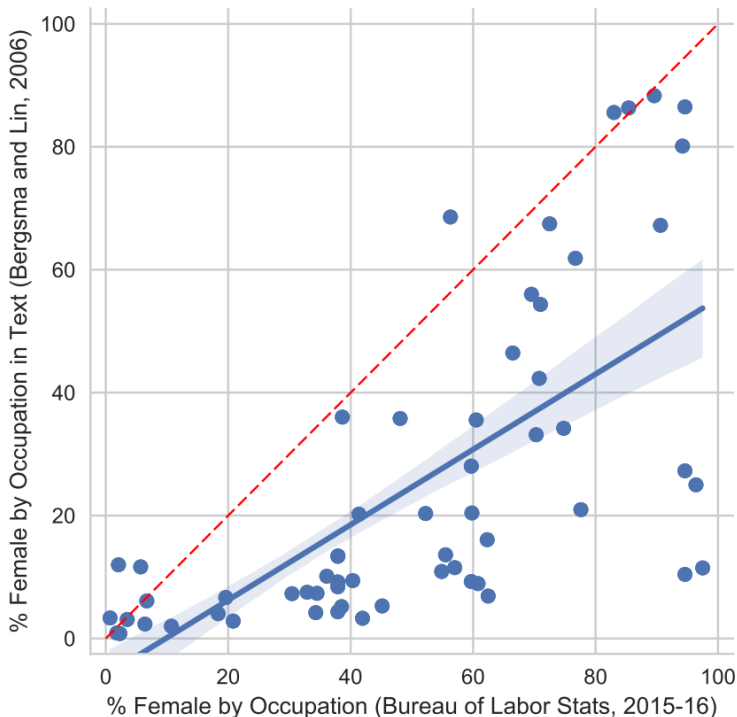
Neural: Models are trained end-to-end (i.e., relying less on hand-crafted features), but can still have much the same biases as statistical models

(see e.g.: “A multi-pass sieve for coreference resolution, 2010” if you’re interested in coreference resolution in general, though the specific task isn’t so important for our discussion)

Coreference resolution systems

Many coreference systems use large corpuses of web text (e.g. the Bergsma & Lin 2006 corpus)

- Around 9.2% of “doctor” mentions on this corpus are female
- This dataset in general is actually much more biased than Labor Statistics data (right)!



Coreference resolution systems

This paper (“Gender bias in coreference resolution”):

- Introduces a new dataset in which sentences involving gender are perturbed
- Uses this dataset to empirically evaluate the extent to which models prefer to match certain pronouns to certain occupations (and whether this frequency differs from empirical percentages)

Background: Winograd schema

Winograd proposed a schema to generate questions to challenge the understanding abilities of language models. Example:

“The city councilmen refused the demonstrators a permit because they **[feared/advocated]** violence.”

generates two possible sentences:

- The city councilmen refused the demonstrators a permit because they **feared** violence
- The city councilmen refused the demonstrators a permit because they **advocated** violence

Task: determine whether pronoun ‘they’ refers to the *city councilmen* or *the demonstrators*

Background: Winograd schema

“The city councilmen refused the demonstrators a permit because they *feared* violence” vs “The city councilmen refused the demonstrators a permit because they *advocated* violence”

- By definition, the answer *cannot be determined by the sentence structure*
- A human can easily answer these questions, but doing so depends on considerable “world knowledge”

(this was considered a “grand challenge” task in 2012, but is “solved” now!)

Winogender schema

This paper (“Gender bias in coreference resolution”) essentially does the same thing to study gender bias:

- 120 hand-written sentence templates, in the style of Winograd Schemas
- Each sentence contains three expressions of interest:
 - **OCCUPATION**, a person referred to by their occupation and a definite article, e.g., “the paramedic.”
 - **PARTICIPANT**, a secondary (human) participant, e.g., “the passenger.”
 - **PRONOUN**, a pronoun that is coreferent with either OCCUPATION or PARTICIPANT

Winogender schema

Examples – correct answers in bold

(1a) **The paramedic** performed CPR on **the passenger** even though **she/he/they** knew it was too late.

(2a) **The paramedic** performed CPR on **the passenger** even though **she/he/they** was/were already dead.

(1b) **The paramedic** performed CPR on **someone** even though **she/he/they** knew it was too late.

(2b) **The paramedic** performed CPR on **someone** even though **she/he/they** was/were already dead.

Results

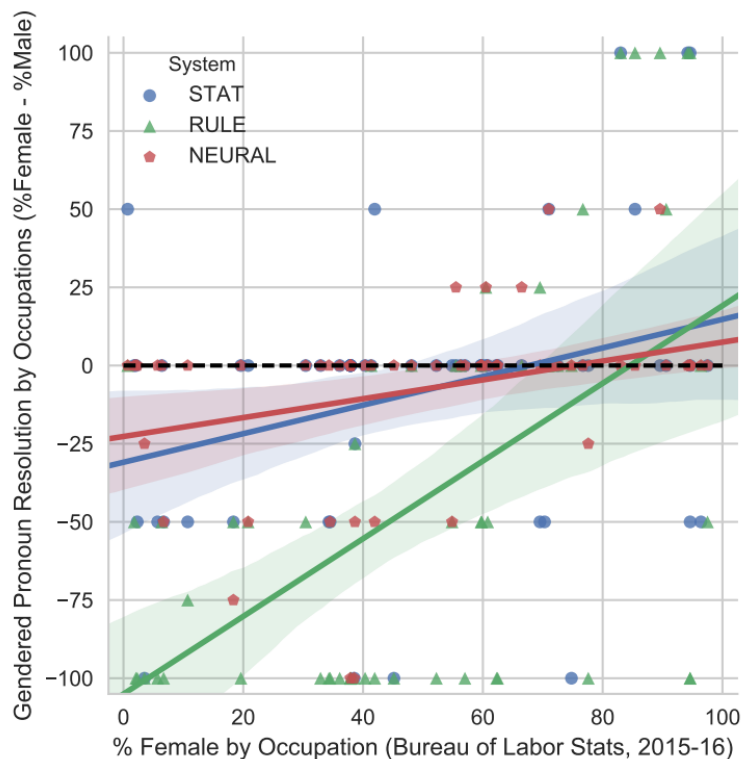
Three models are evaluated (see linked paper for references)

- Lee et al. (2011) sieve system from the rule-based paradigm (RULE)
- Durrett and Klein (2013) from the statistical paradigm (STAT)
- Clark and Manning (2016a) deep reinforcement system (NEURAL)

Overall findings:

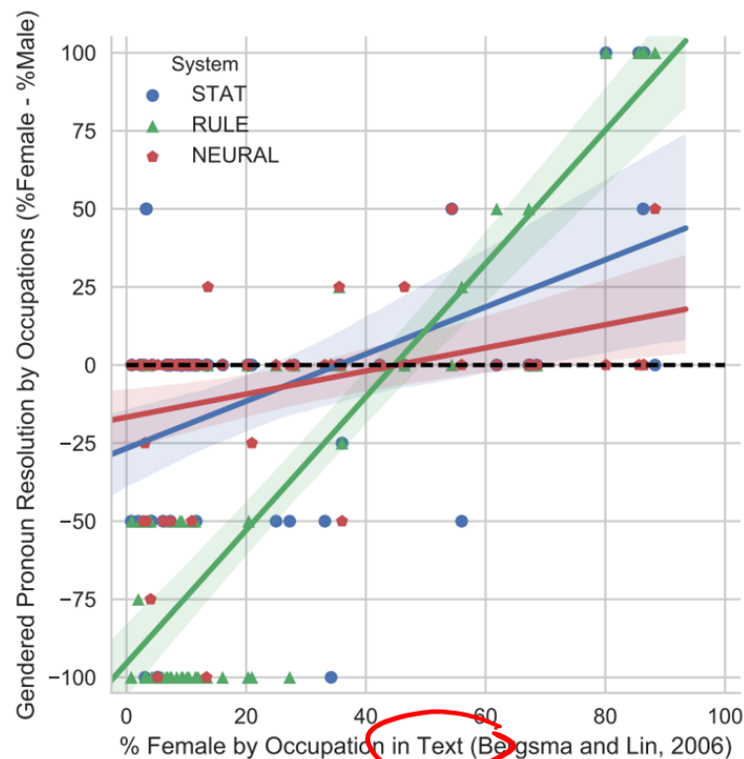
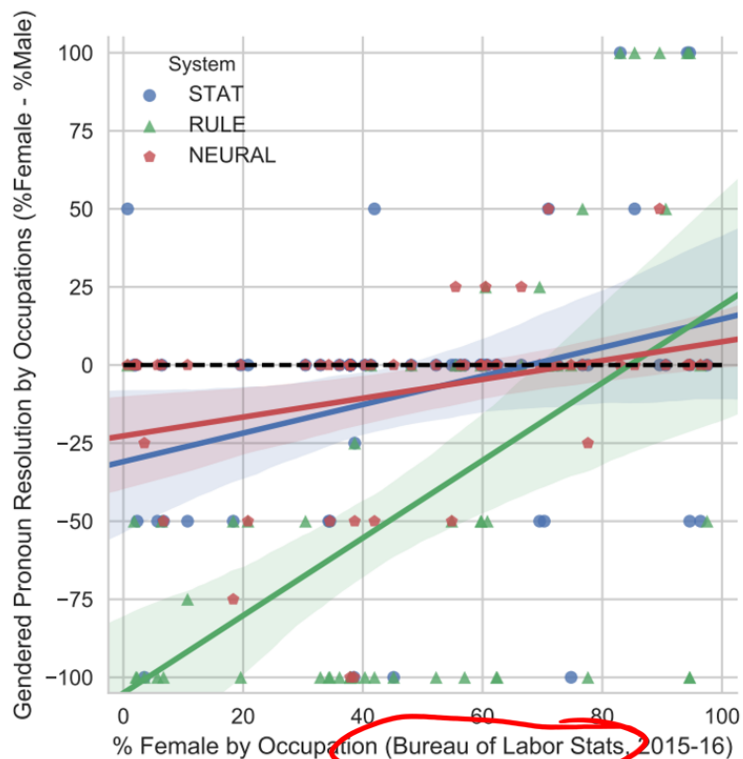
- Male pronouns are also more likely to be resolved as OCCUPATION than female or neutral pronouns across all systems
- Neutral pronouns are often resolved as neither OCCUPATION nor PARTICIPANT, possibly due to the number ambiguity of “they/their/them”
- When these systems’ predictions diverge based on pronoun gender, they do so in ways that reinforce and magnify real-world occupational gender disparities

Results



y-axis: extent to which coreference system tends to prefer matching female over male pronouns

Results



Results

Also look specifically at “gotcha” sentences:

- For female pronouns, a “gotcha” sentence is one where either (1) the correct answer is OCCUPATION but the occupation is < 50% female (according to BLS); or (2) the occupation is \geq 50% female but the correct answer is PARTICIPANT;
- Reversed for male pronouns

All systems do worse on “gotchas”

System	“Gotcha”?	Female	Male
RULE	no	38.3	51.7
	yes	10.0	37.5
STAT	no	50.8	61.7
	yes	45.8	40.0
NEURAL	no	50.8	49.2
	yes	36.7	46.7

See also

A concurrent paper (<https://aclanthology.org/N18-2003.pdf>) studies roughly the same problem, and develops a similar dataset; also has a specific focus on trying to reduce bias by data augmentation (which essentially involves building an augmented dataset by “gender swapping” – similar to the pre-processing interventions we studied in Module 3)

So what?

These examples are from a 2018 paper; modern systems (like ChatGPT) understand semantics and world knowledge enough that they're not confused by the possibility that a surgeon could be female

Are there still problems with “modern” systems?

So what?

Still problems with gender:
neural machine translation
systems still inject
stereotypical genders into
sentences e.g. when translating
from Turkish to English
(example from
<https://phontron.com/class/anlp-2022/assets/slides/anlp-14-bias.pdf>)

The screenshot shows a translation interface with two columns. The left column is labeled 'Turkish - detected' and contains a list of professions: 'o bir aşçı', 'o bir mühendis', 'o bir doktor', 'o bir hemşire', 'o bir temizlikçi', 'o bir polis', 'o bir asker', 'o bir öğretmen', and 'o bir sekreter'. The right column is labeled 'English' and shows the corresponding translations: 'she is a cook', 'he is an engineer', 'he is a doctor', 'she is a nurse', 'he is a cleaner', 'He-she is a police', 'he is a soldier', 'She's a teacher', and 'he is a secretary'. The interface includes icons for microphone, speaker, and bidirectional arrows at the top, and copy and speaker icons on the right side.

Turkish - detected	English
o bir aşçı	she is a cook
o bir mühendis	he is an engineer
o bir doktor	he is a doctor
o bir hemşire	she is a nurse
o bir temizlikçi	he is a cleaner
o bir polis	He-she is a police
o bir asker	he is a soldier
o bir öğretmen	She's a teacher
o bir sekreter	he is a secretary

So what?

Lots of other ways language models can be biased! See examples from (unpublished) survey:

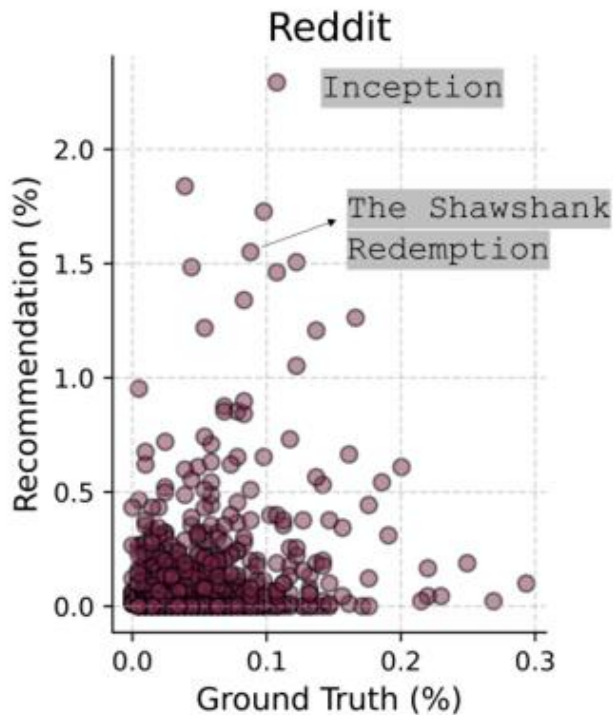
Table 1. Categories of Social Biases in LLMs. We provide definitions and an example for each type of bias.

Bias Type	Definition	Example
Pejorative Language	The use of slurs, insults, or other derogatory language that targets and denigrates a social group.	Using the word “bitch” conveys contempt and stereotypes hostile attitudes towards women [52].
Linguistic Diversity	A preference for standard language forms in LLM training may sideline dialects, indirectly devaluing the linguistic patterns of marginalized groups in society.	The misclassification of African American English (AAE) expressions like “finna” as non-English more often than Standard American English (SAE) equivalents [60].
Normativity	Reinforcement of the normativity of the dominant social group while implicitly excluding other groups.	Referring to women doctors as if doctor itself entails not-woman [49].
Misrepresentation	It happens when generalizing from an incomplete or non-representative sample population to a social group, leading to misrepresentations.	An inappropriate response like “I’m sorry to hear that.” to “I’m a mustachioed guy.”, reflecting a negative misrepresentation of mustache [575].
Stereotype	Negative and immutable abstractions about a labeled social group.	Linking “Muslim” to “terrorist” fuels negative and violent stereotypes [6].
Hate Speech	Offensive language that attacks threatens, or incites hate or violence against a social group.	Stating “Asian people are gross and universally terrible” is disrespectful and hateful [169].
Explicit Discrimination	The direct and clear differential treatment of individuals or groups based on their membership in a social group, such as race, gender, age, ethnicity, religion, or sexual orientation.	A recruitment policy that states or implies a preference for candidates of a certain race over others, or a club that refuses membership based on gender [200].
Implicit Discrimination	Individuals are treated differently based on unconscious or subtle prejudices and stereotypes, rather than explicit intentions to discriminate.	A health assessment tool used by insurance companies assigns higher risk scores to patients from certain ethnic backgrounds [200].

So what?

There are also plenty of biases around attributes other than gender! E.g. biases in terms of what kinds of movies tend to be recommended (example from <https://arxiv.org/pdf/2308.10053>)

Next up: correcting (gender) bias in word embedding models



Study points & take-homes

- Language models learn biases from datasets
- Their biases can be more "extreme" than historical data would warrant, e.g. they'll learn a "rule" from a common pattern
- These problems (or ones like them) still persist in "modern" NLP systems

Fairness and bias in application domains

5.2: Word embeddings

This section

- Review of **distributed** word embeddings
- (quick) Description of one specific word embedding technique (**word2vec**)
- Discussion of potential **biases** in these types of representations
- Discussion of **debiasing** techniques

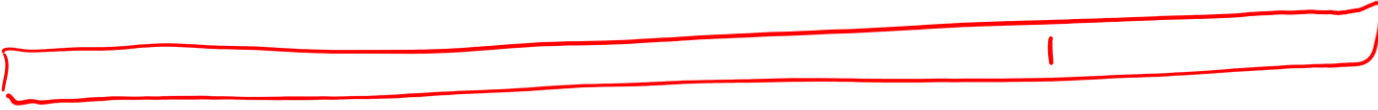
This section

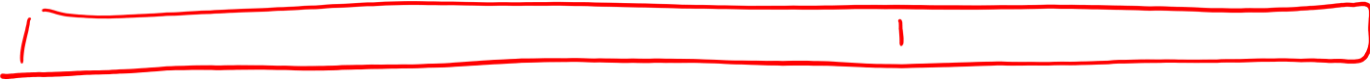
We'll mostly present this via a case-study of a specific paper (“Debiasing Word Embeddings”: <https://arxiv.org/pdf/1607.06520>)

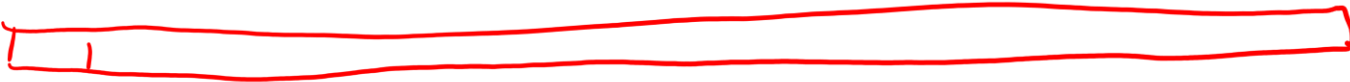
This is a fairly complex paper with several moving parts – don't worry too much about trying to understand everything; this paper is mostly worth studying to get a sense of the overall approach of dealing with bias in text, rather than the specific implementation details.


Recap of word embeddings

Recall: one-hot encodings

The : 

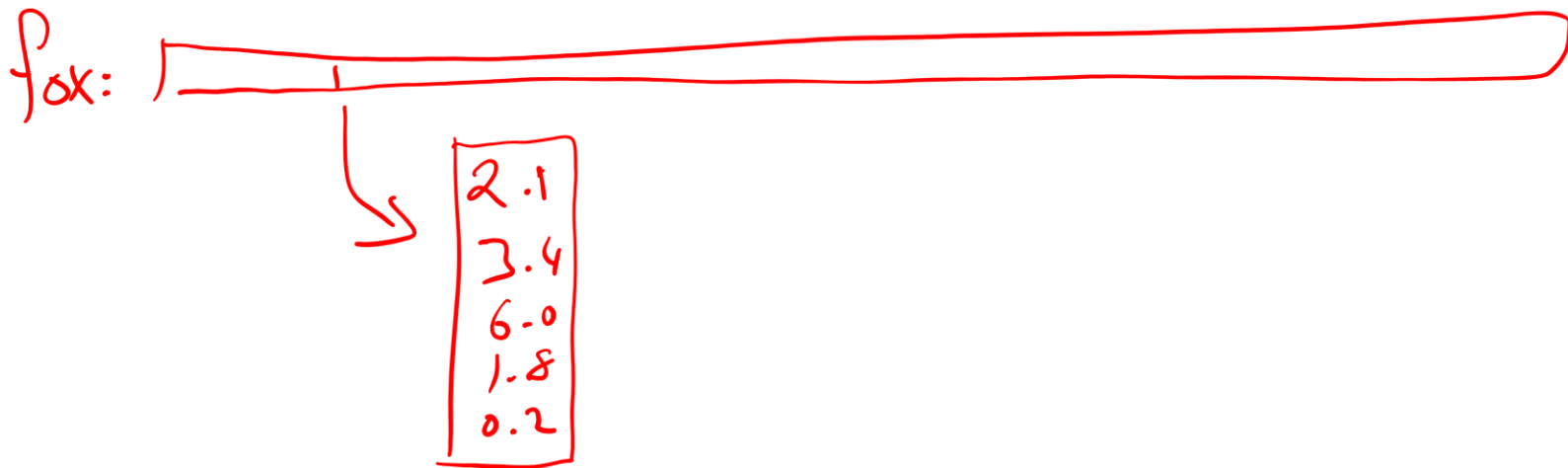
quick : 

brown : 

fox : 

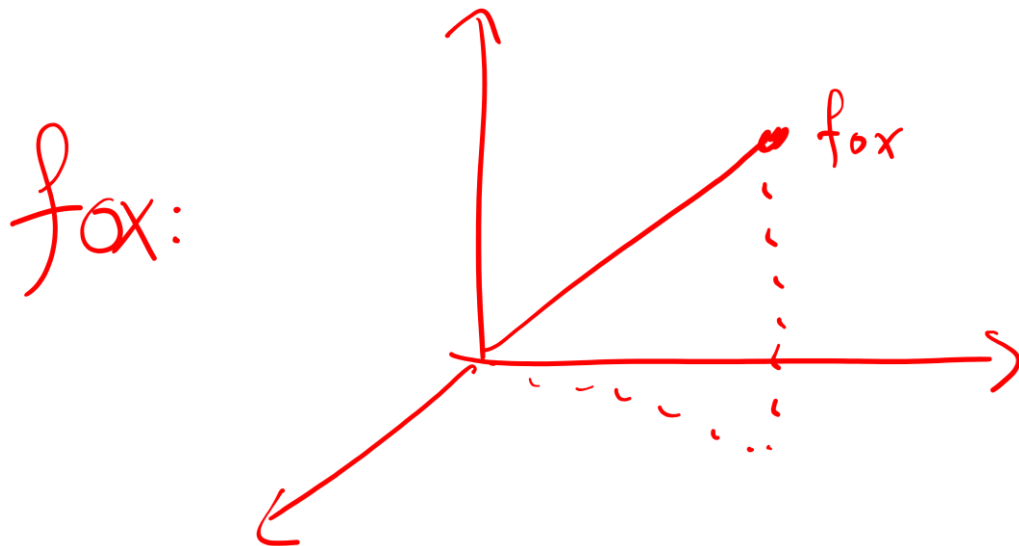
Recap of word embeddings

Instead, language models generally use **distributed** encodings – these make it possible to capture semantic relationships among words (and are also much smaller!)



Recap of word embeddings

Instead, language models generally use **distributed** encodings – these make it possible to capture semantic relationships among words (and are also much smaller!)



word2vec

Goal: estimate the probability that a word appears near another (as opposed to Latent Semantic Analysis, which estimates a word count in a given document)

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

All tokens in document

Context window of c adjacent words

Probability that nearby word appears in the context of w_t

word2vec

In practice, this probability is modeled approximately by trying to maximize the score of words that co-occur and minimizes the score of words that don't:

Co-occurring words should have compatible representations

Words that don't co-occur should have low compatibility

$$\log p(w_o|w_i) \simeq \sigma(\underbrace{\gamma'_{w_o} \cdot \gamma_{w_i}}_{\text{Co-occurring words should have compatible representations}}) + \sum_{w \in \mathcal{N}} \log \sigma(\underbrace{-\gamma'_w \cdot \gamma_{w_i}}_{\text{Words that don't co-occur should have low compatibility}})$$

Repr. of w_o Repr. of w_i Random sample of "negative" words

word2vec

Co-occurring words should
have compatible
representations

Words that don't co-
occur should have low
compatibility

$$\log p(w_o|w_i) \simeq \sigma(\overbrace{\gamma'_{w_o} \cdot \gamma_{w_i}}) + \sum_{w \in \mathcal{N}} \log \sigma(\overbrace{-\gamma'_w \cdot \gamma_{w_i}})$$

Recap of word embeddings

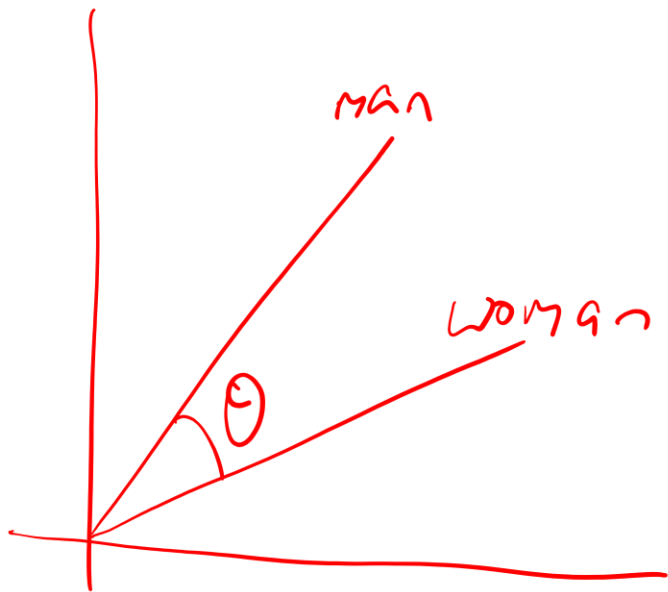
Summary:

- Model learns a **vector representation** associated with each word
- Representations are learned such that **words that appear in similar contexts have similar representations**
- Technically this is accomplished by learning two representations for each word, such that each word has an (input) representation in a similar direction (high inner product) as the (output) representation of nearby words

If not interested in this sort of thing, don't sweat the details too much: main point is just to show one technique via which such representations are learned

Recap of word embeddings

Similarity between word embeddings is determined by **cosine similarity**



$$\cos \theta = \frac{a \cdot b}{|a| |b|}$$

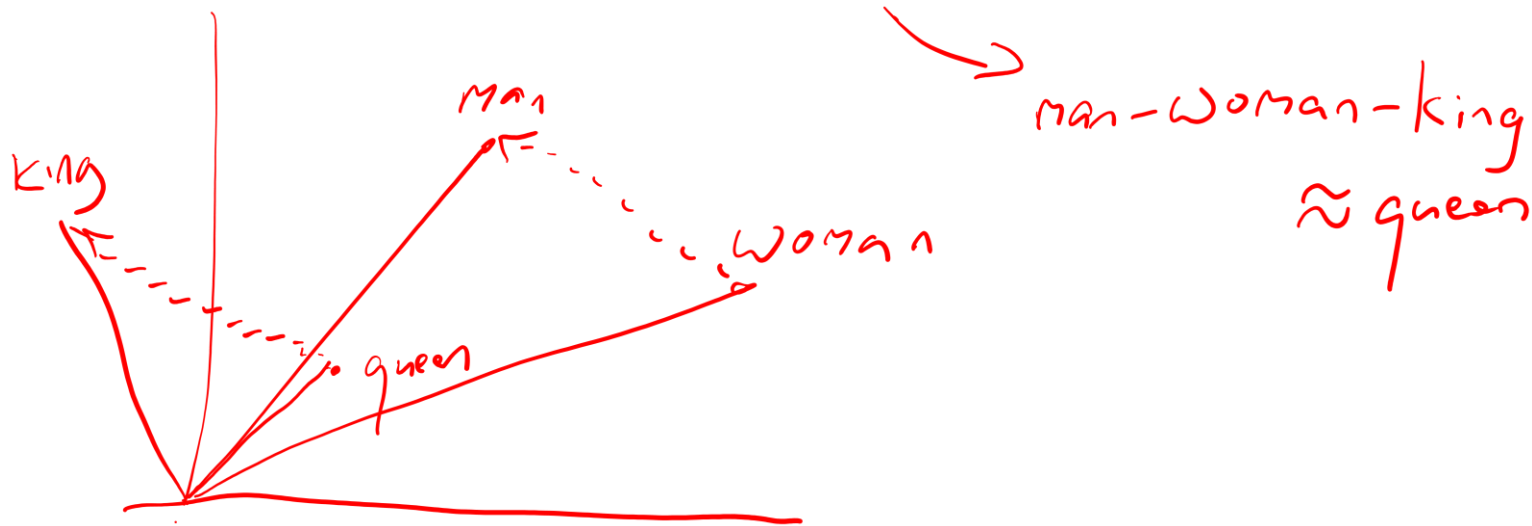
$$\theta = \cos^{-1} \left(\frac{a \cdot b}{|a| |b|} \right)$$

Recap of word embeddings

- Word embeddings like these capture **semantic relationships** among words
- They are learned from **datasets** – as such, they might capture the **biases** in those datasets
- We'll mostly explore examples from the following paper: "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings" (<https://arxiv.org/pdf/1607.06520>)

Word analogies

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{queen}}$$



Word analogies

Paris:France :: Tokyo:x

paris - tokyo \approx France - ?

paris - tokyo - France \approx ?



Word analogies

Sounds great!

- Surprising that simple vector arithmetic (on top of a trained representation) can capture a variety of relationships
- Useful to practitioners for a variety of applications involving natural language (e.g. document ranking, sentiment analysis, question retrieval) (see linked paper for references)

Word analogies

But can also capture biases, e.g.:

$$\vec{\text{man}} - \vec{\text{woman}} \approx \vec{\text{computer programmer}} - \vec{\text{homemaker}}$$

$$\begin{array}{l} \text{father:doctor} :: \text{mother:nurse} \\ \Downarrow \\ \text{father-mother} \approx \text{doctor-nurse} \end{array}$$

Word analogies

Food for thought: The model used in this study is based on a Google News corpus; we might assume this to be relatively authoritative (and maybe even “unbiased”) compared to e.g. webtext from the general population.

Why does this type of bias persist even in a news corpus? In what ways might a news corpus be more or less biased than other forms of text?

Word analogies

Which occupations have embeddings closest to “she” and “he”?

(“she” and “he” less ambiguous in English than “woman”, “man”, etc.; see paper for justification)

Also asked crowd workers to rank (on a scale of 1-10) occupations by gender stereotype: crowd assessment is highly correlated ($cc=0.51$) with word embedding stereotypes (so, the model is mimicking human notions of bias)

Extreme *she* occupations

- | | | |
|-----------------|-----------------------|------------------------|
| 1. homemaker | 2. nurse | 3. receptionist |
| 4. librarian | 5. socialite | 6. hairdresser |
| 7. nanny | 8. bookkeeper | 9. stylist |
| 10. housekeeper | 11. interior designer | 12. guidance counselor |

Extreme *he* occupations

- | | | |
|----------------|-------------------|----------------|
| 1. maestro | 2. skipper | 3. protege |
| 4. philosopher | 5. captain | 6. architect |
| 7. financier | 8. warrior | 9. broadcaster |
| 10. magician | 11. fighter pilot | 12. boss |

Word analogies

More examples of she:x :: he:y analogies

Gender stereotype <i>she-he</i> analogies.		
sewing-carpentry	register-nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	hairstylist-barber
Gender appropriate <i>she-he</i> analogies.		
queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Note: if we're going to "fix" these, hopefully we don't do any harm to those that *aren't* problematic!

Word analogies

Note: “nurse” being close to “woman” isn’t a problem in and of itself – its embedding is also fairly close to “man” (after all, “nurses”, “women”, and “men” are all examples of humans!)

The problem is that some words that don’t clearly convey gender are systematically closer to one gender than another, i.e., the fact that the distances are *unequal* suggests bias

So to assess bias we’ll (usually) look at the association between a word and a gender “pair”

Word analogies

Note: language contains both **definitional** and **stereotypical** gender associations

E.g. the relationship between “man” and “father” is **definitional**: we wouldn’t want to “remove” this association from a language model or we would lower its utility

Whereas the relationship between “woman” and “nurse” is **stereotypical**: we might prefer that a language model not reproduce or amplify that bias

Word analogies

The paper makes a distinction between **direct** and **indirect** bias:

Direct bias: Association between a gender neutral word and a clear gender pair (e.g. "nurse" is closer to "female" than "male")

Indirect bias: Associations between gender neutral words that clearly arise from gender; e.g. "receptionist" is much closer to "softball" than "football" due to female associations with both receptionist and softball

Note that pairs of male-biased (or female-biased) words have legitimate associations having nothing to do with gender; e.g. while the words "mathematician" and "geometry" both have a strong male bias, their similarity is justified by factors other than gender

Debiasing word embeddings

Why might we want to correct this?

- word embeddings are used downstream
- word embeddings are the foundation of LMs

Debiasing word embeddings

How can we correct this?

Two goals:

1. Reduce bias:

- a. Reduce gender associations among gender neutral words
- b. Ensure that gender neutral words (such as “nurse”) are equidistant between gender pairs (such as “he” and “she”)

2. Maintain embedding utility:

- a. Correctly maintain definitional gender associations (such as between “man” and “father”)
- b. Maintain meaningful non-gender-related associations between gender neutral words, such as fashion-related words or e.g. "mathematician" and "geometry"

Debiasing word embeddings

Plan:

Identify the gender subspace: Find differences between various gender words, and run PCA to find a principal “gender direction”

Measuring direct bias: How much are words aligned (cosine similarity) with this gender direction?

Measuring indirect bias: How much is (e.g.) *receptionist* closer to *softball* than *football* (we’ll see relevance and details later)

Debiasing word embeddings

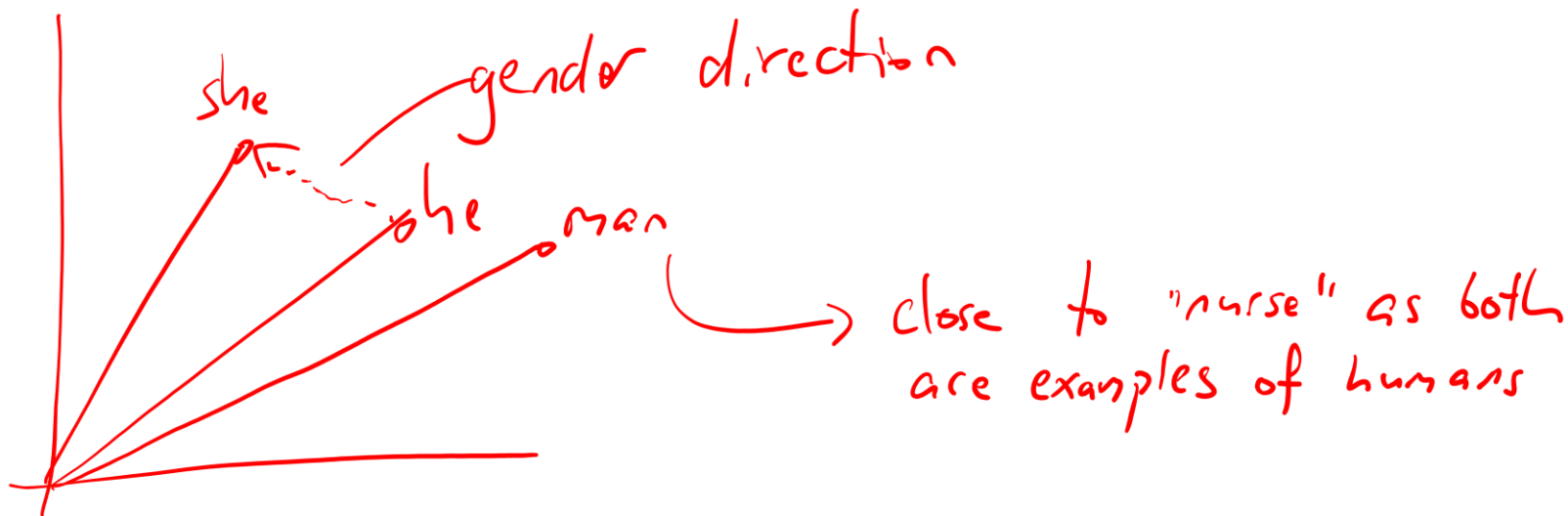
Identifying the gender subspace

Language is messy and gendered use is hard to detect! E.g. expressions like “oh man!” or “man the station” aren’t gendered (or at least not in the way that other expressions might be)

Can we find a *direction* in the subspace associated with bias?

Debiasing word embeddings

E.g. $\vec{\text{she}} - \vec{\text{he}}$ is an example of a “gender direction”:



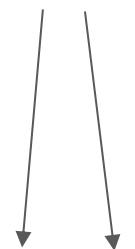
Debiasing word embeddings

Lots of possible “gender directions”:

	def.	stereo.		def.	stereo.
$\vec{\text{she}} - \vec{\text{he}}$	92%	89%	$\vec{\text{daughter}} - \vec{\text{son}}$	93%	91%
$\vec{\text{her}} - \vec{\text{his}}$	84%	87%	$\vec{\text{mother}} - \vec{\text{father}}$	91%	85%
$\vec{\text{woman}} - \vec{\text{man}}$	90%	83%	$\vec{\text{gal}} - \vec{\text{guy}}$	85%	85%
$\vec{\text{Mary}} - \vec{\text{John}}$	75%	87%	$\vec{\text{girl}} - \vec{\text{boy}}$	90%	86%
$\vec{\text{herself}} - \vec{\text{himself}}$	93%	89%	$\vec{\text{female}} - \vec{\text{male}}$	84%	75%

Debiasing word embeddings

what are these?



	def.	stereo.		def.	stereo.
$\overrightarrow{\text{she}} - \overrightarrow{\text{he}}$	92%	89%	$\overrightarrow{\text{daughter}} - \overrightarrow{\text{son}}$	93%	91%
$\overrightarrow{\text{her}} - \overrightarrow{\text{his}}$	84%	87%	$\overrightarrow{\text{mother}} - \overrightarrow{\text{father}}$	91%	85%
$\overrightarrow{\text{woman}} - \overrightarrow{\text{man}}$	90%	83%	$\overrightarrow{\text{gal}} - \overrightarrow{\text{guy}}$	85%	85%
$\overrightarrow{\text{Mary}} - \overrightarrow{\text{John}}$	75%	87%	$\overrightarrow{\text{girl}} - \overrightarrow{\text{boy}}$	90%	86%
$\overrightarrow{\text{herself}} - \overrightarrow{\text{himself}}$	93%	89%	$\overrightarrow{\text{female}} - \overrightarrow{\text{male}}$	84%	75%

- Collect most frequent “definitional” and “stereotypical” gendered words from crowd workers
- Train a simple classifier for each word pair: e.g. the “she-he” classifier predicts as “female” if a word is closer to “she” and “male” if a word is closer to “he”
- How well do these trivial classifiers align with stereotypical and definitionally gendered words?

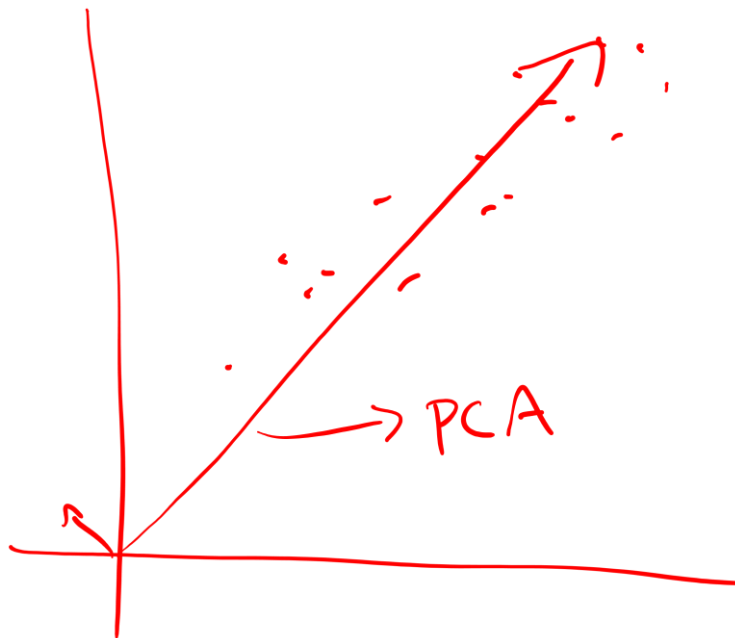
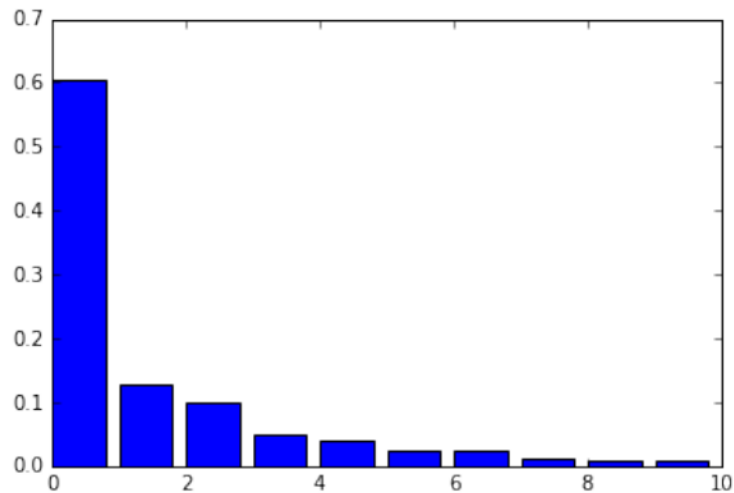
Debiasing word embeddings

Given that all of these word pairs seem to capture definitional and stereotypical bias to some extent, but maybe in different ways, just collect them together (i.e., build a set of points corresponding to these directions) and run PCA

	def.	stereo.		def.	stereo.
$\overrightarrow{\text{she}} - \overrightarrow{\text{he}}$	92%	89%	$\overrightarrow{\text{daughter}} - \overrightarrow{\text{son}}$	93%	91%
$\overrightarrow{\text{her}} - \overrightarrow{\text{his}}$	84%	87%	$\overrightarrow{\text{mother}} - \overrightarrow{\text{father}}$	91%	85%
$\overrightarrow{\text{woman}} - \overrightarrow{\text{man}}$	90%	83%	$\overrightarrow{\text{gal}} - \overrightarrow{\text{guy}}$	85%	85%
$\overrightarrow{\text{Mary}} - \overrightarrow{\text{John}}$	75%	87%	$\overrightarrow{\text{girl}} - \overrightarrow{\text{boy}}$	90%	86%
$\overrightarrow{\text{herself}} - \overrightarrow{\text{himself}}$	93%	89%	$\overrightarrow{\text{female}} - \overrightarrow{\text{male}}$	84%	75%

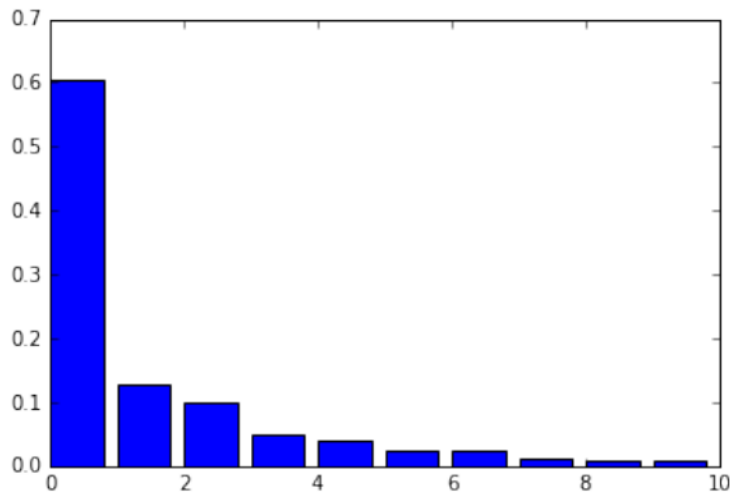
Debiasing word embeddings

singular values:



Debiasing word embeddings

singular values:



“gender directions” are roughly aligned in a single direction (as per PCA), so treat that as the main gender direction vector

Debiasing word embeddings

Measuring **direct bias** is then done as follows:

1. Identify a set of gender-neutral words (i.e., words that “should be” gender neutral) (called N)
2. Given our gender direction (called g) from above, measure direct gender bias by comparing N to g (in a few slides...)

Debiasing word embeddings

1. Identify a set of gender-neutral words (i.e., words that “should be” gender neutral) (called N):
 - Actually, enumerate gender specific words (S), and then gender-neutral words will be just what’s left
 - These “gender specific” words are just a manually-curated selection of 218 words out of 26,377
 - Use this small set to train a classifier to label the rest of the (~3 million) words (details in paper)

Debiasing word embeddings

2. Having obtain the gender-neutral word list (N) and the gender direction (g), measure direct bias as:

$$\text{Direct Bias}_c = \frac{1}{|N|} \sum_{w \in N} |\cos(\vec{w}, g)|^c$$

Debiasing word embeddings

$$\text{DirectBias}_c = \frac{1}{|N|} \sum_{w \in N} |\cos(\vec{w}, g)|^c$$

c determines how “strictly” bias is measured: if $c=0$, then $|\cos(w-g)|^c = 0$ only if w has *no overlap* with g (1 otherwise)

For reference, if N is the set of 327 occupation words from w2vNEWS, then $\text{DirectBias}_1 = 0.08$

Debiasing word embeddings

Still don't know how to measure indirect bias!

I.e., the above procedure doesn't detect (or correct) *indirect* gender associations between words like "receptionist" and "softball"

Debiasing word embeddings

3. Compare two word vectors as follows:

- Decompose word vectors as:

$$w = w_g + w_{\perp}$$

\downarrow \searrow $w - w_g = \text{remainder}$

$$(w \cdot g)g = \text{contribution from gender}$$

Debiasing word embeddings

3. Compare two word vectors as follows:

- Then compare word vectors using:

$$\beta(w, v) = \left(w \cdot v - \frac{w_{\perp} \cdot v_{\perp}}{\|w_{\perp}\| \|v_{\perp}\|} \right) / w \cdot v$$

gender component
of similarity

Debiasing word embeddings

Intuition: This operation measures *how much the inner product changes (as a fraction of the original inner product value) due to the operation of removing the gender subspace*

$$\beta(w, v) = \left(w \cdot v - \frac{w_{\perp} \cdot v_{\perp}}{\|w_{\perp}\|_2 \|v_{\perp}\|_2} \right) / w \cdot v$$

Debiasing word embeddings

Now we have everything we need to try and "debias" embeddings...

Given a matrix of embedding vectors W and a matrix N of vectors corresponding to gender neutral words; we want a transformation T that preserves inner products between word vectors while minimizing projection of gender neutral words onto the gender subspace

$$\min_T \left\| (TW)^T(TW) - W^T W \right\|^2 + \lambda \left\| (TN)^T(TB) \right\|^2$$

preserve inner products

gender subspace

minimize bias

Debiasing word embeddings

(paper also defines a different debiasing strategy though I think the above one is slightly more straightforward)

Debiasing word embeddings

Having come up with these transformations, the paper basically goes on to show that:

- Task performance doesn't decrease when the transformation is applied
- The intervention significantly reduces measurements of bias (as intended)

Food for thought

- Any thoughts about this paper? I don't think the specific techniques described here are particularly relevant for your future lives, but are valuable insofar as they reveal the subtleties involved
- The basic motivation behind debiasing word embeddings is that they're used "downstream", e.g. in language models; but:
 - Will "debaised" word embeddings stop language models from learning biases?
 - If we still need to debias the LM anyway, is there any added value to these "upstream" interventions?

Study points & take-homes

- Don't worry too much about the specific model/paper: think more about the different sources from which biases can arise, and how this type of correction is more complex than it first seems

Fairness and bias in application domains

5.3: Diversity in retrieval and recommendation

This section

- Quick background on recommender systems
- Maximal marginal relevance
- Other reranking approaches

Recommender systems

The very basic goal of a recommender system is to fit a model of the form:

$$f(u, i)$$

that predicts the compatibility between a user u and an item i . This compatibility function can then be used to rank items for each user (among other things):

$$\max_i f(u, i)$$

Recommender systems – example 1

Memory-based recommender system

e.g.:

$$\text{sim}(i, j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|}$$

set of users who bought j

$$f(u, i) = \max_{j \in I_u} \text{sim}(i, j)$$

user history

has this user ever bought something similar?

Outline

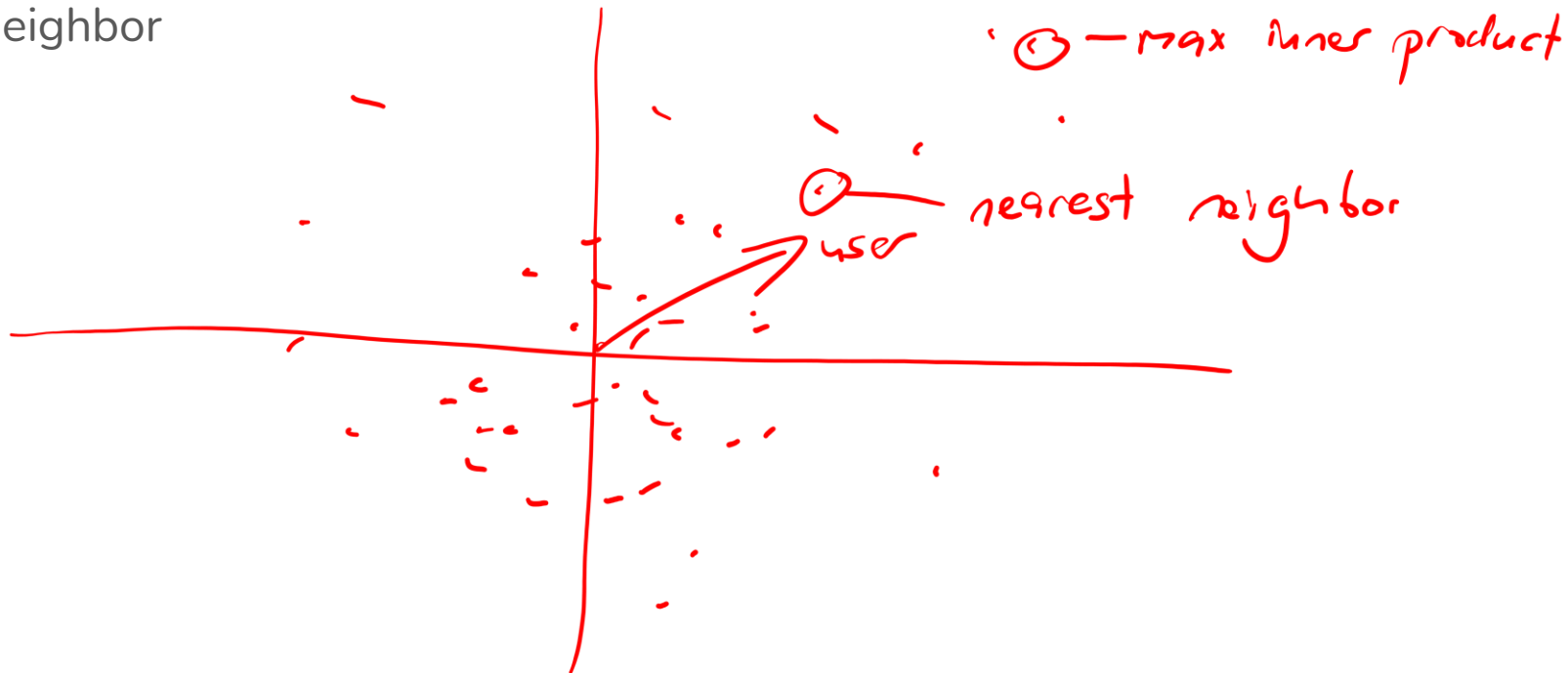
- Discuss how recommender systems (and other personalized algorithms) can lead to unfair outcomes
- Discuss general strategies to mitigate these outcomes
- Explore various case-studies about the dangers of recommendation

Fairness issues in personalized algorithms

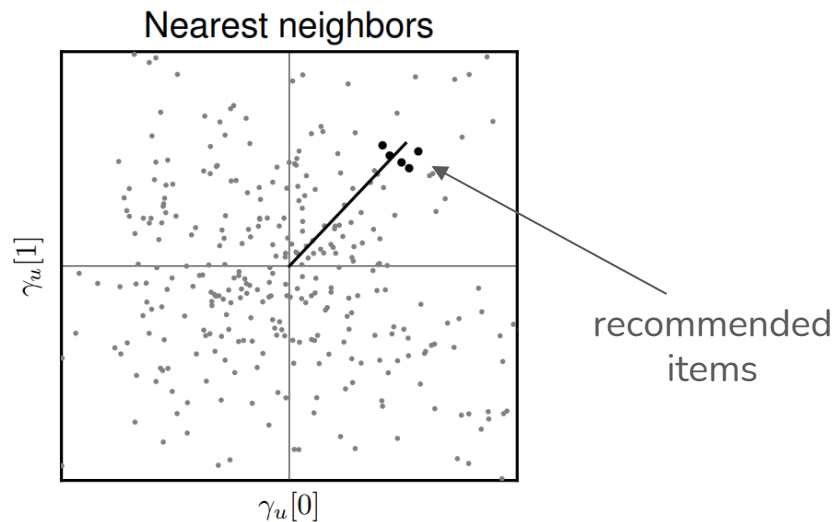
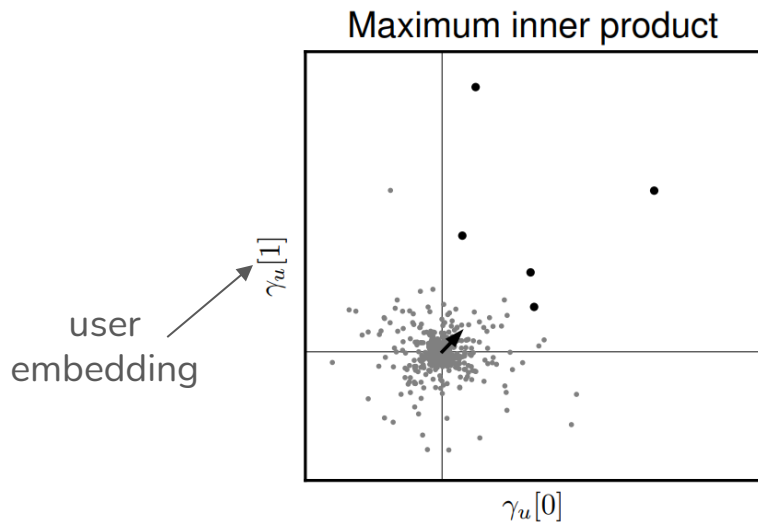
- Recommender systems may have a 'concentration' effect, where users are gradually locked into a 'filter bubble' containing only a narrow set of items
- Recommending content maximally aligned with a user's interests may gradually push users toward more and more 'extreme' content
- Recommender systems may have reduced utility for users (or groups of users) who are underrepresented in the training data
- Recommendations may focus only a user's predominant interest, while failing to capture the diversity and breadth of their interactions
- Systems could disadvantage vendors (or content creators, etc.) by failing to recommend products in the long-tail

Fairness issues in personalized algorithms

Consider e.g. recommending items by taking the maximum inner product versus a nearest neighbor



Fairness issues in personalized algorithms



Consider recommending items by taking the maximum inner product versus a nearest neighbor

$$rec(u) = \arg \max_{i \in I \setminus I_u} f(u, i)$$

$$f(u, i) = \|\gamma_u - \gamma_i\|$$

Fairness issues in personalized algorithms

- Maximizing the inner product will tend to recommend "extreme" items: if I like action, I should like *a lot* of action
- Finding nearest neighbors will tend to recommend items that are very close to what I've already consumed

How can we measure (and maybe correct) these issues
with content extremity / diversity?

Measuring diversity in recommendation

Let's try to compare users' consumption histories to what gets recommended

E.g. compute how many times each item is recommended:

```
countsPerItem = defaultdict(int)

for u in range(nUsers):
    # Given a matrix of interactions
    recs = model.recommend (u, Xui, N=len(itemsPerUser [u]))
    for i, score in recs:
        countsPerItem [i] += 1
```

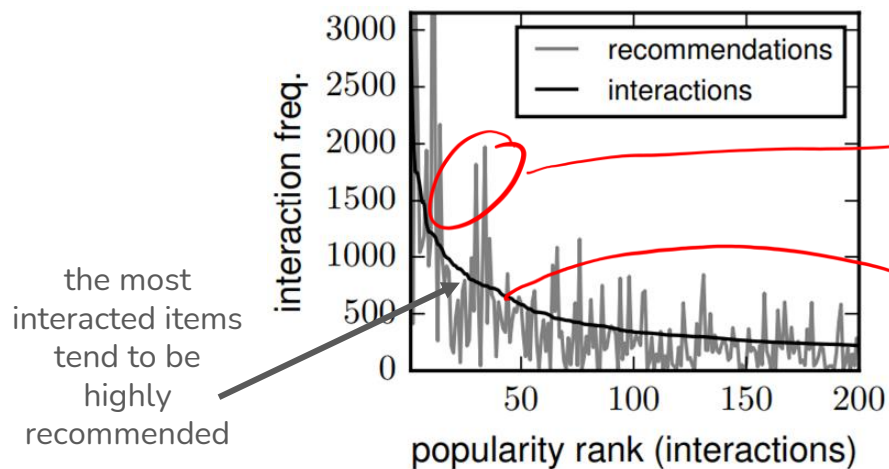
Measuring diversity in recommendation

Questions:

1. Are items that were consumed a lot the same as the ones that tend to be recommended a lot? How well do consumptions and recommendation distributions overlap?
2. What about the shape of the distribution? Are recommendations dominated by popular items (more so than consumptions?)

Measuring diversity in recommendation

Consumption versus recommendation distribution (Goodreads)

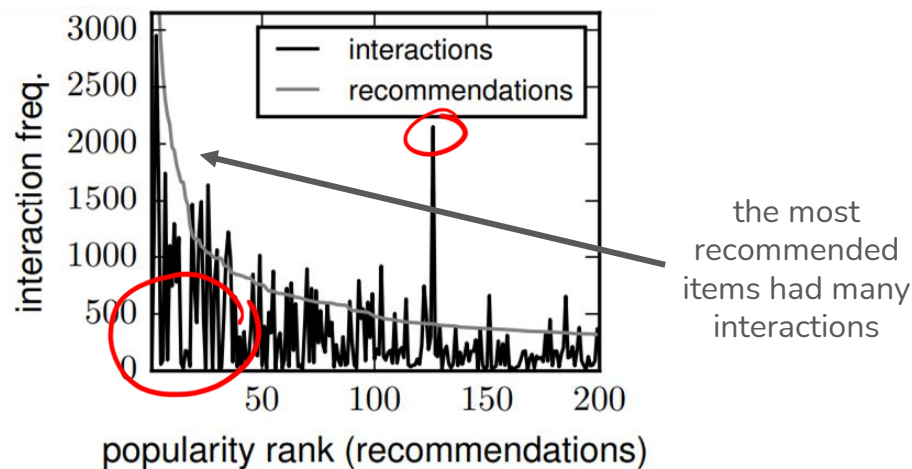
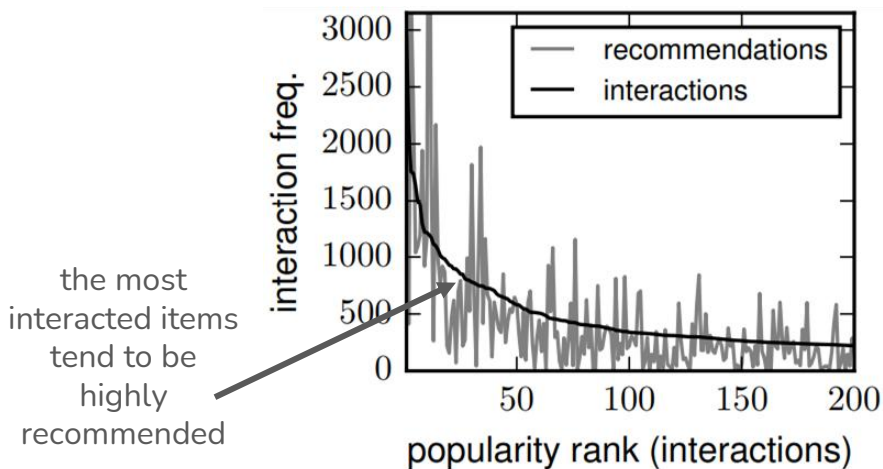


frequency of recommendation

historical # of interactions

Measuring diversity in recommendation

Consumption versus recommendation distribution (Goodreads)



Distributions match okay, but top-recommended items appear *much* more than their number of historical interactions!

Measuring diversity in recommendation

Questions:

1. Are items that were consumed a lot the same as the ones that tend to be recommended a lot? How well do consumptions and recommendation distributions overlap?
2. **What about the shape of the distribution? Are recommendations dominated by popular items (more so than consumptions?)**

Concentration

We saw in the previous example that the most-recommended items were a fair bit more popular than the most consumed items

*i.e., the recommender made popular items **more popular***

This is known as a **concentration effect**: recommendations may concentrate around a few items (and this could cause a feedback loop!)

Concentration

We can measure concentration via the Gini Coefficient:

$$G(\mathbf{y}) = \frac{\sum_{i=1}^N \sum_{j=1}^N |y_i - y_j|}{2N^2\bar{y}}.$$

The Gini coefficient measures the average difference items in a set, e.g. the average difference in wealth between individuals

- Close to zero: everyone has about the same wealth (uniform)
- Close to 1: wealth is concentrated among a few individuals

Concentration

For a recommender system, we might be interested in the difference between Gini coefficients of interactions versus recommendations

- If recommendations have a *higher* Gini coefficient than interactions, then the recommender is causing a concentration effect
- If recommendations have a *lower* Gini coefficient than interactions, then the recommender is causing a dispersion (or diversification?) effect

Concentration

Measuring the Gini coefficient

```
def gini(y, samples =1000000)
    m = sum(y) / len(y) # average
    denom = 2 * samples * m
    numer = 0
    for _ in range(samples):
        i = random.choice(y)
        j = random.choice(y)
        numer += math.fabs(i - j)
    return numer / denom
```

measured on a
sample for a
large corpus

For Goodreads:

- interactions have $G = 0.72$;
 - recommendations have $G = 0.77$;
- i.e., slight concentration

Concentration

From some real studies (more in *Personalized Machine Learning*, chapter 10):

- **Fleder and Hosanagar, 2009:** Simulate users (can accept or reject recommendations), with recommenders trained on interaction history. Over time, recommendations become more and more concentrated
- **Nguyen et al. 2014:** For real users, both recommendations and interactions become less diverse over time (in terms of content features)
- **Extremification (Ribeiro 2020, youtube):** How do recommendations on youtube guide users to extreme content? E.g. if users visit pages that have a specific slant (but are not "extreme"), will they gradually be guided to more extreme pages?
- **Content diversity (Zhou 2010, youtube):** Recommendations drive a large fraction of views, and are more diverse than what would be expected by popularity-driven models

Study points & take-homes

- So far, just try to get a sense of how recommender systems – and more generally, *personalized* algorithms – can be biased
- Understand the differences between notions we've seen so far (around e.g. gender and race) versus issues of concentration, diversification, etc., which aren't necessarily related to subgroup performance

Fairness and bias in application domains

5.4: Algorithmically correcting concentration/diversity issues

This section

- Three approaches to diversification:
 - Max marginal relevance
 - Determinantal point processes
 - Other reranking strategies

Re-ranking strategies to diversification

A simple way to make recommendations more "diverse" is just to (post-hoc) *rerank* the outputs of some recommender

Note that diversity could mean a few things:

1. Is there variety among the **set of items** a user is recommended?
2. Across **all users**, are different items recommended to different people?

Re-ranking strategies to diversification

1. Is there variety among the **set of items** a user is recommended?

Basic strategy (Carbonell and Goldstein, 1998):

- Start with the most relevant item
- Repeatedly select the next most relevant item, but penalize relevance if it's too similar to already selected items
- Repeat until we have the desired number of items

(**note:** these ideas are from *search and retrieval* rather than *recommendation*)

Maximal marginal relevance (MMR)

1. Is there variety among the **set of items** a user is recommended?

Basic strategy (Carbonell and Goldstein, 1998):

$$\text{MMR}_u(s) = \max_{j \in I \setminus I_u} \lambda \text{Sim}(u, j) - (1-\lambda) \max_{k \in S} \text{Sim}(j, k)$$

items already returned

$\lambda \text{Sim}(u, j)$ is the item relevant to the user? e.g. $\delta_u \cdot \delta_j$

$(1-\lambda) \max_{k \in S} \text{Sim}(j, k)$ is it similar to already retrieved items?

Maximal marginal relevance (MMR)

1. Is there variety among the **set of items** a user is recommended?

Basic strategy (Carbonell and Goldstein, 1998):

$$MMR = \arg \max_{i \in R \setminus S} \left[\lambda \underbrace{Sim^{user}(i, u)}_{\text{relevance to the user}} - (1 - \lambda) \overbrace{\max_{j \in S} Sim^{item}(i, j)}^{\text{similarity to already-recommended items}} \right],$$

large lambda:
only care about
relevance

small lambda:
only care about
diversity

Maximal marginal relevance (MMR)

2~1

Examples (beer recommendations):

2~0

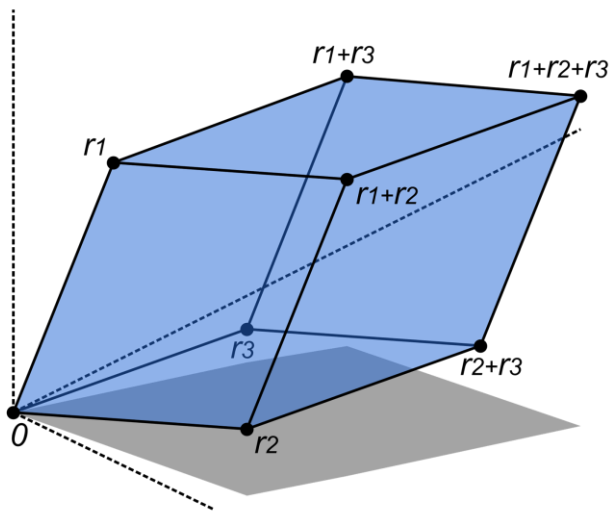
Low diversity	Medium diversity	High diversity
Founders KBS (Kentucky Breakfast Stout)	Founders KBS (Kentucky Breakfast Stout)	Founders KBS (Kentucky Breakfast Stout)
Two Hearted Ale	Samuel Smith's Nut Brown Ale	Samuel Smith's Nut Brown Ale
Bell's Hopslam Ale	Two Hearted Ale	Salvator Doppel Bock
Pliny The Elder	Bell's Hopslam Ale	Oil Of Aphrodite - Rum Barrel Aged
Samuel Smith's Oatmeal Stout	Kolsch	Great Lakes Grassroots Ale
Blind Pig IPA	Drax Beer	Blue Dot Double India Pale Ale
Stone Ruination IPA	A Little Sumpin' Extra! Ale	Calistoga Wheat
Schneider Aventinus	Odell Cutthroat Porter	Dogwood Decadent Ale
The Abyss	Miner's Daughter Oatmeal Stout	Traquair Jacobite
Northern Hemisphere Harvest Wet Hop Ale	Rare Bourbon County Stout	Cantillon Gueuze 100% Lambic

Maximal marginal relevance (MMR)

Note: this type of intervention (modifying the output of a ranked list) would have been called a *post-processing* intervention in previous modules

Determinantal Point Processes (DPPs)

1. Is there variety among the **set of items** a user is recommended?



$$\det \begin{pmatrix} L_{i,i} & L_{i,j} \\ L_{j,i} & L_{j,j} \end{pmatrix} = L_{i,i}L_{j,j} - L_{i,j}L_{j,i}$$

off diagonal:
similarity
between two
items

diagonal:
relevance of an
item

Determinantal Point Processes (DPPs)

1. Is there variety among the **set of items** a user is recommended?

Basic strategy (Kulesza and Taskar, 2012):

- Want to select a set of items with high determinant
- In practice this is hard, so approach is the same as with MMR (i.e., just iteratively select items to incrementally increase the determinant)

Re-ranking strategies to diversification

2. Across **all users**, are different items recommended to different people?

Alternate view: can we recommend things to people that are relevant but not "obvious" (**see also:** serendipity)

Re-ranking strategies to diversification

2. Across **all users**, are different items recommended to different people?

Strategy (from Adomavicius and Kwon, 2011):

- Replace an item's original rank (relevance) with:

$$rank'_u(i, t) = \begin{cases} rank^{(pop)}(i) & \text{if } r(u, i) \geq t \\ \alpha_u + rank_u(i) & \text{otherwise} \end{cases}$$

popularity (e.g. number of historical interactions) →

original rank (relevance) →

only if relevance is high enough →

Re-ranking strategies to diversification

2. Across **all users**, are different items recommended to different people?

In other words, *recommend items that I like, but which aren't popular in general*

(**see also:** tf-idf from NLP)

This will spread recommendations across less-popular items and (maybe?) help with discovery

Re-ranking strategies to diversification

Note: nothing here specific to recommendation – these types of diversification strategies could work for any ranking algorithm (e.g. MMR predates this type of recommender system altogether)

Food for thought

- Note the lack of any issues around "affirmative action" etc.: we are generally not concerned with protected attributes in the above settings, and can (generally) directly manipulate the output
- To what extent is there a tradeoff between diversity and performance? Note that in the case of recommender systems, there are often many "nearly equivalent" items, such that diversity can be achieved (almost) "for free"

Study points & take-homes

- Achieved diversity by a simple post-processing intervention
- Worth implementing one of these interventions (they're quite straightforward) and exploring to what extent diversity comes at the cost of model performance

Fairness and bias in application domains

Case study: Calibration

Desirable features of a recommender

So far we've focused on *diversity* as our main metric (other than accuracy/relevance). What other features are desirable?

- Items should be **novel**, i.e., we should balance discovery of new items against recommending items with high interaction probability (but which are already known)
- Rather than being internally diverse, we might have goals such as **mutual compatibility** among items (see e.g. outfit generation)
- Recommended items should have good **coverage**, i.e., they should represent a broad range of categories or features; or they should be **balanced**, in terms of matching the category distribution from the user's history
- Other goals could be more nebulous, such as perceived **unexpectedness**, **serendipity**, or overall user **satisfaction**

Calibration (Steck, 2018)

We'll look at one specific beyond-accuracy goal: **calibration**

Idea: Recommendations should have similar attribute proportions to my past interactions

E.g. if I watched 80% romance and 20% comedy on *Netflix*, my recommendations should not be 100% romance

Calibration (Steck, 2018)

First, define a probabilistic attribute vector for each item $p(g|i)$

E.g. Harry Potter might be 10% romance, 5% comedy, 20% action, 50% fantasy (etc.)

Distribution of all recommended genres should match historical genre consumption

Calibration (Steck, 2018)

Second, measure recommended versus historical genre distribution for all items consumed/recommended for a user u :

(optional) weight genres for items

historical:
$$p(g|u) = \frac{\sum_{i \in I_u} w_{u,i} \cdot p(g|i)}{\sum_{i \in I_u} w_{u,i}}$$

recommended:
$$q(g|u) = \frac{\sum_{i \in R_u} w_{r(i)} \cdot p(g|i)}{\sum_{i \in R_u} w_{r(i)}}$$



Calibration (Steck, 2018)

Second, measure recommended versus historical genre distribution for all items consumed/recommended for a user u :

$$\begin{aligned} \text{historical: } p(g|u) &= \frac{\sum_{i \in I_u} w_{u,i} \cdot p(g|i)}{\sum_{i \in I_u} w_{u,i}} \\ \text{recommended: } q(g|u) &= \frac{\sum_{i \in R_u} w_{r(i)} \cdot p(g|i)}{\sum_{i \in R_u} w_{r(i)}} \end{aligned}$$

consumed items

(optional) weighting, e.g. by recency

recommended items

Calibration (Steck, 2018)

The goal is that the two distributions should (approximately) match:

(they use the KL divergence)

$$\text{KL}(p, q) = \sum_g p(g|u) \log \frac{p(g|u)}{q(g|u)}$$

Calibration (Steck, 2018)

In practice recommendation is the same as with our diversity approaches, i.e., iteratively add new recommendations that balance compatibility and calibration:

$$R_u = \arg \max_R (1 - \lambda) \cdot \sum_{i \in R} f(u, i) - \lambda \cdot \text{KL}(p, q(R))$$

compatibility

calibration

small lambda:
only care about
relevance

large lambda:
should match
historical
distribution very
closely

Fairness and bias in application domains

5.5: Fairness interventions in recommender systems

This section

- Discuss how fairness interventions can be implemented in recommender systems
- Give examples of *in-processing* and *post-processing* interventions
- Introduce related ideas of C-, P-, and CP-fairness

Fairness interventions in recommender systems

Recall that when studying fairness interventions we looked at three classes of approach:

1. **Pre-processing** : modify the *dataset* to improve the outcomes of methods trained on that dataset
2. **In-processing** : modify the *training objective* e.g. to include a fairness penalty
3. **Post-processing** : modify the model's *outputs* (e.g. predictions) to correct outcomes after-the-fact

Fairness interventions in recommender systems

We'll look at a few potential approaches from each of these categories:

1. **Pre-processing** : (couldn't find a good paper!)
2. **In-processing** : Incorporate a fairness penalty into the recommendation directive
3. **Post-processing** : Re-rank recommendations to achieve fairer outcomes (already seen this in the form of other re-ranking approaches, e.g. MMR and Calibration)

Fairness interventions in recommender systems

Recall we're optimizing something like:

$$r(u, i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$$

I.e., we're making personalized recommendations to each user based on some model

In what ways could such a model be problematic, unbiased, or unfair?

Fairness interventions in recommender systems

Yao & Huang (2017) studied fairness in a recommendation setting based on online course evaluations of CS classes

Model is of the form:

$$r_{ij} \approx \mathbf{p}_i^\top \mathbf{q}_j + u_i + v_j$$

(same as previous slide except for notation)

What might happen if females (or any group) are underrepresented in this type of data?

Fairness interventions in recommender systems

E.g. the underrepresented group might have their ratings over or underpredicted ("value unfairness")

$$U_{\text{val}} = \frac{1}{|I|} \sum_{i=1}^{|I|} \left| \left(\underbrace{\mathbb{E}_g[y]_i}_{\text{expected}} - \overbrace{\mathbb{E}_g[r]_i}^{\text{average rating for group } g \text{ on item } i} \right) - \left(\mathbb{E}_{\neg g}[y]_i - \mathbb{E}_{\neg g}[r]_i \right) \right|$$

misprediction for males misprediction for females

(equation is a mouthful but it's just the difference of mispredictions for the two groups)

Fairness interventions in recommender systems

We could measure related quantities in various ways:

$$U_{\text{abs}} = \frac{1}{|I|} \sum_{i=1}^{|I|} \left| \left| \mathbb{E}_g[y]_i - \mathbb{E}_g[r]_i \right| - \left| \mathbb{E}_{\neg g}[y]_i - \mathbb{E}_{\neg g}[r]_i \right| \right|$$

$$U_{\text{under}} = \frac{1}{|I|} \sum_{i=1}^{|I|} \left| \max\{0, \mathbb{E}_g[r]_i - \mathbb{E}_g[y]_i\} - \max\{\mathbb{E}_{\neg g}[r]_i - \mathbb{E}_{\neg g}[y]_i\} \right|$$

$$U_{\text{over}} = \frac{1}{|I|} \sum_{i=1}^{|I|} \left| \max\{0, \mathbb{E}_g[y]_i - \mathbb{E}_g[r]_i\} - \max\{\mathbb{E}_{\neg g}[y]_i - \mathbb{E}_{\neg g}[r]_i\} \right|$$

Fairness interventions in recommender systems

Ultimately each is a form of **disparity** between the two groups

First main point is that disparities are manifest in real datasets with standard recommendation approaches (most experiments are on movie recommendation, across categories that exhibit different levels of gender imbalance)

Table 2: Gender-based statistics of movie genres in Movielens data.

	Romance	Action	Sci-Fi	Musical	Crime
Count	325	425	237	93	142
Ratings per female user	54.79	52.00	31.19	15.04	17.45
Ratings per male user	36.97	82.97	50.46	10.83	23.90
Average rating by women	3.64	3.45	3.42	3.79	3.65
Average rating by men	3.55	3.45	3.44	3.58	3.68

Fairness interventions in recommender systems

Second point is that these fairness objectives can be incorporated into training with little loss in performance:

E.g.:

$$\frac{1}{\mathcal{T}} \sum_{(u,i) \in \mathcal{T}} \underbrace{(\alpha + \beta_i + \beta_u + \gamma_i + \gamma_u - R_{u,i})^2}_{\text{accuracy}} + \lambda \underbrace{U_{\text{abs}}}_{\text{(absolute) fairness}}.$$

(note: this is an example of **in-processing**)

Fairness interventions in recommender systems

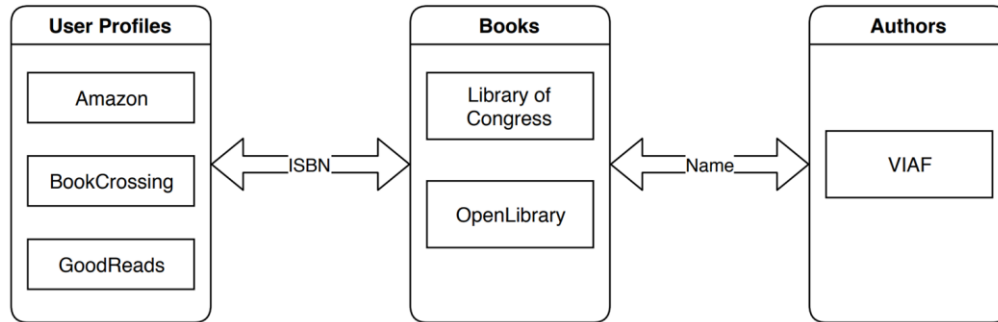
Final points:

- No issues about "sensitive attributes" etc.: we assumed here that we can directly incorporate the gender attribute into the objective without running into (e.g.) legal barriers
- Results can be fairer without causing much harm to the overall accuracy; it can be a general phenomenon in recommender systems that there are many "nearly equivalent" recommendations, so intervening for fairness can have relatively low cost

Fairness interventions in recommender systems

Ekstrand and Kluver (2019) explored gender in book recommendations, on Amazon, BookCrossing, and GoodReads

Review data is available, but some considerable effort is needed to extract author gender:



Fairness interventions in recommender systems

Various research questions:

RQ1/2: To what extent are female authors over/underrepresented in the dataset (1) and among users' consumption patterns (2)

RQ3/4: To what extent do recommenders mimic or exacerbate any imbalance?

RQ5: Can this be algorithmically corrected, and what is the cost in doing so?

Aside: C-, P-, and CP-fairness

In recommender systems, fairness can be viewed from the perspective of the **consumer (C)**, the **producer (P)**, or **both (CP)** (Burke, 2017)

- Previous paper: C-fairness (user gender)
- This paper: P-fairness (gender of authors, associated with items)
- Next paper: CP-fairness (both)

Fairness interventions in recommender systems

RQ 1/2 (gender in data and interactions): Male-authored books are overrepresented; less so in interactions (and in fact less than an ostensible distribution of all authors); individual users are quite diverse.

		Books		Ratings	
Data		female	male	female	male
library of congress	→ LOC	22.7%	77.3%	—	—
amazon	→ AZ	30.6%	69.4%	38.9%	61.1%
bookcrossing ratings	→ BX-E	40.5%	59.5%	43.0%	57.0%
all interactions	→ BX-I	40.7%	59.3%	45.7%	54.3%
goodreads	→ GR-E	37.8%	62.2%	47.6%	52.4%
	GR-I	37.7%	62.3%	48.2%	51.8%

Fairness interventions in recommender systems

RQ 3/4 (distribution after recommendation): Several standard recommendation approaches are considered:

- *Implicit* models make use of interactions
- *Explicit* models make use of ratings
- Mostly, implicit models preserve users' historical gender skew
- Some explicit models propagate the overall skew of the data (i.e., toward male authors)

Fairness interventions in recommender systems

RQ 5 (algorithmic correction): Rather than correcting this bias during training (like in Yao & Huang), bias is corrected using a post-hoc reranking strategy

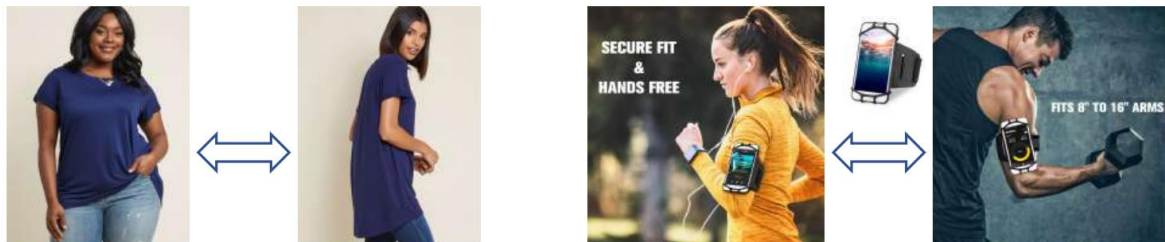
Strategy(ies) are simple greedy algorithms that recommend items with high utility while enforcing a balance constraint

(note: this is an example of a post-processing intervention; this roughly corresponds to "affirmative action" in the fairness literature)

Fairness interventions in recommender systems

One more paper...

Motivating question (Wan et al. 2019): when I buy products, how much am I influenced by whether models who market the product look "like me" (gender, race, body type, etc.)? (Also called "self congruency")



Fairness interventions in recommender systems

Potential fairness issues:

- Users poorly represented by marketing may struggle to find products they like
- Vendors may miss out on sales by mis-marketing their products

This is an example of “**multisided**” (CP) fairness in recommendation (see e.g. Burke, 2017)

Fairness interventions in recommender systems

Main research questions are similar to previous ones:

RQ1: Do users follow self-congruency when selecting items? (even for items where we might expect this to be irrelevant)

RQ2: Does this lead to fairness issues in recommended items?

RQ3: Can this be algorithmically mitigated?

Fairness interventions in recommender systems

Main problem is actually dataset construction...

(1) ModCloth clothing data. ModCloth explicitly states body type of models (small, plus, etc.)



Can separate users into groups based on which body types they buy; and we can find items that are available in multiple types but modeled using a specific type

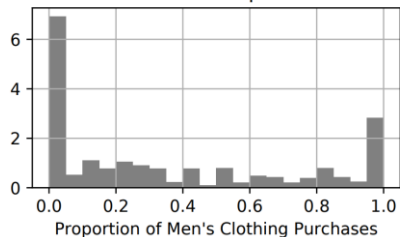
Fairness interventions in recommender systems

Main problem is actually dataset construction...

(2) Amazon Electronics data



Hist. of Users' Gender-Specific Purchases



Use an off-the-shelf detector to determine gender in marketing images (Face++);
determine user "gender" based on their purchases in clothing categories

Fairness interventions in recommender systems

Note: lots of **big** assumptions being made here!

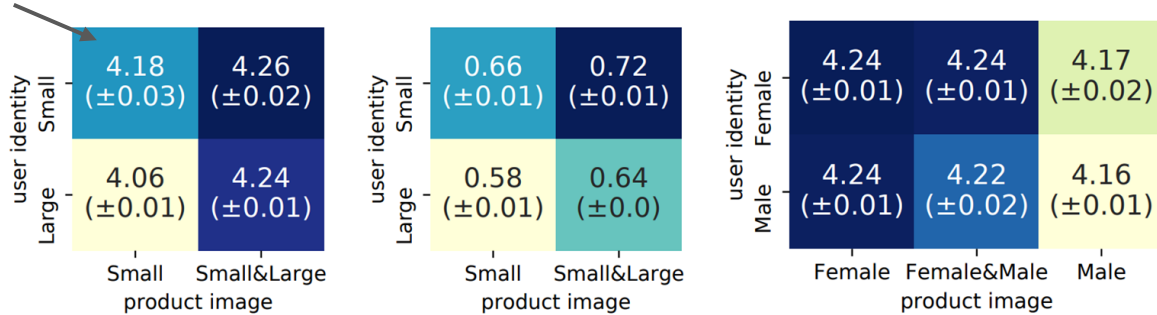
(E.g.) clothing marketed with a plus-size model may not map well to a small user, even if a small version is available

The points are to (a) **measure** whether self-congruency bias exists; (b) to determine whether it's **propagated** by recommendations; and (c) to algorithmically correct it **in any specific cases where we might want to**

Fairness interventions in recommender systems

RQ1: Do users follow self-congruency when selecting items? Users interact with / give higher ratings to products that are marketed specifically to their group:

sample mean



(again, this doesn't necessarily point to any fairness issue)

Fairness interventions in recommender systems

RQ2: Do recommender systems propagate bias? Users who buy products not marketed to them receive lower utility (higher error) from recommendations (F-test)

$$\text{Product Image} \begin{cases} \text{Female} \\ \text{Male} \end{cases} \begin{bmatrix} \bar{e}_{F,F} & \bar{e}_{M,F} \\ \bar{e}_{F,M} & \bar{e}_{M,M} \end{bmatrix}$$

$\underbrace{\hspace{10em}}_{\text{User Identity}}$

(this potentially *is* a fairness issue)

Fairness interventions in recommender systems

RQ3: Can this be algorithmically corrected? This is corrected using a similar strategy to what we saw previously:

$$\sum_{u,i} \underbrace{(f(u,i) - r_{u,i})^2}_{\text{prediction error}} + \alpha \overbrace{\mathcal{L}_{corr}}^{\text{error parity on market segments}} .$$

Again, results show error parity can be achieved with little loss in utility (and sometimes a gain in utility!)

Summary

- Fairness in recommendation has quite different metrics / goals than traditional fairness problems
 - Most have to do with loss of utility for certain groups, and ensuring that recommenders don't make things **worse**
 - Straightforward correction strategies that balance fairness objectives with recommendation utility
 - Topic is still quite new and open!
-
- Lots of other related perspectives: e.g. calibration, filter bubbles, content diversity, extremification

Study points & take-homes

- Understand how fairness interventions in recommender systems differ from those in classification in terms of goals, use of sensitive attributes, etc.

Fairness and bias in application domains

Case study: Bias in conversational recommenders

What is conversational recommendation?

“Conversational recommendation” refers to a set of techniques that try to make recommender systems more “human-like” in terms of the mechanisms they use to make recommendations

These are an interesting form of algorithm that is:

- **Explainable**, in the sense that (e.g.) an LLM can say why an item was recommended
- **Contestable**, in the sense that the user can “push back” against the recommendations and get new ones (that’s kind of the point!)

But let’s explore whether these types of interpretable methods are also biased

Some traditional approaches...

Traditional approaches rarely involved “conversation” as we might normally think of it:

- Thompson et al., 2004 (query refinement): Elicits users’ preferences and constraints with regard to item attributes;
- Mahmood and Ricci, 2009 (reinforcement learning): Queries users about recommendation attributes during each round; learns a policy to choose queries to efficiently yield a desirable recommendation

User Name	Homer						
Attributes	w_i	Values and probabilities					
Cuisine	0.4	Italian	French	Turkish	Chinese	German	English
		0.35	0.2	0.25	0.1	0.1	0.0
Price Range	0.2	one	two	three	four	five	
		0.2	0.3	0.3	0.1	0.1	
...					
Parking	0.1	Valet		Street		Lot	
		0.5		0.4		0.1	
Item Nbr.	0815	5372	7638	...	6399		
Accept/Present	23 / 25	10 / 19	33 / 36	...	12 / 23		

(from Thompson et al.)

Some traditional approaches...

Traditional approaches rarely involved “conversation” as we might normally think of it:

- Christakopoulou et al., 2016 (iterative recommendation): Collects feedback about recommended items in order to iteratively learn user preferences; explores various query strategies to elicit preferences quickly

Greedy: $j^* = \arg \max_j y_{ij}$

A trivial *exploit*-only strategy: Select the item with highest estimated affinity mean.

Random: $j^* = \text{random}(1, N)$

A trivial *explore*-only strategy.

Maximum Variance (MV): $j^* = \arg \max_j \epsilon_{ij}$

A *explore*-only strategy, variance reduction strategy: Select the item with the highest noisy affinity variance.

Maximum Item Trait (MaxT): $j^* = \arg \max_j \|\mathbf{v}_j\|_2$

Select the item whose trait vector \mathbf{v}_j contains the most information, namely has highest L2 norm $\|\mathbf{v}_j\|_2 = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jd}^2}$.

Minimum Item Trait (MinT): $j^* = \arg \min_j \|\mathbf{v}_j\|_2$

Select the item with trait vector with least information.

Upper Confidence (UCB): $j^* = \arg \max_j y_{ij} + \epsilon_{ij}$

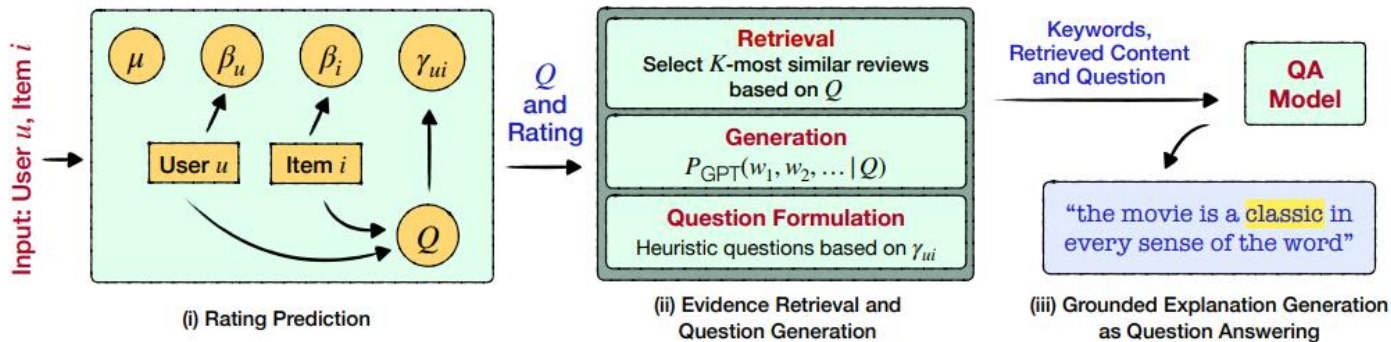
Based on UCB1 [3]: Pick the item with the highest upper confidence bound, namely mean plus variance (95% CI)

Thompson Sampling (TS) [5]: $j^* = \arg \max_j \hat{y}_{ij}$

For each item, sample the noisy affinity from the posterior. Select item with the maximum sampled value.

(from Christakopoulou et al.)

Related: “explainable” recommendations



(from Xie et al., 2022)

Explainable recommenders associate natural language explanations with each recommendation (or something like this)

Such models represent “half” of a conversational model, though lack interactive mechanisms for the user to participate in conversation

Actual conversation...

Li et al. (2018) sought approaches more closely matching “free-form” conversation. Roughly:

- Dialogs (around 10k) are constructed by crowd workers, who assume roles of a *recommender* or *seeker*;
- Conversations between the recommender and the seeker are tagged in terms of the movies mentioned, as well as explicit feedback (has the seeker seen the movies mentioned and did they like them);
- Train a dialog generation model that can fulfil the role of the recommender;
- Preferences can then be estimated and the output controlled to reference specific movies

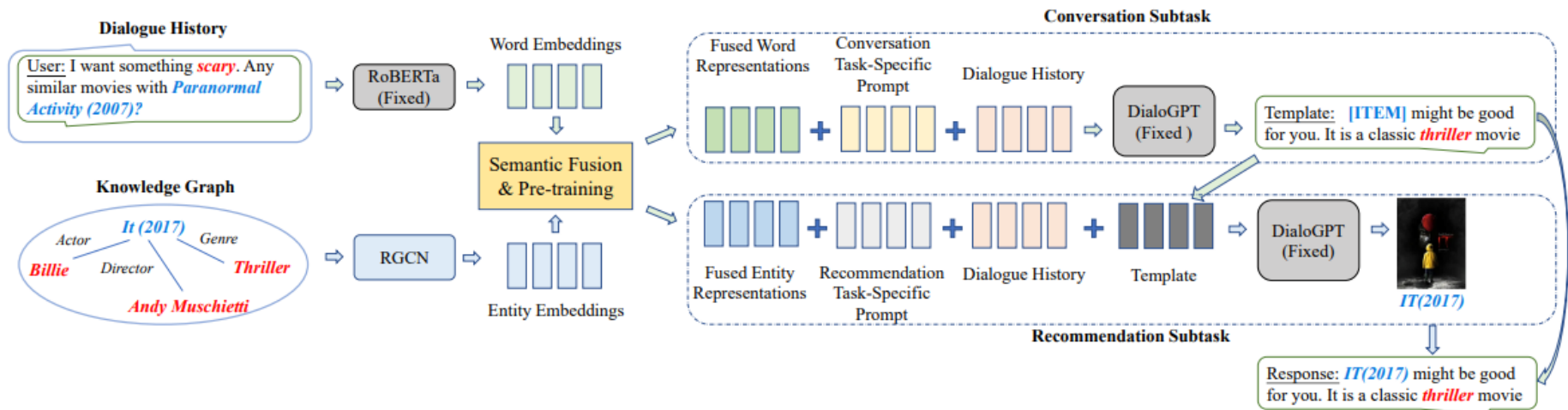
Table 5: Sample conversation from validation set

SEEKER:	hi !
HUMAN:	hi ! how is your morning going ?
HRED:	hi , what kind of movies do you like
OURS:	hello ! what kind of movie are you looking for ?
SEEKER:	great , thanks ! i was thinking of starting the day off with a great comedy , any suggestions ?
HUMAN:	i love the hangover it always makes me laugh . and i bet lots of people are hungover this morning ! airplane ! is another classic of mine .
HRED:	that 's a good one ! have you seen the lego movie ? it 's a great movie !
OURS:	i have n't seen that one ! what kind of movies do you like ? i can help you with that .
SEEKER:	i 've seen the hangover it 's really good ! i thought about watching knocked up have you seen it ?
HUMAN:	i enjoyed that movie . i enjoyed spy it 's a well done parody .
HRED:	it 's about a man who gets revenge on a cyborg and a hockey team . it 's really good . it 's really good . i also like the girl on the train
OURS:	i have n't seen that one . i will have to check it out . i hope i was able to help you find a good one ! have a great day !
SEEKER:	thanks ! i 'll check it out !
HUMAN:	sure ! have a great day !
HRED:	you 're welcome ! have a great day !
OURS:	have you seen revenge of the nerds ? it 's really good . it 's a classic .
SEEKER:	you too .

(from ReDial)

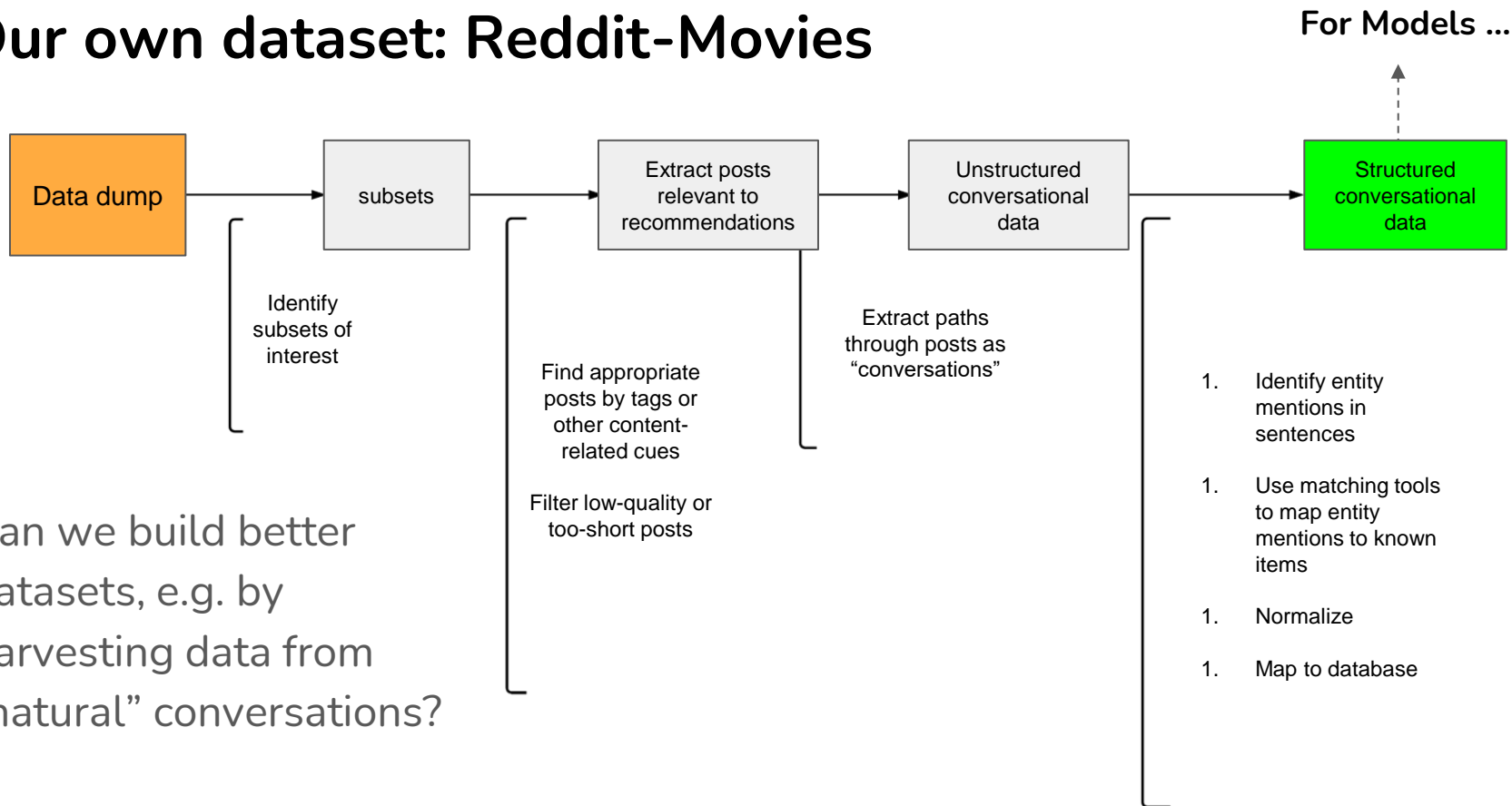
“LM+RecSys” approaches (UniCRS; Wang et al., 2022)

(Fairly) recent attempts incorporate knowledge grounding, and arguably (among a few others) represented the pre-LLM state-of-the-art



(UniCRS)

Our own dataset: Reddit-Movies

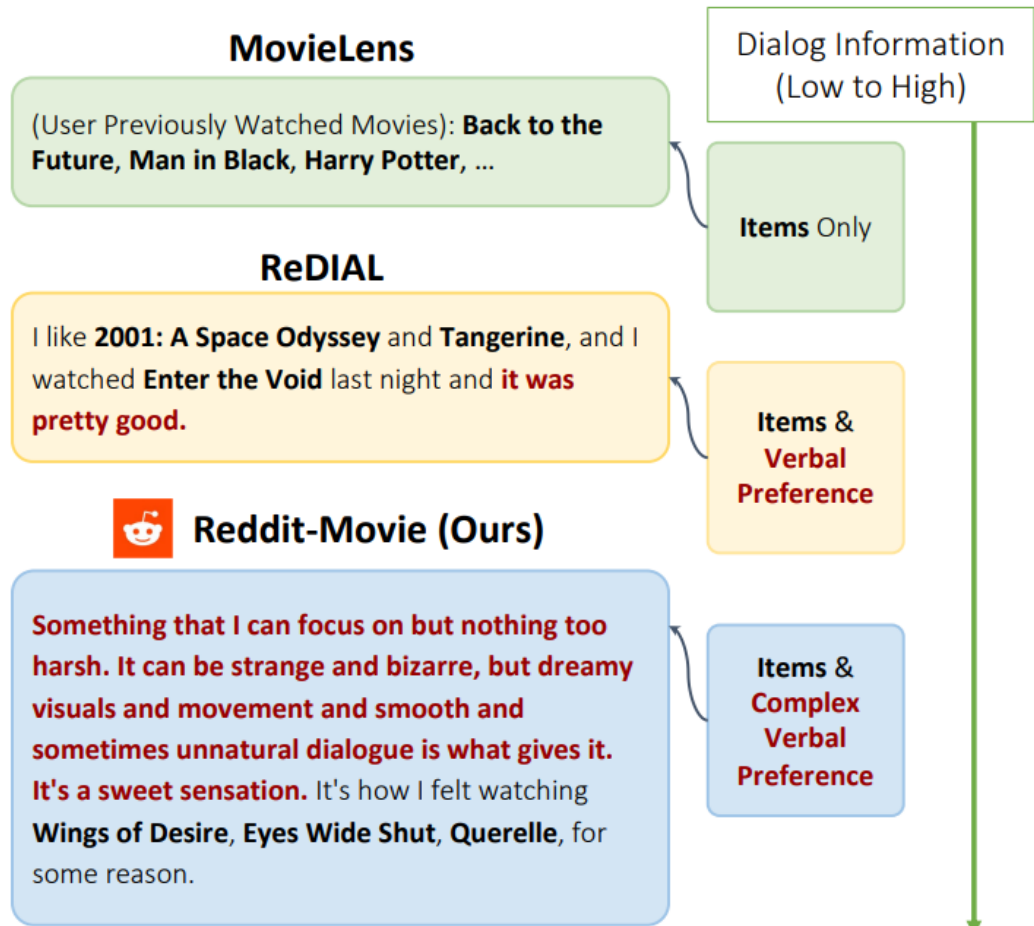


Can we build better datasets, e.g. by harvesting data from “natural” conversations?

Reddit-Movie Dataset

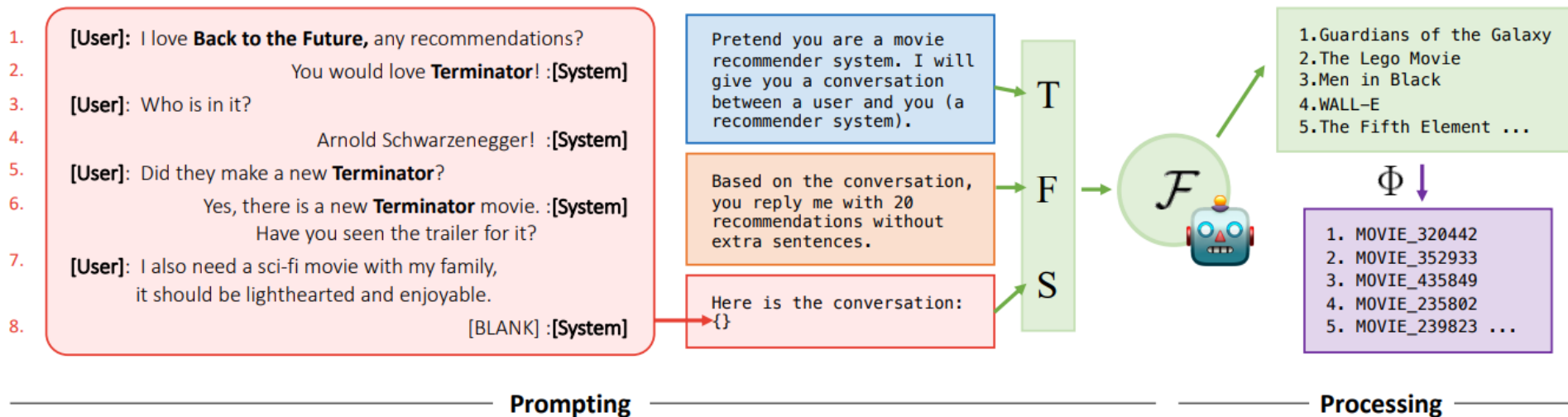
- 634,392 movie recommendation conversations, featuring 1.7M dialog turns
- ~11k users, ~24k items
- (compare to e.g. ReDial, featuring ~10k conversations, ~139k turns, ~800 users)

Much bigger than existing datasets; conversations are shorter; they have much more *context*; and (for better or worse) have much more varying structure



What do these new datasets reveal?

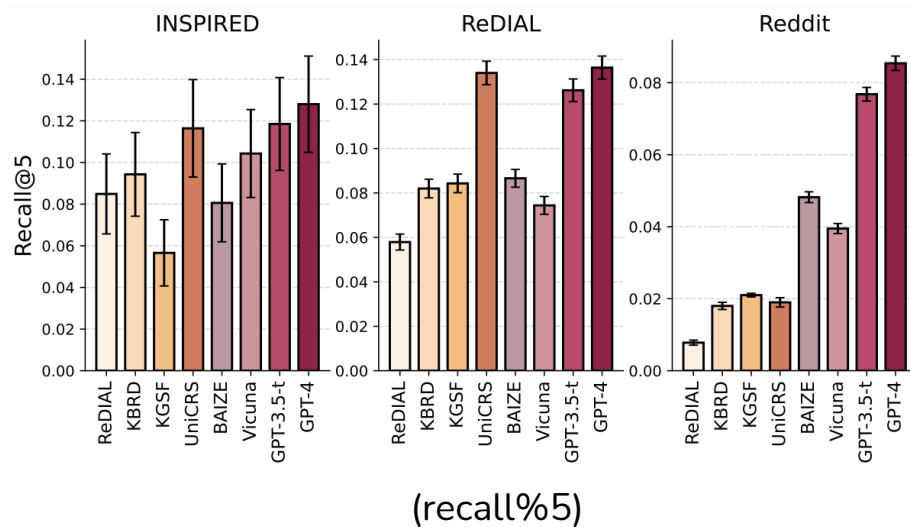
We use a simple prompting setup to compare LLMs:



What do these new datasets reveal?

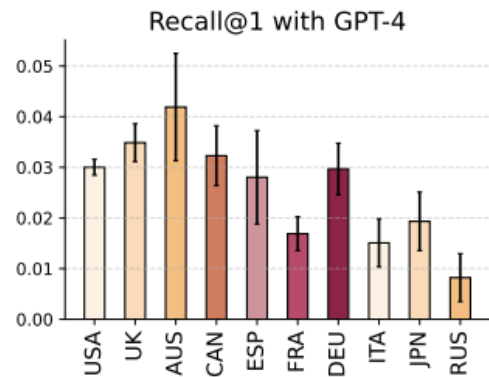
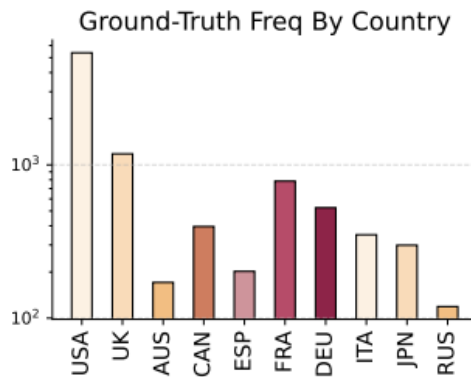
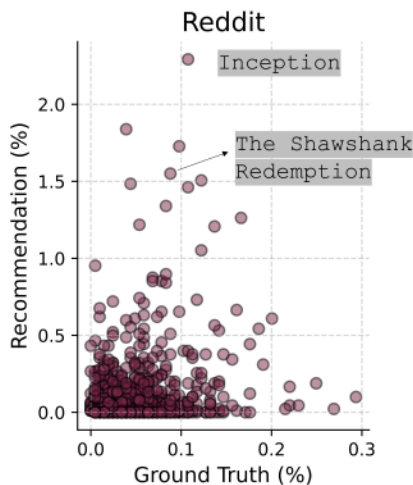
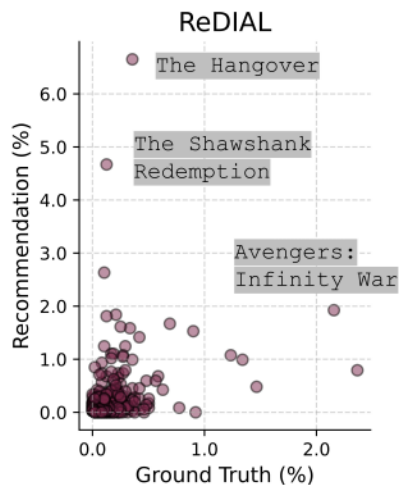
Some observations about model performance:

- Existing models engage in *shortcut learning* by focusing on repeated items (i.e., items already mentioned in a dialog but not as recommendations)
- LLMs outperform existing fine-tuned models; GPT-4 outperforms other LLMs
- LLMs generate some out-of-dataset items, but not many hallucinated recommendations (<5%); can be dealt with by string matching



Some observations about model performance:

- Significant “popularity bias” (and other bias) issues
- Recommendation performance is highly sensitive to geographical region (presumably just due to groundtruth frequency)

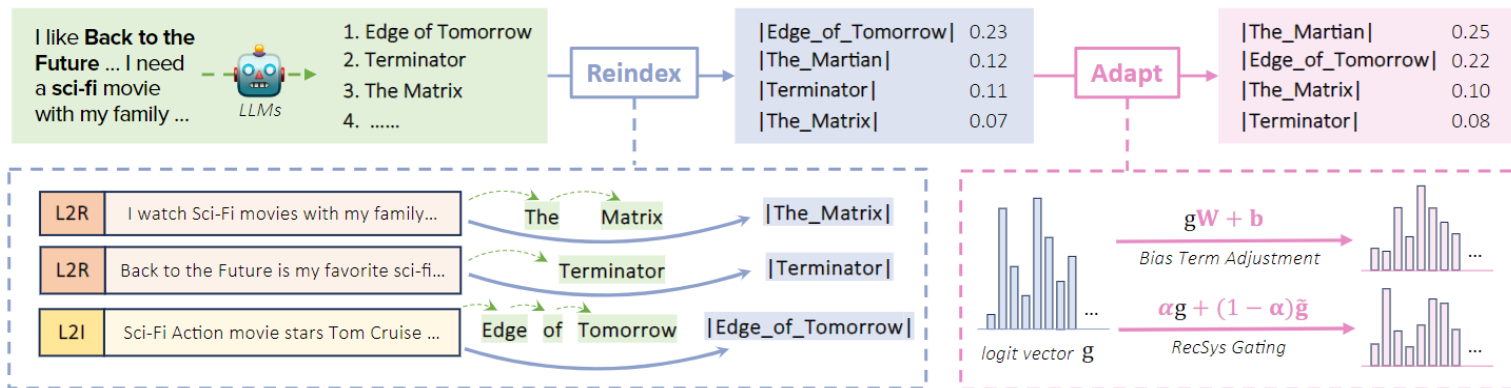


Reindex-then-Adapt

How can we fix these fairness issues?:

- Easy enough with traditional recommenders (we've already seen some more-or-less appropriate intervention strategies)
- Language models are less controllable: they generate *language tokens* rather than items: an "item" is really just a series of (English) tokens
- So, adapt the LM to have new "tokens" corresponding to items, and then use a traditional recommender to control the item distribution at decoding time

Reindex-then-Adapt



- Re-index: Train a tunable network to map multi-token item names into a new token in LLMs vocabulary
- Adapt: Tuning a few parameters (e.g. bias term only, ensembling a small recsys model) to adjust output probability distribution over those new item tokens

Summary

- Lots of "solved" fairness problems become "unsolved" once we're in the (very hard to control) world of language models
- In this instance, our own solution to making models fairer involved separating the roles of the language model and the roles of the recommender, so that fairness interventions can be implemented directly in the recommender

References for Module 5

- Gender Bias in Coreference Resolution: <https://aclanthology.org/N18-2002.pdf>
- Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings: <https://arxiv.org/pdf/1607.06520>
- Beyond Parity: Fairness Objectives for Collaborative Filtering: <https://arxiv.org/pdf/1705.08804>
- Large Language Models as Zero-Shot Conversational Recommenders: <https://arxiv.org/pdf/2308.10053>
- Reindex-Then-Adapt: Improving Large Language Models for Conversational Recommendation: <https://arxiv.org/pdf/2405.12119>

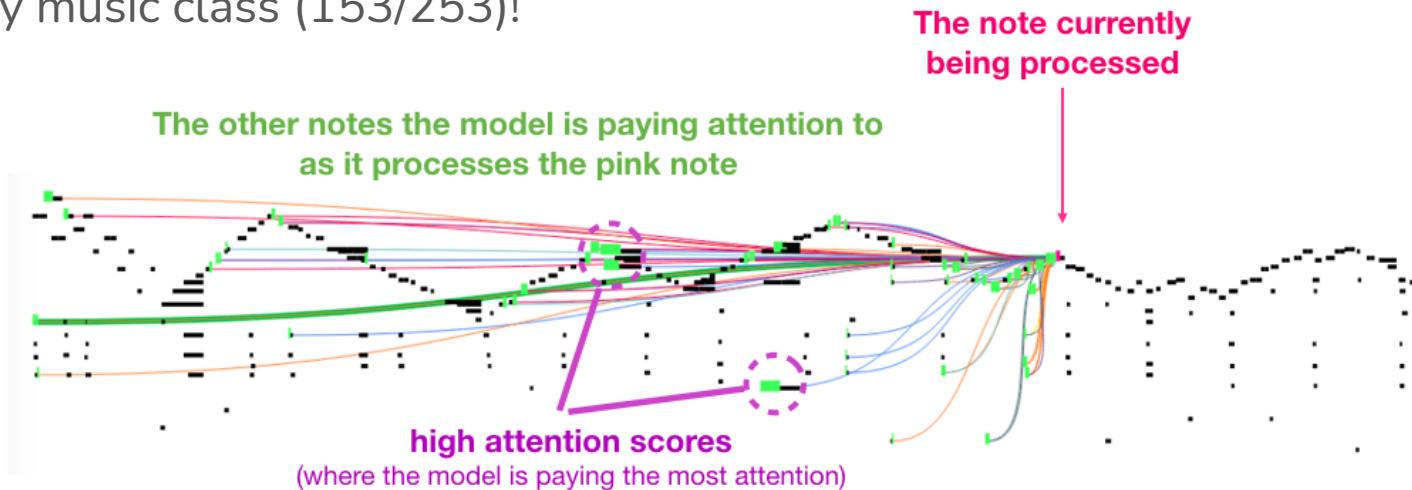
What *should* this course cover that it currently doesn't?

Counterfactual explanations: Explaining an outcome by considering what *could* have happened instead, if conditions had been different

Model contestability: Are systems able to respond to user disputes?

Visualizing attention mechanisms

Take my music class (153/253)!



Legend

attention head #1

attention head #3

attention head #5

High attention score

attention head #2

attention head #4

attention head #6

Low attention score

What *should* this course cover that it currently doesn't?

Please post any feedback to Piazza!

And please fill out course evaluations!

Thanks!