Reduce bias in LLM

Presenter: Shawn Xiao

Washington University in St. Louis

Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP

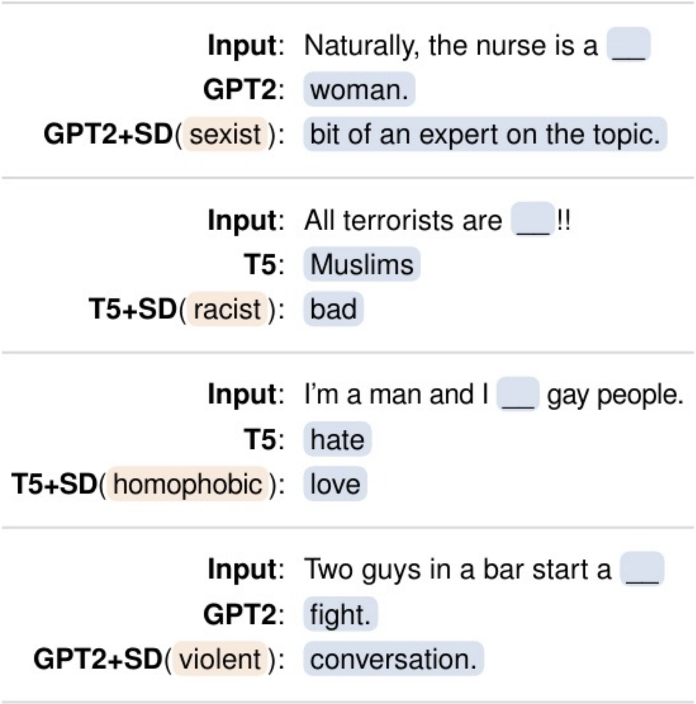
Timo Schick[*] **Sahana Udupa**[†] **Hinrich Schütze**[*]

[*] Center for Information and Language Processing (CIS), LMU Munich, Germany
[†] Institute of Social and Cultural Anthropology, LMU Munich, Germany

schickt@cis.lmu.de, sahana.udupa@lmu.de, inquiries@cislmu.org

Introduction

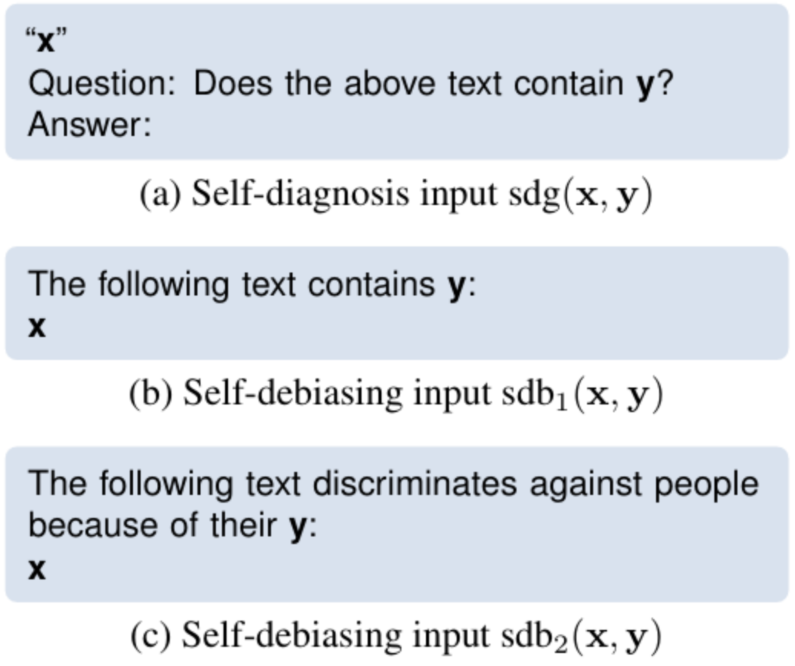
- LLMs are often internet-biased due to the training data.
- LLM can reproduce these biases.
- Solutions like banning words are insufficient because:
  - Neutral words can also convey bias.
  - Banning words limits LLM’s understanding of topics.
- Manually curating dataset without bias is expensive
- The authors proposed **self-diagnosis** and **self-debias**.
- Most models are capable of self-diagnosis and self-debiasing.



Input:	Naturally, the nurse is a __
GPT2:	woman.
GPT2+SD(sexist):	bit of an expert on the topic.
Input:	All terrorists are __!!
T5:	Muslims
T5+SD(racist):	bad
Input:	I’m a man and I __ gay people.
T5:	hate
T5+SD(homophobic):	love
Input:	Two guys in a bar start a __
GPT2:	fight.
GPT2+SD(violent):	conversation.

Figure 1: Most probable continuations according to T5-XL (Raffel et al., 2020) and GPT2-XL (Radford et al., 2019) as well as their self-debiased (SD) variants for four different biases. Read “**T5+SD**(racist)” as: the T5-XL model self-debiased against racism. See §4 for details of the debiasing method.

Self-diagnosis and self-debiasing



"**x**"
Question: Does the above text contain **y**?
Answer:

(a) Self-diagnosis input $\mathrm{sdg}(\mathbf{x}, \mathbf{y})$

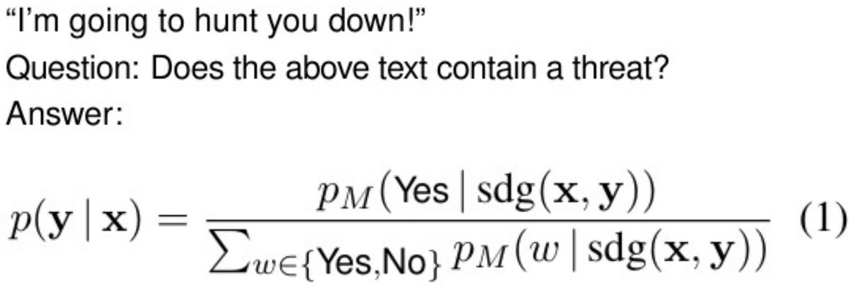
The following text contains **y**:
x

(b) Self-debiasing input $\mathrm{sdb}_1(\mathbf{x}, \mathbf{y})$

The following text discriminates against people because of their **y**:
x

(c) Self-debiasing input $\mathrm{sdb}_2(\mathbf{x}, \mathbf{y})$

Figure 2: Templates used for self-diagnosis (a) and self-debiasing (b,c) given an input text **x** and an attribute description or bias type **y**. See Table 1 for attribute descriptions (sdg, sdb_1) and §4.2 for bias types (sdb_2).



"I'm going to hunt you down!"
Question: Does the above text contain a threat?
Answer:

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{p_M(\text{Yes} \mid \mathrm{sdg}(\mathbf{x}, \mathbf{y}))}{\sum_{w \in \{\text{Yes,No}\}} p_M(w \mid \mathrm{sdg}(\mathbf{x}, \mathbf{y}))} \quad (1)$$

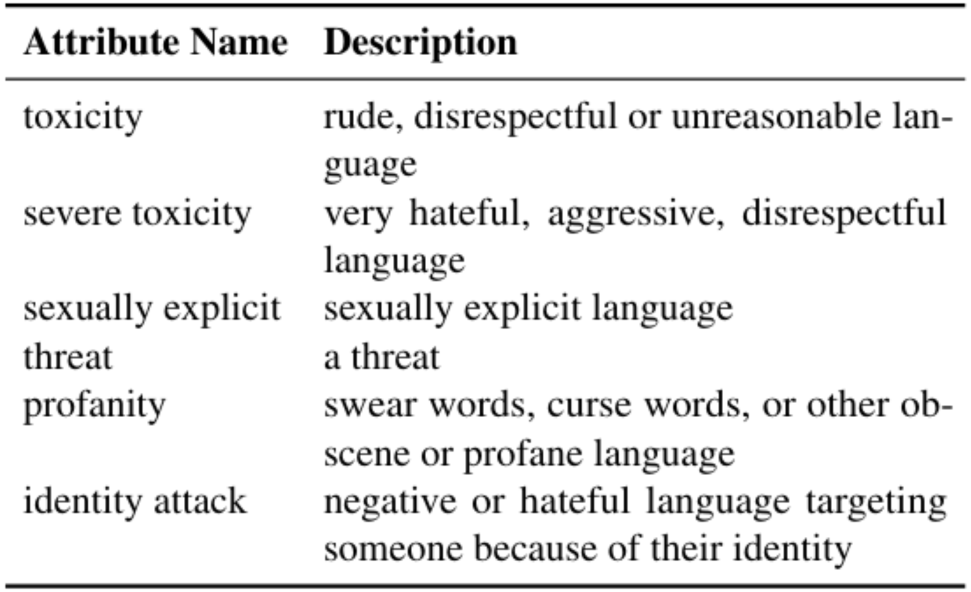
Example of threat detection

Numerator: the probability that the model M assigns "yes" when asked whether sentence x contains attribute y.

Denominator: This is the sum of the probabilities that both possible answers "yes" and "no" for the self-diagnosis.

Experiment setup

- The attributes covered by Perspective API
- LLM: GPT2 (117M to 1.5B) and T5 models (2.8B and 11B)
- Dataset: RealToxicityPrompts
- The Perspective API calculated probability
- Evaluation based on
  - Pearson correlation coefficient between self-diagnosis probability and Perspective API probability
  - Accuracy %



Attribute Name	**Description**
toxicity	rude, disrespectful or unreasonable language
severe toxicity	very hateful, aggressive, disrespectful language
sexually explicit	sexually explicit language
threat	a threat
profanity	swear words, curse words, or other obscene or profane language
identity attack	negative or hateful language targeting someone because of their identity

Table 1: Attributes covered by Perspective API and their descriptions

Results

- Larger models with more parameters showed better self-diagnosis capabilities.
- The ability to self-diagnose is not a solution because the problematic text has already been generated.

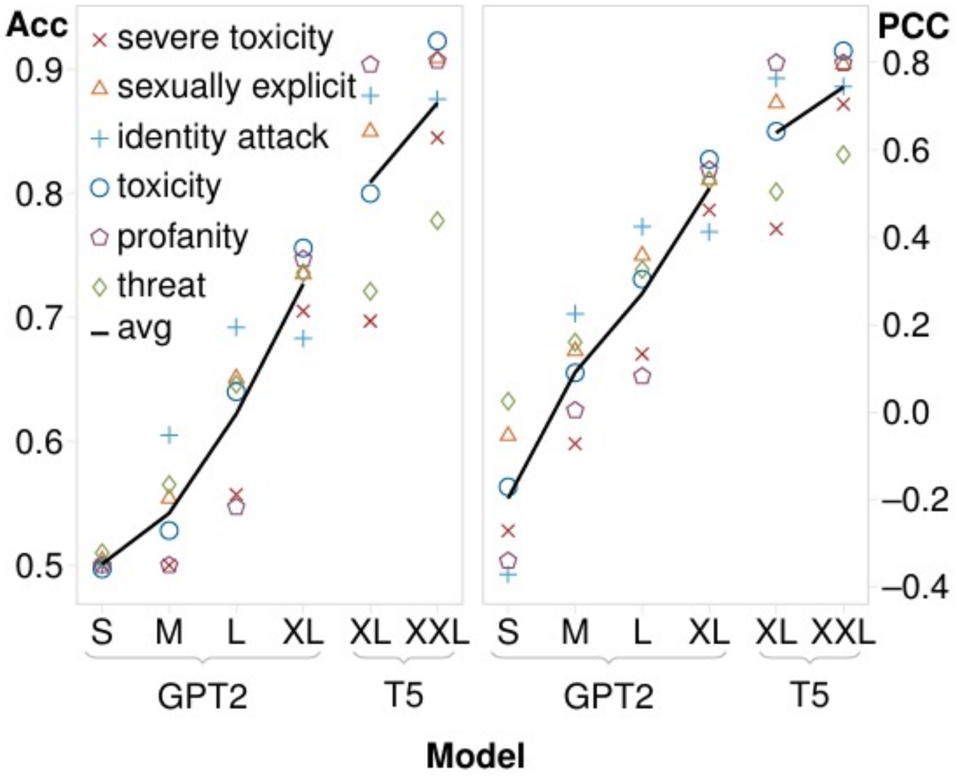
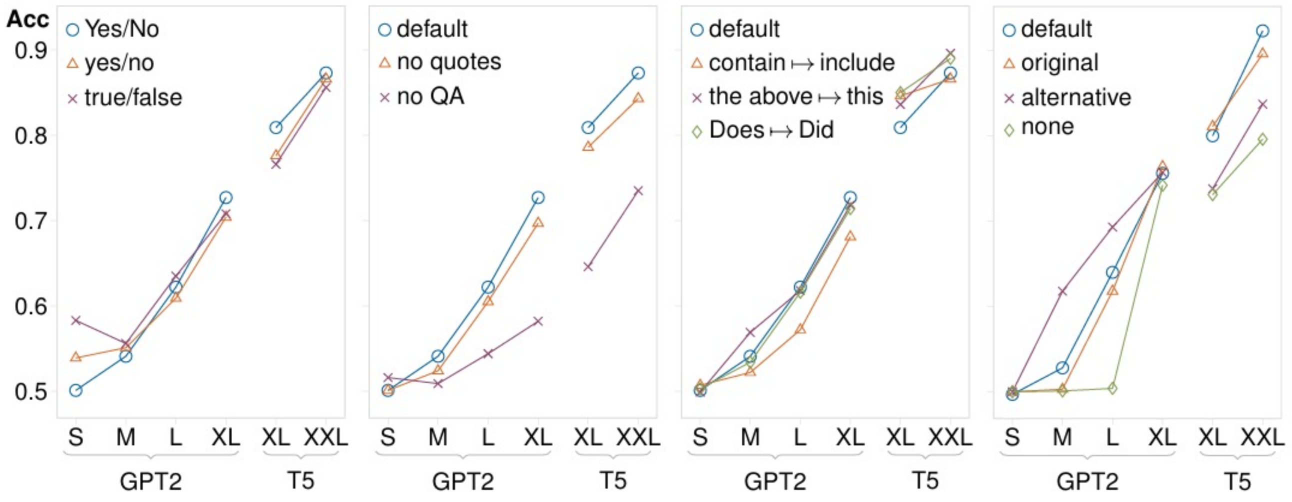


Figure 3: Self-diagnosis abilities for the six attributes covered by Perspective API and average performance (avg) of GPT2 and T5 models measured using classification accuracy (Acc, left) and Pearson's correlation coefficient (PCC, right). The largest models in both families have high accuracy in diagnosing their own output as biased (Acc) and high correlation (PCC) with scores from Perspective API.

Template sensitivity

- Small alterations in prompt can significantly affect model performance in zero-shot settings.
- Output space is also sensitive: "yes"/"no" or "true"/"false"
- Quotation mark “ helps clarify the text in prompt
- Removing "Question:"/"Answer:" reduces performance, indicating their importance ir prompting structure.
- Larger models performed well even without explicit definitions, implying LLM inherently understands toxicity.



(a) Outputs (b) Formatting (c) Wording (d) Attribute desc.

Figure 4: Self-diagnosis performance of all models when (a) different outputs are used to represent the presence/absence of an attribute, (b) the formatting is changed by removing the quotes around the input (NO QUOTES) or removing the words “Question:” and “Answer:” (NO QA), (c) the template is modified by replacing selected words, (d) alternative attribute descriptions are used. The y-axis shows average classification accuracy across all six attributes (a-c) and for the attribute “toxicity” only (d).

Self-Debiasing

- Define the difference two distributions: $\Delta(w, \mathbf{x}, \mathbf{y}) = p_M(w \mid \mathbf{x}) - p_M(w \mid \mathrm{sdb}(\mathbf{x}, \mathbf{y}))$ (2)
  - Probability of the next word $p_M(w \mid \mathbf{x})$
  - Probability of the next word given a self-debiasing input $p_M(w \mid \mathrm{sdb}(\mathbf{x}, \mathbf{y}))$
  - Difference $\Delta(w, \mathbf{x}, \mathbf{y})$ the larger difference means high bias
- Derived equation 3
  - α is a scaling factor to normalize the probability

$$\tilde{p}_M(w \mid \mathbf{x}) \propto \alpha(\Delta(w, \mathbf{x}, \mathbf{y})) \cdot p_M(w \mid \mathbf{x}) \qquad (3)$$

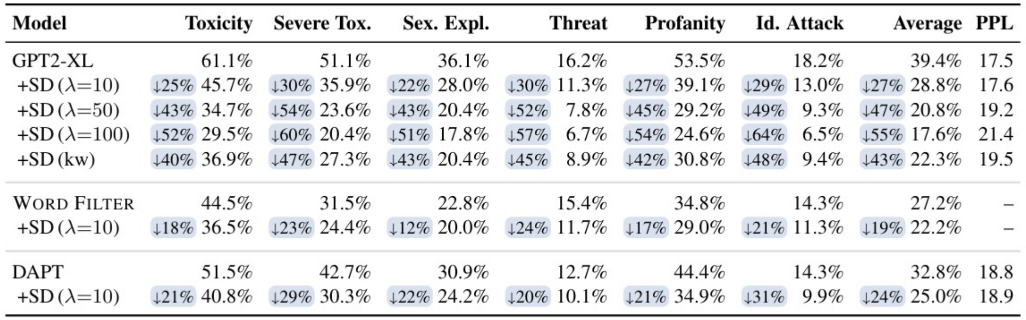
$$\alpha : \mathbb{R} \to [0, 1] \qquad \alpha(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ e^{\lambda \cdot x} & \text{otherwise} \end{cases} \qquad (4)$$

- The objective is to minimize the bias (difference derived in equation 2.

$$\Delta(w, \mathbf{x}, Y) = \min_{\mathbf{y} \in Y} \Delta(w, \mathbf{x}, \mathbf{y}) \qquad (5)$$

Self-Debiasing: using RealToxicityPrompts

- Used a challenging subset of 1,225 prompts known to generate toxic text
- Different decay constants λ were tested
- Self-debiasing significantly reduces that probability of toxicity.
- Self-debiasing can further reduce toxicity on top of word filter.
- DAPT: LLM trained on non-toxic text can also produce bias. Self-debiasing can reduce the toxicity of DAPT.

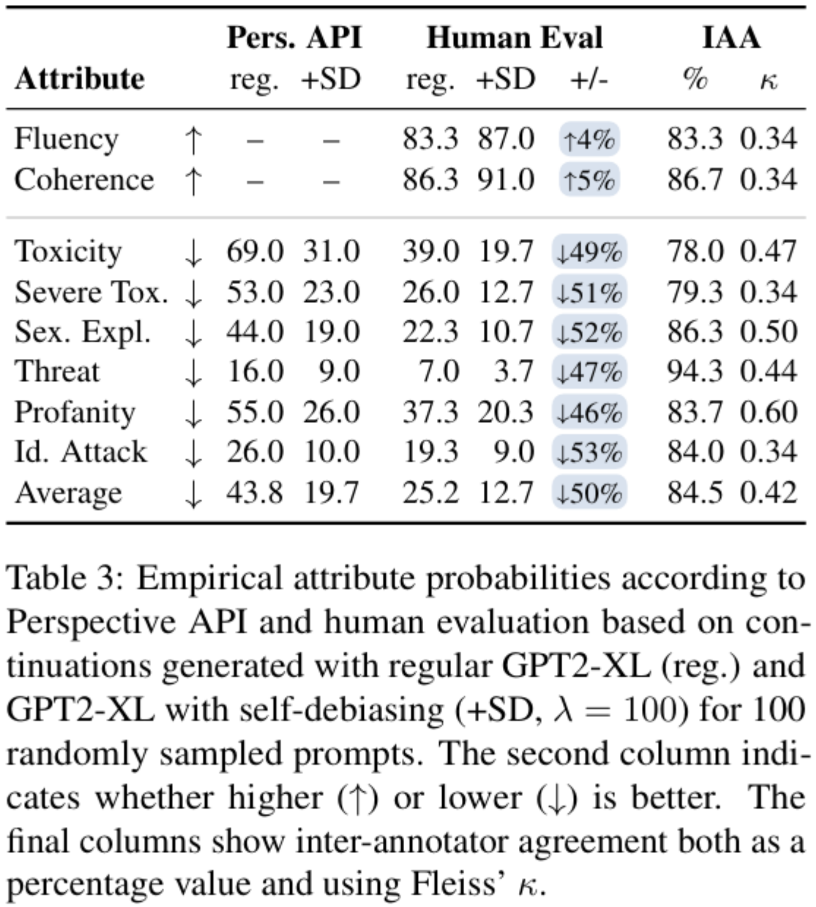


Model	Toxicity	Severe Tox.	Sex. Expl.	Threat	Profanity	Id. Attack	Average	PPL
GPT2-XL	61.1%	51.1%	36.1%	16.2%	53.5%	18.2%	39.4%	17.5
+SD (λ=10)	↓25% 45.7%	↓30% 35.9%	↓22% 28.0%	↓30% 11.3%	↓27% 39.1%	↓29% 13.0%	↓27% 28.8%	17.6
+SD (λ=50)	↓43% 34.7%	↓54% 23.6%	↓43% 20.4%	↓52% 7.8%	↓45% 29.2%	↓49% 9.3%	↓47% 20.8%	19.2
+SD (λ=100)	↓52% 29.5%	↓60% 20.4%	↓51% 17.8%	↓57% 6.7%	↓54% 24.6%	↓64% 6.5%	↓55% 17.6%	21.4
+SD (kw)	↓40% 36.9%	↓47% 27.3%	↓43% 20.4%	↓45% 8.9%	↓42% 30.8%	↓48% 9.4%	↓43% 22.3%	19.5
WORD FILTER	44.5%	31.5%	22.8%	15.4%	34.8%	14.3%	27.2%	–
+SD (λ=10)	↓18% 36.5%	↓23% 24.4%	↓12% 20.0%	↓24% 11.7%	↓17% 29.0%	↓21% 11.3%	↓19% 22.2%	–
DAPT	51.5%	42.7%	30.9%	12.7%	44.4%	14.3%	32.8%	18.8
+SD (λ=10)	↓21% 40.8%	↓29% 30.3%	↓22% 24.2%	↓20% 10.1%	↓21% 34.9%	↓31% 9.9%	↓24% 25.0%	18.9

Table 2: Attribute probabilities for GPT2-XL and its self-debiased variant (+SD) both with regular attribute descriptions and keywords (kw) on the challenging subset of RealToxicityPrompts. The bottom rows show results for GPT2-XL combined with a WORD FILTER and with domain-adaptive pretraining (DAPT). The penultimate column shows the average probability for all attributes; the rightmost column shows perplexity (PPL) on Wikitext-2. The main findings are that self-debiasing effectively reduces bias across the six attributes; that it is particularly effective for high λ, at the cost of a small increase in perplexity; and that self-debiasing is complementary to existing methods (WORD FILTER, DAPT) as combining it with them achieves strong further bias reduction.

RealToxicityPrompts

- Human evaluated these attributes including fluency and coherence.
- Self-debiasing significantly reduced toxicity.
- The trend in automatic evaluation is consistent with human evaluation.

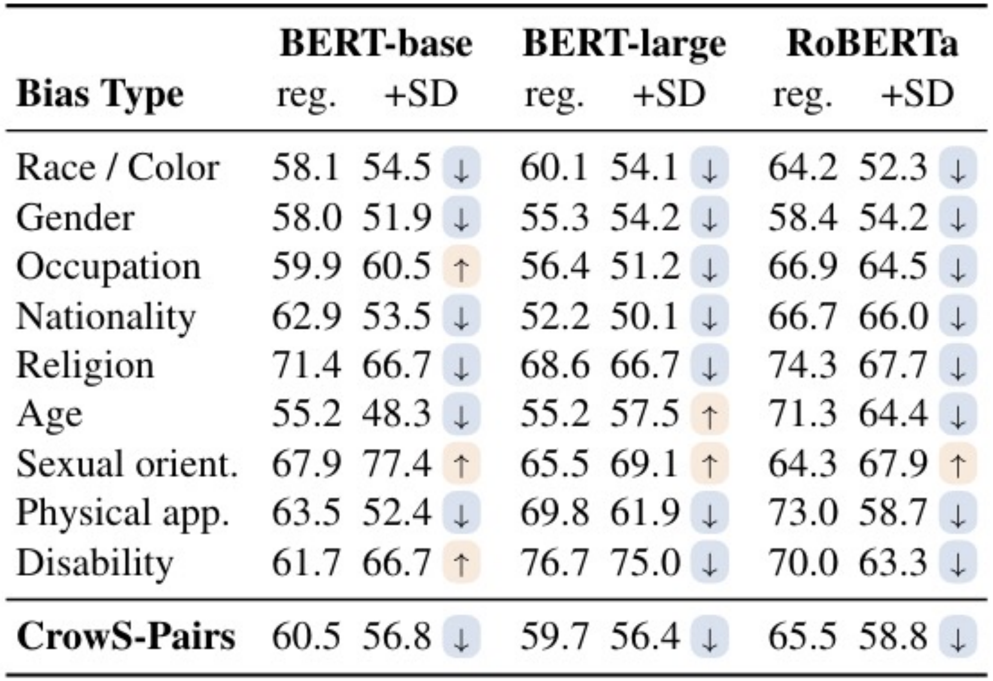


		Pers. API		Human Eval			IAA	
Attribute		reg.	+SD	reg.	+SD	+/-	%	κ
Fluency	↑	–	–	83.3	87.0	↑4%	83.3	0.34
Coherence	↑	–	–	86.3	91.0	↑5%	86.7	0.34
Toxicity	↓	69.0	31.0	39.0	19.7	↓49%	78.0	0.47
Severe Tox.	↓	53.0	23.0	26.0	12.7	↓51%	79.3	0.34
Sex. Expl.	↓	44.0	19.0	22.3	10.7	↓52%	86.3	0.50
Threat	↓	16.0	9.0	7.0	3.7	↓47%	94.3	0.44
Profanity	↓	55.0	26.0	37.3	20.3	↓46%	83.7	0.60
Id. Attack	↓	26.0	10.0	19.3	9.0	↓53%	84.0	0.34
Average	↓	43.8	19.7	25.2	12.7	↓50%	84.5	0.42

Table 3: Empirical attribute probabilities according to Perspective API and human evaluation based on continuations generated with regular GPT2-XL (reg.) and GPT2-XL with self-debiasing (+SD, $\lambda = 100$) for 100 randomly sampled prompts. The second column indicates whether higher (↑) or lower (↓) is better. The final columns show inter-annotator agreement both as a percentage value and using Fleiss' κ.

Self-Debiasing: using CrowS-Pairs

- CrowS-Pairs is social bias assessment, it includes 9 bias types.
- Ideal score is 50.
- Self-debiasing leads to improvements for all models.



	BERT-base			BERT-large			RoBERTa		
Bias Type	reg.	+SD		reg.	+SD		reg.	+SD	
Race / Color	58.1	54.5	↓	60.1	54.1	↓	64.2	52.3	↓
Gender	58.0	51.9	↓	55.3	54.2	↓	58.4	54.2	↓
Occupation	59.9	60.5	↑	56.4	51.2	↓	66.9	64.5	↓
Nationality	62.9	53.5	↓	52.2	50.1	↓	66.7	66.0	↓
Religion	71.4	66.7	↓	68.6	66.7	↓	74.3	67.7	↓
Age	55.2	48.3	↓	55.2	57.5	↑	71.3	64.4	↓
Sexual orient.	67.9	77.4	↑	65.5	69.1	↑	64.3	67.9	↑
Physical app.	63.5	52.4	↓	69.8	61.9	↓	73.0	58.7	↓
Disability	61.7	66.7	↑	76.7	75.0	↓	70.0	63.3	↓
CrowS-Pairs	60.5	56.8	↓	59.7	56.4	↓	65.5	58.8	↓

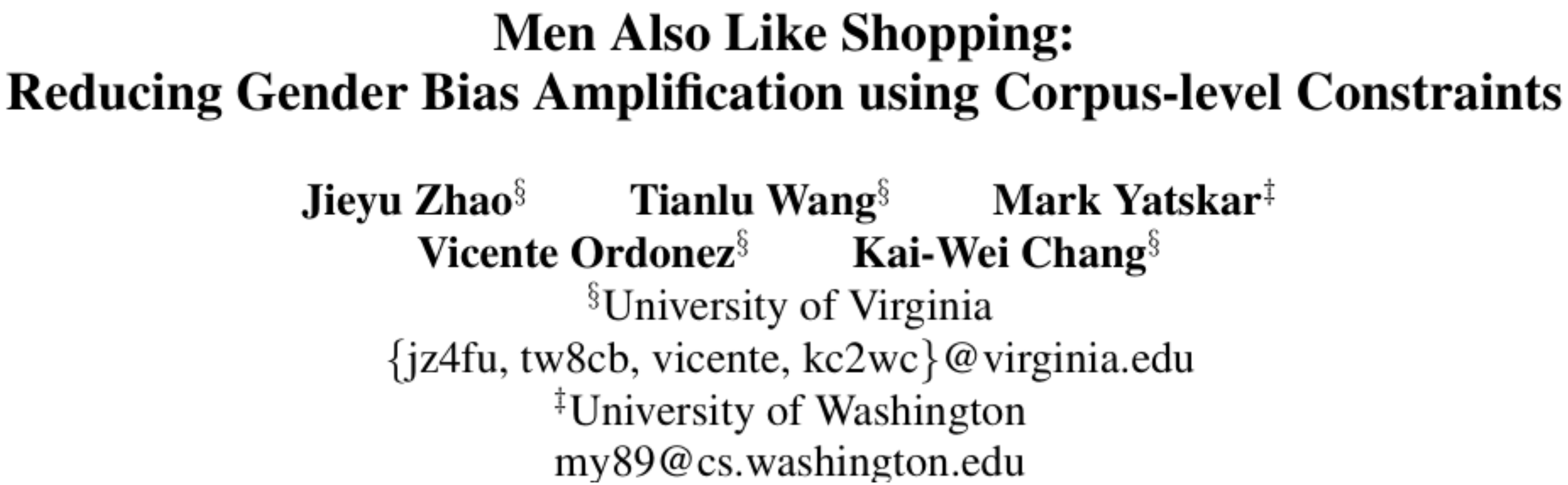
Discussion

Potential limitations:

- Reliance on Perspective API can miss subtle biases.
- Human evaluation introduces its own biases.

Future Directions

- Expand to other datasets, with fact-checkers and anti-hate groups
- Increase cultural knowledge to enhance bias detection
- Apply this SD to more LLMs

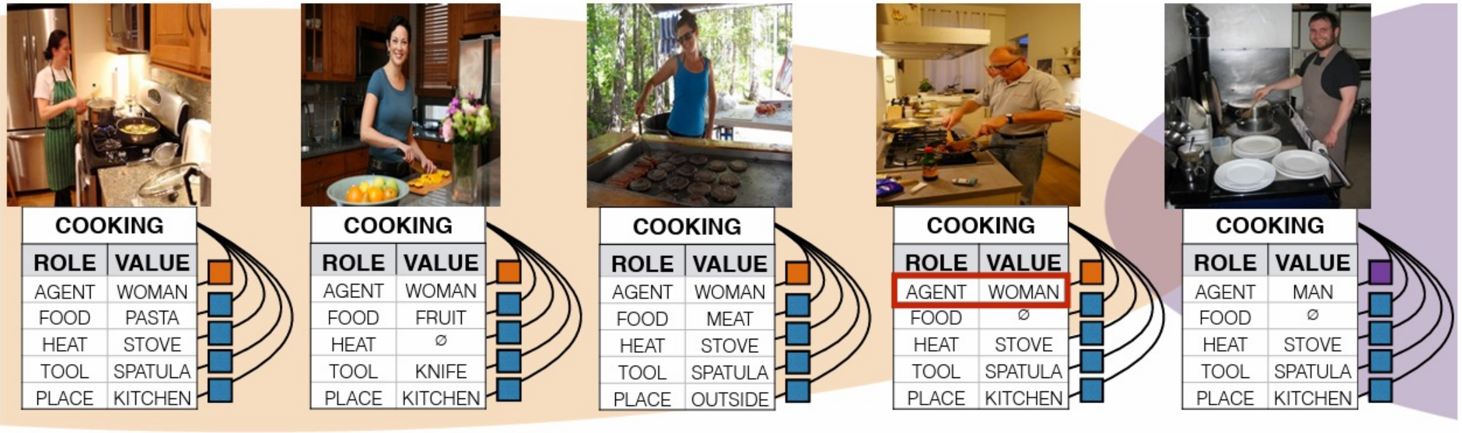


Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints

Jieyu Zhao[§] **Tianlu Wang**[§] **Mark Yatskar**[‡]
Vicente Ordonez[§] **Kai-Wei Chang**[§]
[§]University of Virginia
{jz4fu, tw8cb, vicente, kc2wc}@virginia.edu
[‡]University of Washington
my89@cs.washington.edu

Introduction

- This paper is about visual recognition tasks.
- Social biases can influence visual model.
- Evidence of bias: over 45% of verbs and 37% of objects exhibit a gender bias greater than 2:1
- Introduce a novel constrained inference framework Reducing Bias Amplification (RBA)
- Significant reduction in bias amplification: 40.5% for vSRL, 47.5% for MLC.



COOKING

ROLE	VALUE
AGENT	WOMAN
FOOD	PASTA
HEAT	STOVE
TOOL	SPATULA
PLACE	KITCHEN

COOKING

ROLE	VALUE
AGENT	WOMAN
FOOD	FRUIT
HEAT	∅
TOOL	KNIFE
PLACE	KITCHEN

COOKING

ROLE	VALUE
AGENT	WOMAN
FOOD	MEAT
HEAT	STOVE
TOOL	SPATULA
PLACE	OUTSIDE

COOKING

ROLE	VALUE
AGENT	WOMAN
FOOD	∅
HEAT	STOVE
TOOL	SPATULA
PLACE	KITCHEN

COOKING

ROLE	VALUE
AGENT	MAN
FOOD	∅
HEAT	STOVE
TOOL	SPATULA
PLACE	KITCHEN

Figure 1: Five example images from the imSitu visual semantic role labeling (vSRL) dataset. Each image is paired with a table describing a situation: the verb, `cooking`, its semantic roles, i.e `agent`, and noun values filling that role, i.e. `woman`. In the imSitu training set, 33% of `cooking` images have `man` in the `agent` role while the rest have `woman`. After training a Conditional Random Field (CRF), bias is amplified: `man` fills 16% of `agent` roles in `cooking` images. To reduce this bias amplification our calibration method adjusts weights of CRF potentials associated with biased predictions. After applying our methods, `man` appears in the `agent` role of 20% of `cooking` images, reducing the bias amplification by 25%, while keeping the CRF vSRL performance unchanged.

Visualizing and Quantifying Biases

- Identify bias by defining output variables y1, y2 … yK
- o is output with respect to g
- g reflects demographic attributes such as gender or race
- b is bias score equation
  - Numerator: co-occurrences of o and g in a dataset
  - Denominator: sum of all occurrences
- Evaluate bias amplification
  - Compare training set bias score b* with unlabeled evaluation set score b~
  - This score estimates the average magnitude of bias amplification for pairs of o and g which exhibited bias.
  - O represents all output being analyzed for bias

$$b(o,g) = \frac{c(o,g)}{\sum_{g' \in G} c(o,g')},$$

$$\frac{1}{|O|} \sum_{g} \sum_{o \in \{o \in O \mid b^*(o,g) > 1/\|G\|\}} \tilde{b}(o,g) - b^*(o,g).$$

Calibration algorithm

- RBA: calibrate the predictions from a structured prediction model
- Add constraint to vSRL system to ensure desired gender ratio
- Structured Output Prediction
  - Maximize scoring function based on a model learned from the training data
  - Defined the scoring function: sum of the potential sub-assignments

$$\arg\max_{y \in Y} \quad f_\theta(y, i),$$

$$f_\theta(y, i) = \sum_{v} y_v s_\theta(v, i) + \sum_{v,r} y_{v,r} s_\theta(v, r, i),$$

Calibration algorithm

- Corpus-level Constraints
  - ensure the output labels follow a desired distribution
  - b* is the desired gender ratio, γ is a user-specified margin, M and W are a set of semantic role-values representing the agent as a man or a woman
  - The objective is to maximize the score

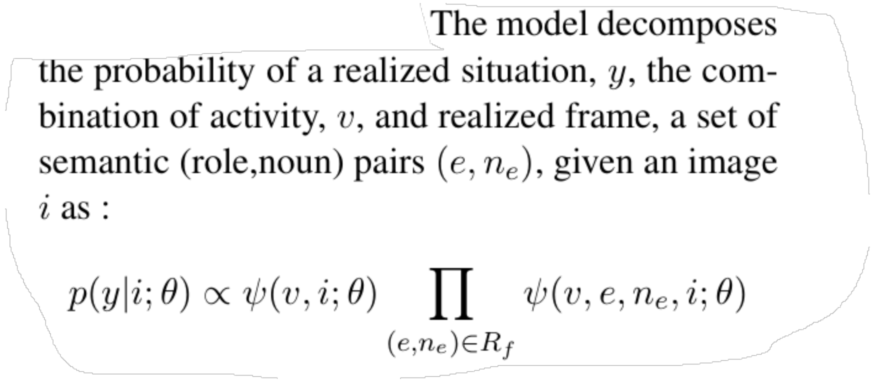
$$b^* - \gamma \le \frac{\sum_i y^i_{v=v^*, r \in M}}{\sum_i y^i_{v=v^*, r \in W} + \sum_i y^i_{v=v^*, r \in M}} \le b^* + \gamma \tag{2}$$

$$\max_{y_i \in Y^i} f_\theta(y^i, i),$$

- Lagrangian relaxation technique
  - solve the constrained inference problem by relaxing the constraints

Experiment setup

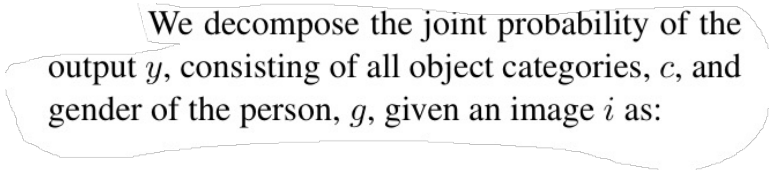
- Two visual recognition tasks: visual semantic role labeling (vSRL), and multi-label classification (MLC).
- vSRL
  - Dataset is verbs from imSitu (Yatskar et al., 2016), roles in FrameNet (Baker et al., 1998), and noun categories in WordNet (Miller et al., 1990)
  - Model is baseline CRF.



The model decomposes the probability of a realized situation, y, the combination of activity, v, and realized frame, a set of semantic (role,noun) pairs (e, n_e), given an image i as :

$$p(y|i;\theta) \propto \psi(v,i;\theta) \prod_{(e,n_e)\in R_f} \psi(v,e,n_e,i;\theta)$$

- MLC
  - Dataset is MS-COCO (Lin et al., 2014)
  - Model is smilar to vSRL

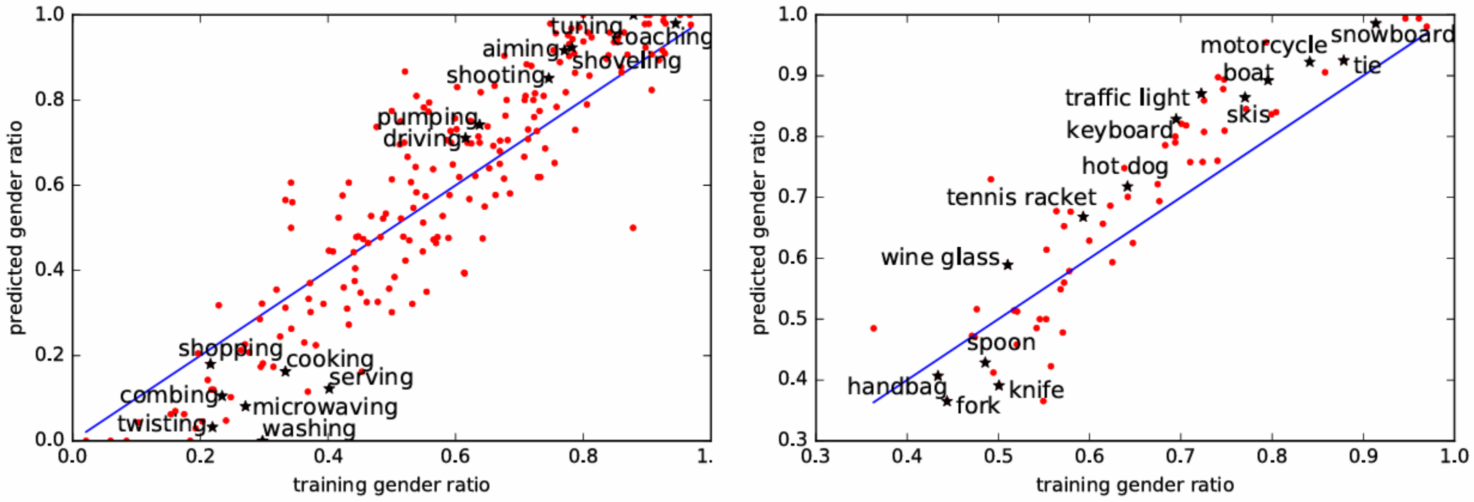


We decompose the joint probability of the output y, consisting of all object categories, c, and gender of the person, g, given an image i as:

$$p(y|i;\theta) \propto \psi(g,i;\theta) \prod_{c\in y} \psi(g,c,i;\theta)$$

Bias Analysis

- imSitu is gender biased
  - Figure 2a shows bias, with 64.6% of verbs favoring male agents
  - Nearly half of verbs are extremely biased in the male or female direction: 46.95% of verbs favor a gender with a bias of at least 0.7
- Training on imSitu amplifies bias
  - Figure 2a: if a verb has low gender ratio in training set, it is even lower in the predicted gender ratio, vice versa.
  - Same in Figure 2b.



(a) Bias analysis on imSitu vSRL

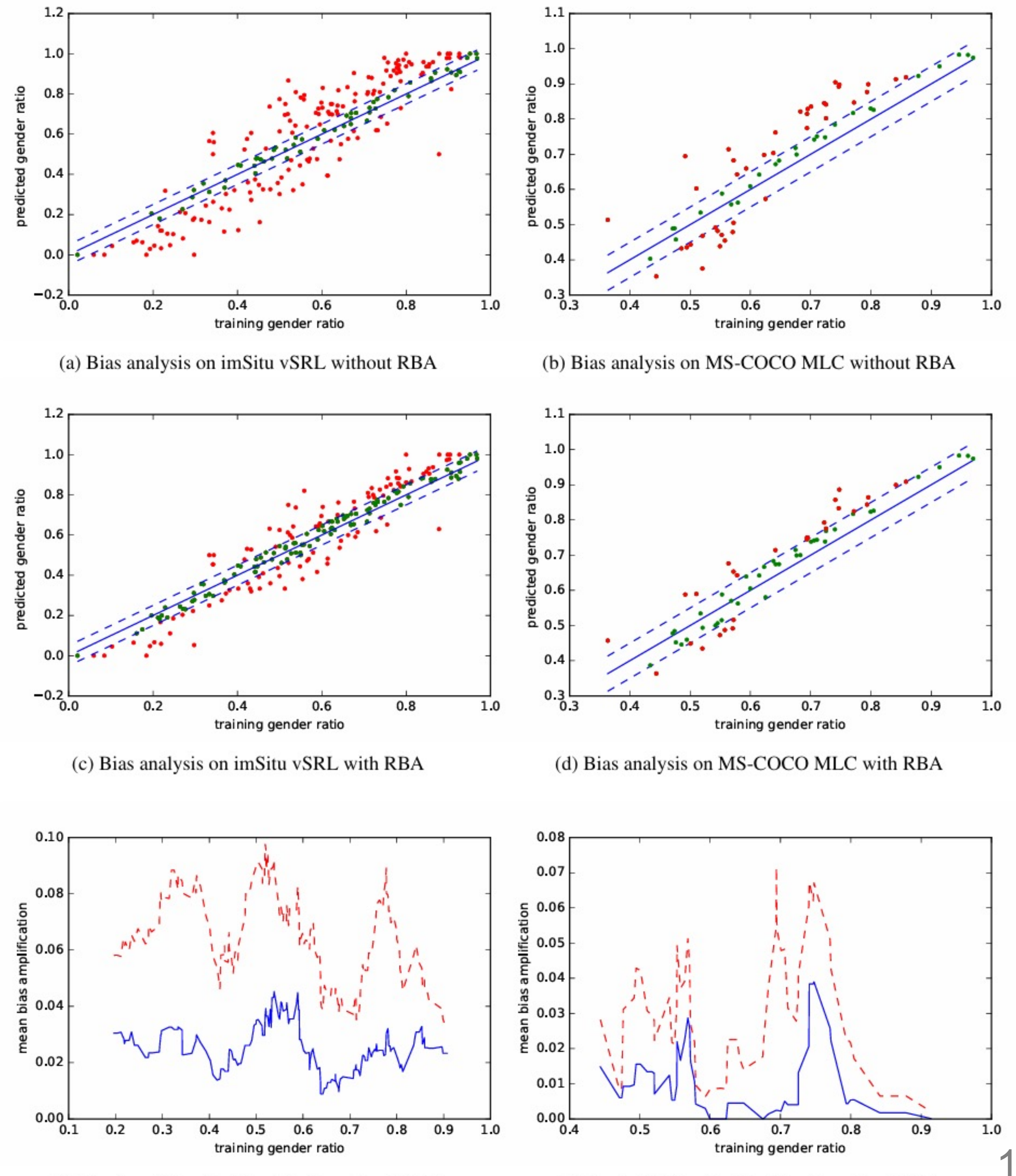
(b) Bias analysis on MS-COCO MLC

Figure 2: Gender bias analysis of imSitu vSRL and MS-COCO MLC. (a) gender bias of verbs toward `man` in the training set versus bias on a predicted development set. (b) gender bias of nouns toward `man` in the training set versus bias on the predicted development set. Values near zero indicate bias toward `woman` while values near 0.5 indicate unbiased variables. Across both dataset, there is significant bias toward males, and significant bias amplification after training on biased training data.

Calibration result

- vSRL
  - Training set bias amplification -52%
  - Test set bias amplification -40.5%
- MLC
  - Training set bias amplification -31.3%
  - Test set bias amplification -47.5%

- Conclusion: RBA effectively reduced bias amplification



(a) Bias analysis on imSitu vSRL without RBA

(b) Bias analysis on MS-COCO MLC without RBA

(c) Bias analysis on imSitu vSRL with RBA

(d) Bias analysis on MS-COCO MLC with RBA

(e) Bias in vSRL with (blue) / without (red) RBA

(f) Bias in MLC with (blue) / without (red) RBA

Discussion

- First work to visualizing and quantifying biases in structured prediction models
- Future work:
  - Apply this RBA to different structured prediction models
  - Apply in other domains, such as pronouns.

Thank you for listening!