ethics in NLP

CS 585, Fall 2018

Introduction to Natural Language Processing

http://people.cs.umass.edu/~miyyer/cs585/

Mohit Iyyer

College of Information and Computer Sciences

University of Massachusetts Amherst

many slides from Yulia Tsvetkov

your remaining to-dos:

- posters due **Dec 6** (instructions for submission on Piazza)
- HW3 due **Dec 7**
  - remember, everyone gets 3 late days for homeworks, so if you haven't used yours yet then you may want to :)
- final presentations **Dec 11** in CS 150/151
- final reports due **Dec 20** on Gradescope / Moodle

Song Genre Classification

too much text!

Introduction

Our project objective was to run various natural language processing classification algorithms on a dataset of songs to compare the effectiveness of these algorithms in identifying the genre of the songs.
We used a bag of words representation of the song lyrics linked to ground truth genre tags to train the algorithms and then predict genres for new sets of lyrics.

Dataset information

Our dataset contains 13 genres with a distribution of:

Pop_Rock 75.15%
Reggae 0.70%
Country 4.00%
Jazz 0.50%
Vocal 1.06%
New Age 0.16%
Latin 4.30%
Rap 4.06%
RnB 3.93%
International 1.78%
Blues 0.57%
Electronic 2.78%
Folk 1.00%

- dataset is a BOW representation of the stemmed lyrics
- Derived from Million Songs Dataset
- Split 90-10 training vs test
- 114,643 songs in the dataset

Approach

We were unable to find a dataset that linked lyrics directly to genre, so we first had to compile information from multiple datasets into one that we could use. The musiXmatch dataset maps songs to lyrics while the MSD Allmusic Top Genre Dataset maps songs to genre, creating the perfect combination for what our project needed. Once we had our data, we began implementing different natural language processing algorithms using python's scikit-learn library. After training these algorithms on a large percentage of our dataset and testing their ability to correctly classify the remaining portion, we were able to identify which type of algorithm generally

Results

- **Decision Tree Algorithm**: 70.06% accuracy
- **Multi-Layer Perception**: 76.45% accuracy
- **Stochastic Gradient Descent (SGD)**: 76.16% accuracy
- **Support Vector Machine Classifier (SVM)**: 75.22% accuracy
- **Voting Classifier**: 78.51% accuracy

The Voting Classifier used the other algorithms and implemented a voting system such that each classifier had a say in the genre assigned to a given example. This turned out to get a small boost in accuracy over the other classifiers as it could weed out any outliers when one of the algorithms predicted the wrong result.

The Multi-Layer Perceptron and SGD classifiers performed a bit better than the others

Conclusions

- We were unable to use many of the more "advanced" algorithms on our dataset due to its limitations as a pre-stemmed/lemmatized BOW representation of the lyrics.
- Given more time/resources it probably be possible to compile a "better" dataset which we could run algorithms that would obtain higher accuracy.

References

https://labrosa.ee.columbia.edu/millionsong/

http://web.stanford.edu/class/cs224n/reports/2728368.pdf

https://nlp.stanford.edu/courses/cs224n/2006/fp/sadovsky-x1n9-1-224n_final_report.pdf

too much text!

Twitter Sentiment Classification and Analysis

Purpose

The purpose of this project is to predict the sentiment of a tweet based on a 5-point scale (from very negative to very positive) and compare the sentiment of the topic of a tweet among various demographics through graphs.

We have previously classified text sentiment based on a two-point scale (negative versus positive) in class, so this project is meant to push the boundaries. Because the source of the tweet data also provides user demographic data, it seemed interesting to visually analyze sentiment trends based on a user's location.

Data and Tools

The SemEval-2017 Task 4 Data and Tools page provided all of the needed materials for obtaining the data for this project. This data included training, development, and testing sets for tweets written in English, as well as information about the users who wrote the tweets. For reading and parsing reasons, the data needed to be cleaned using a script.

Tools used:

- Python 2.7.13
- Natural Language Toolkit (NLTK)
- Matplotlib

Method & Results

Of the many ways to classify sentiment, the first attempted for this project was the Naïve Bayes, bag-of-words method, where the tweets are tokenized and evaluated based on each individual token. The classifier is trained on the tokens stored in each sentiment dictionary (one for each rating on the scale) based on the provided sentiment of the tweets in the training data.

I additionally attempted to include an external dictionary with generally known words and their sentiment weights to add to the weights calculated during the classifier training. When comparing the two implementations, the external dictionary proved to hurt rather than help the classification accuracy.

While the classification accuracy remained above 50% on all data sets, this method proved inefficient compared to others learned in class.

Graphs

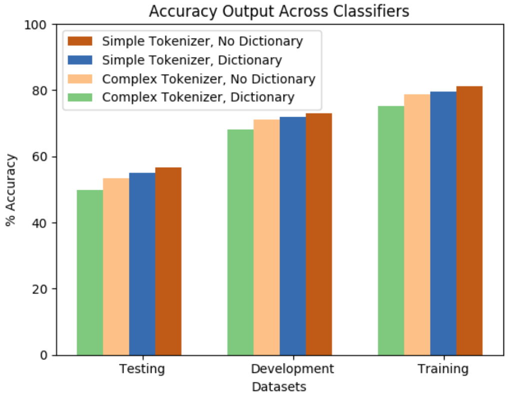


Fig 1: Multi-bar chart to compare accuracy outputs across classifier implementations on different datasets

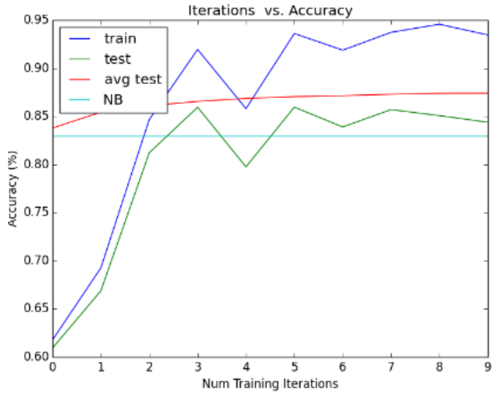


Fig 2: Line chart taken from the solutions of a previous homework displays the anticipated accuracy of the perceptron implementation

Future Work

Because the bag-of-words method was found to be inefficient, I am currently working on implementing a classification perceptron method to replace it, since it proved to have a much higher accuracy when compared to the Naïve Bayes method.

In addition, graphs displaying the sentiment among users from different location have yet to be created. There will be two types of graphs: the first will show the sentiment across a single group on a single topic, and the second will compare the general sentiment (if there is a clear one) of two different groups on a single topic.

References

- Farra, N., Nakov, P., & Rosenthal, S. (2016). *SemEval-2017 Task 4: Sentiment Analysis in Twitter*, SIGLEX. Retrieved from alt.qcri.org/semeval2017/task4/
- Taboada, M., Brooke, J., Voll, K., Anthony, C., & Grieve, J. (2009). SO-CAL (Version 1.11). github.com/DrOttensooser/BiblicalNLPworks/tree/master/SkyDrive/NLP/CommonWorks/Data/Opion-Lexicon-English/SO-CAL

Price Prediction of Alternative Cryptocurrencies using Telegram Group Chats

could have less text, overall not bad!

Overview

This project uses existing sentiment analysis and machine learning techniques to anticipate price movements of alternative cryptocurrencies using popular Telegram chat groups. Telegram is a popular chat application that has been adopted by cryptocurrency communities for price speculation, and as an interface between project teams and the community. Since Bitcoin is the de facto bridge between fiat and all other cryptocurrencies, backtesting against the market will be evaluated according to maximization of a simulated Bitcoin account.

Datasets

Coin	Ticker	Telegram Chat	Members	Msg / Hour
Litecoin	LTC	Litecoin LTC	8535	36.5
XEM	XEM	NEMberia 2.0	1768	17.7
Ethereum	ETH	EthTrader	5046	14.3

Sentiment Lexicon

A random subset of messages in Litecoin LTC was manually annotated as displaying strong positive or negative indications of sentiment or outlook regarding price. Using the results of annotations, a custom lexicon was developed by hand using the keywords found with sentiment weights. This lexicon used generic keywords allowing it to be reused for other cryptocurrencies. Cryptocurrency slang (e.g. 'mooning'), trading terminology ('long', 'short'), and common slang ('rekt') were incorporated into the lexicon, in addition to words in the existing VADER lexicon.

Relevance Annotations

Messages were manually annotated according to perceived relevance to the coin or its market behavior.

Chat	Annotations	Relevant	Irrelevant
Litecoin LTC	3207	1248	1959
NEMberia 2.0	3295	875	2422
EthTrader	2682	474	2208

Relevance Classifier (Neural Net)

A multi-level perceptron classifier with a single hidden layer of size 50 was trained on single word ngrams of the training set's annotated data with 10,000 iterations. Train/Test split was done on 07/01/2017.

	Train Size	Dev Size	Precision	Recall	F1
LTC	2511	694	.70	.69	.68
XEM	2425	721	.78	.80	.78
ETH	1860	816	.80	.81	.81

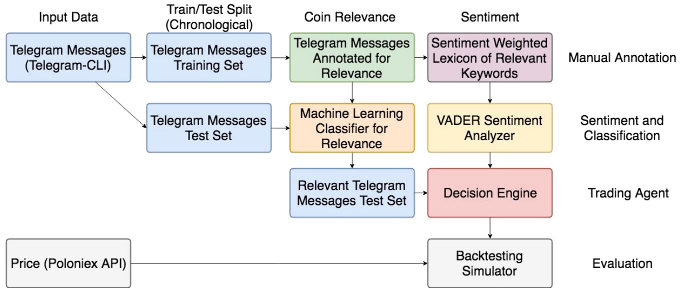
VADER Sentiment Analysis and Granger Causality

Granger causality was calculated based on VADER sentiment and price, using custom and stock lexicons. This established correlation between the price and sentiment time series expressed with both lexicons.

	Max Time Lag w/ p value > .05	
	Stock Lexicon	Custom Lexicon
LTC	>15h	>15h
XEM	8h	9.5h
ETH	8.5h	8h

Trading Algorithm

Sentiment was calculated for each 60 minute group of messages, and a exponential weighted moving average (EWMA) of sentiment, and deviation is maintained. When sentiment rises or drops above the EWMA of sentiment past a deviation threshold, a percentage of the altcoin account proportional to the difference between sentiment and sentiment EWMA is transferred to the altcoin, or Bitcoin account, respectively.

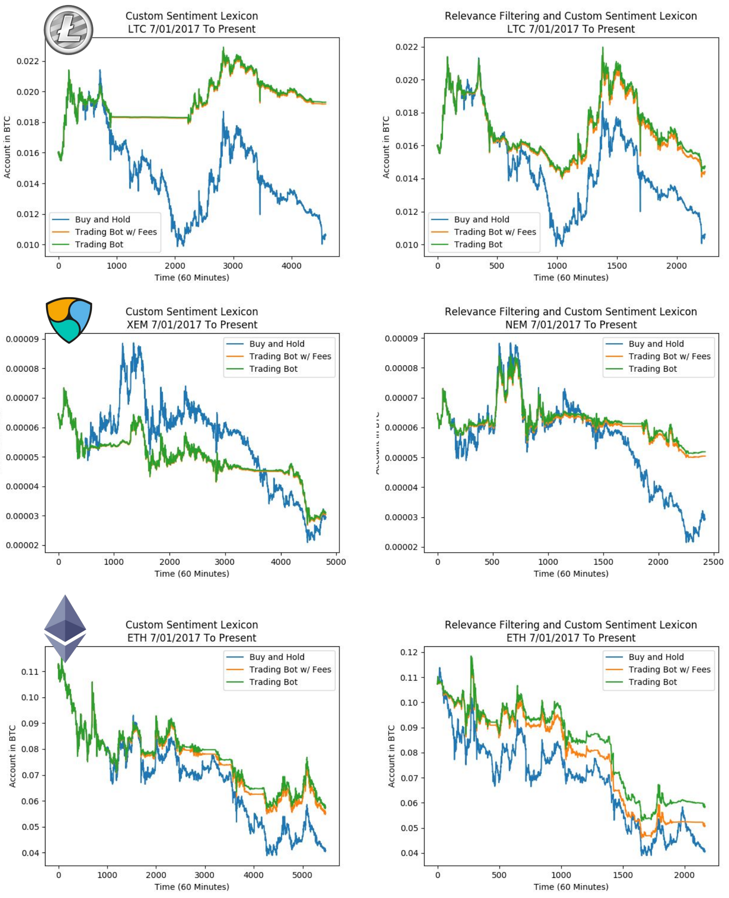


Input Data
Train/Test Split (Chronological)
Coin Relevance
Sentiment
Telegram Messages (Telegram-CLI)
Telegram Messages Training Set
Telegram Messages Annotated for Relevance
Sentiment Weighted Lexicon of Relevant Keywords
Manual Annotation
Telegram Messages Test Set
Machine Learning Classifier for Relevance
VADER Sentiment Analyzer
Sentiment and Classification
Relevant Telegram Messages Test Set
Decision Engine
Trading Agent
Price (Poloniex API)
Backtesting Simulator
Evaluation

Evaluation

The trading agent with relevance filtering outperformed buy and hold both with, and without a standard .25% transaction fee for each order made.

Without Relevance Filtering **With Relevance Filtering**



pretty good!

Aspect Extraction using Dependency Parsing and Semantic Clustering

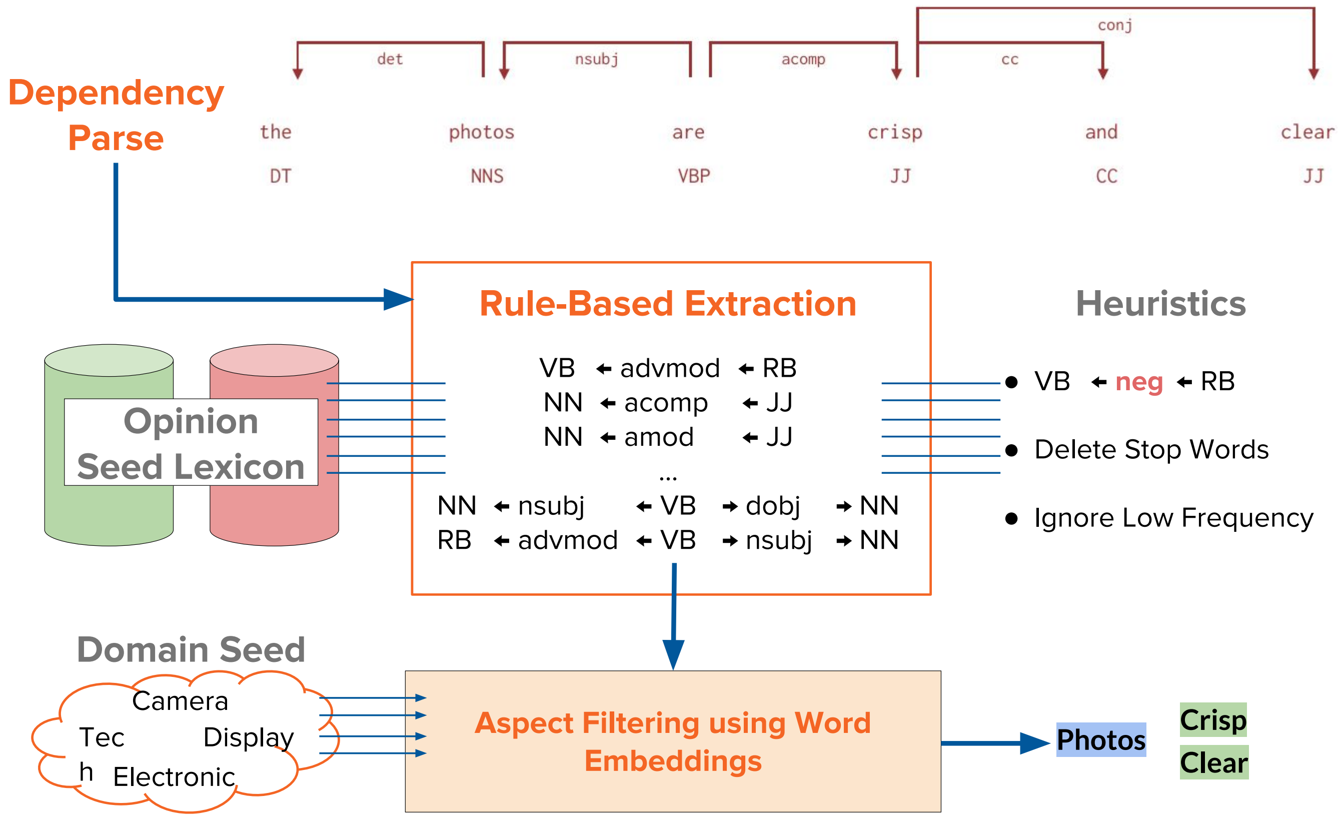
Problem Description

" it gives **great** **pictures**,
the **controls** are **easy to use**,
the **battery** **lasts forever** on one single charge,
but the **software** is **not user-friendly** at all! "

Pictures	great
Controls	easy to use,
Battery	lasts forever
Software	not user-friendly

📷 Pictures
Controls
Battery
Software

Procedural Steps



Dependency Parse

Rule-Based Extraction

VB ← advmod ← RB
NN ← acomp ← JJ
NN ← amod ← JJ
...
NN ← nsubj ← VB → dobj → NN
RB ← advmod ← VB → nsubj → NN

Opinion Seed Lexicon

Heuristics

- VB ← neg ← RB
- Delete Stop Words
- Ignore Low Frequency

Domain Seed

Camera
Tec h
Display
Electronic

Aspect Filtering using Word Embeddings

Photos
Crisp
Clear

Results

	Aspect Precision	Aspect Recall	Opinion Precision
DVD Player	0.316	0.201	0.492
Camera-1	0.347	0.487	0.596
Camera-2	0.516	0.534	0.341
MP3 Player	0.360	0.411	0.571
Cell Phone	0.545	0.525	0.478
OVERALL	**0.385**	**0.384**	**0.504**

Further Work

- More Heuristics
- Recursive Seed Expansion
- Better Semantic Clustering

- *Hu and Liu. 2004. **Mining and summarizing customer reviews**, 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*
- *Qiu, Liu, Bu, and Chen. 2011. **Opinion word expansion and target extraction through double propagation.** Computational Linguistics*

great!

Wordify: A Reverse Dictionary for Everyone

COMPSCI 585
Introduction to Natural Language Processing

Task

Ever had a word at the tip of your tongue and still be unable to speak or write it?

Using a Reverse Dictionary, you can turn your thoughts into words!

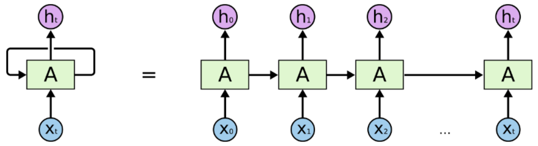
Aim: Develop a reverse dictionary by learning to map the definitions in a dictionary to the word embeddings of the words that they define.

A native of a cold country - **eskimo**
A way of moving through the air - **glide**

Approach

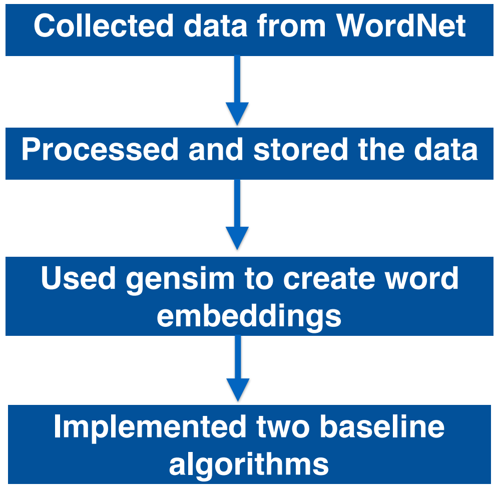
Step 1 Learn word embeddings using Word2Vec

Step 2 Train a RNN to map the sentence or phrase to the word embedding of the word that it defines



Step 3 Map the input phrase to a point in the embedding space and return the words closest to that point

Progress so far….

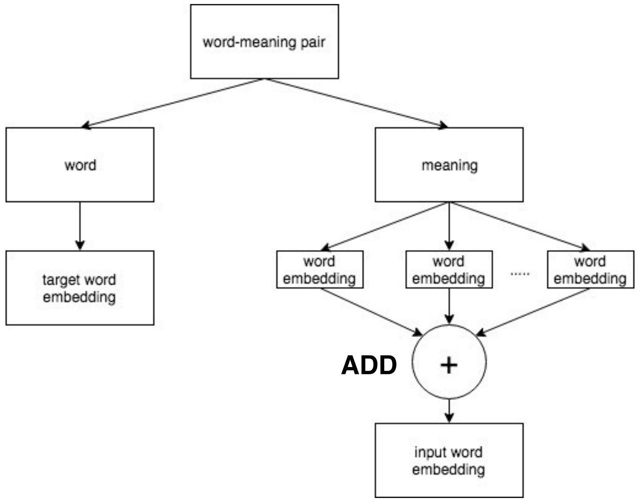


Collected data from WordNet

Processed and stored the data

Used gensim to create word embeddings

Implemented two baseline algorithms



Preliminary Results

Baseline algorithm	Mean Rank	%acc@500/1k/5k/10k	%match
ADD	29912	1.7/5.1/8.5/16.2	48
MUL	62601	0.0/1.7/4.2/5.9	49

Future Work

- Use pre-trained word embeddings from spaCy to improve the baseline performance.
- Implement a RNN model to learn the word embeddings and compare the performance with respect to the baseline methods.

I. Felix Hill, Kyunghyun Cho, Anna Korhonen and Yoshua Bengio. Learning to understand phrases by embedding the dictionary. *Association for Computational Linguistics*, vol 4, 2016.
II. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

what are we talking about today?

- many NLP systems affect actual people
  - systems that interact with people (conversational agents)
  - perform some reasoning over people (e.g., recommendation systems, targeted ads)
  - make decisions about people’s lives (e.g., parole decisions, employment, immigration)

- questions of *ethics* arise in all of these applications!

why are we talking about it?

- the explosion of data, in particular user-generated data (e.g., social media)

- machine learning models that leverage huge amounts of this data to solve certain tasks

Language and People

The common misconception is that language has to do with **words** and what they mean.

It doesn't.

It has to do with **people** and what ***they*** mean.

Dan Jurafsky's keynote talks at CVPR'17 and EMNLP'17

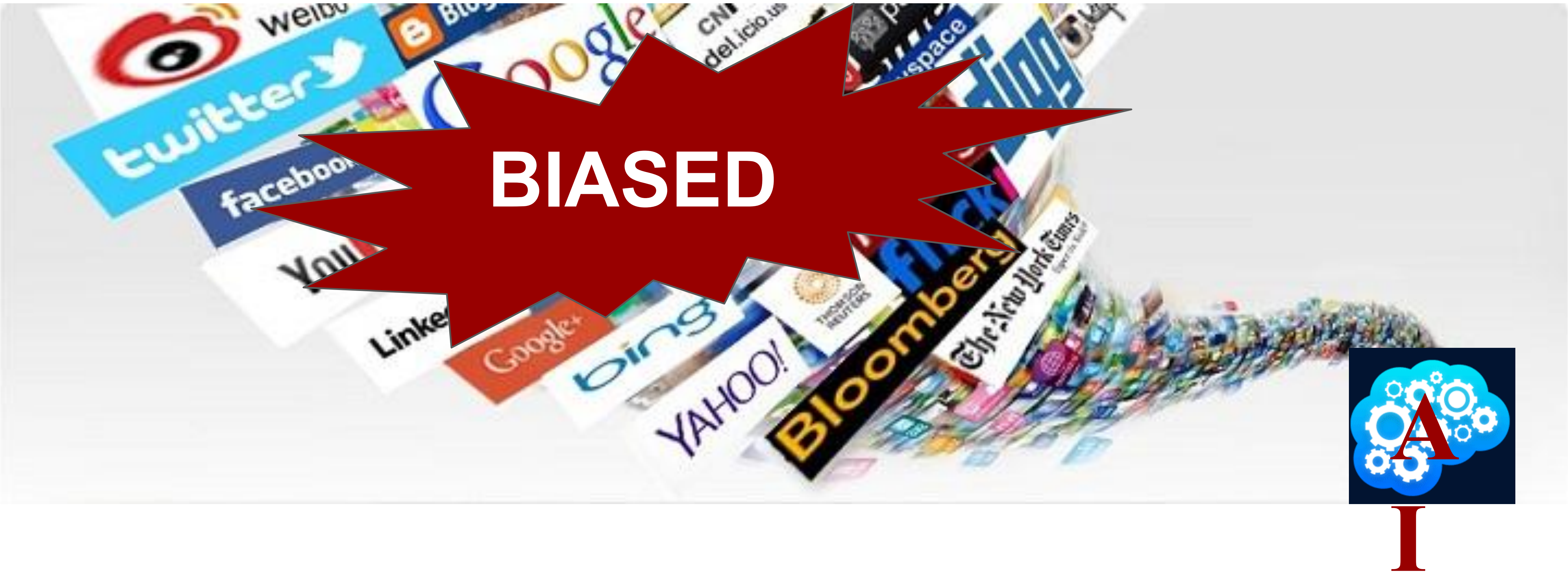
Learn to Assess AI Systems Adversarially

- Who could benefit from such a technology?
- Who can be harmed by such a technology?

- Representativeness of training data
- Could sharing this data have major effect on people’s lives?

- What are confounding variables and corner cases to control for?
- Does the system optimize for the “right” objective?
- Could prediction errors have major effect on people’s lives?

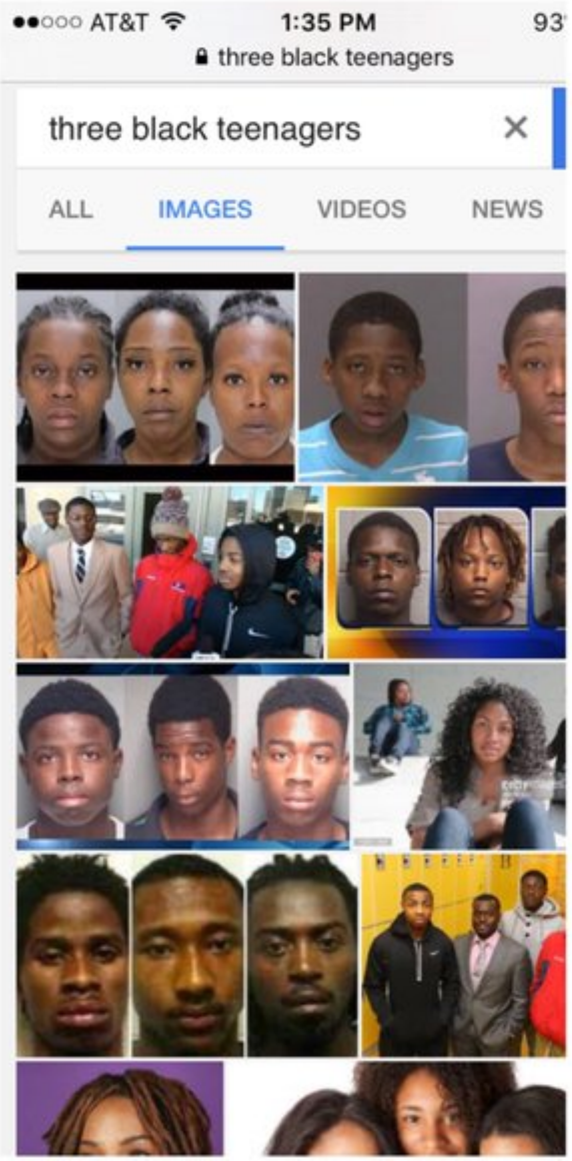
let’s start with the data…

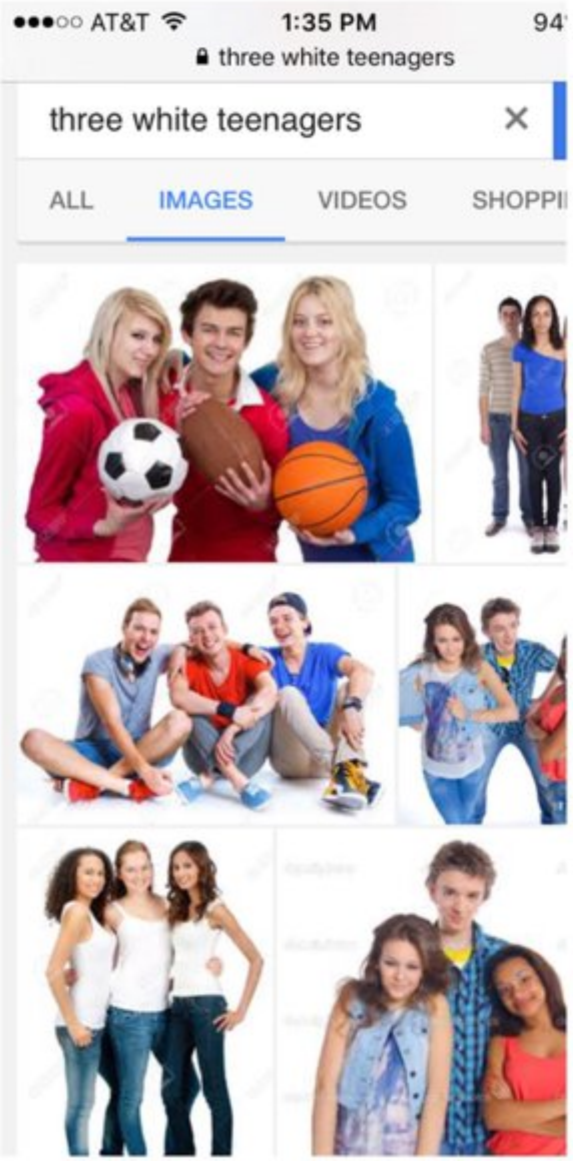


Online data is riddled with **SOCIAL STEREOTYPES**

Racial Stereotypes

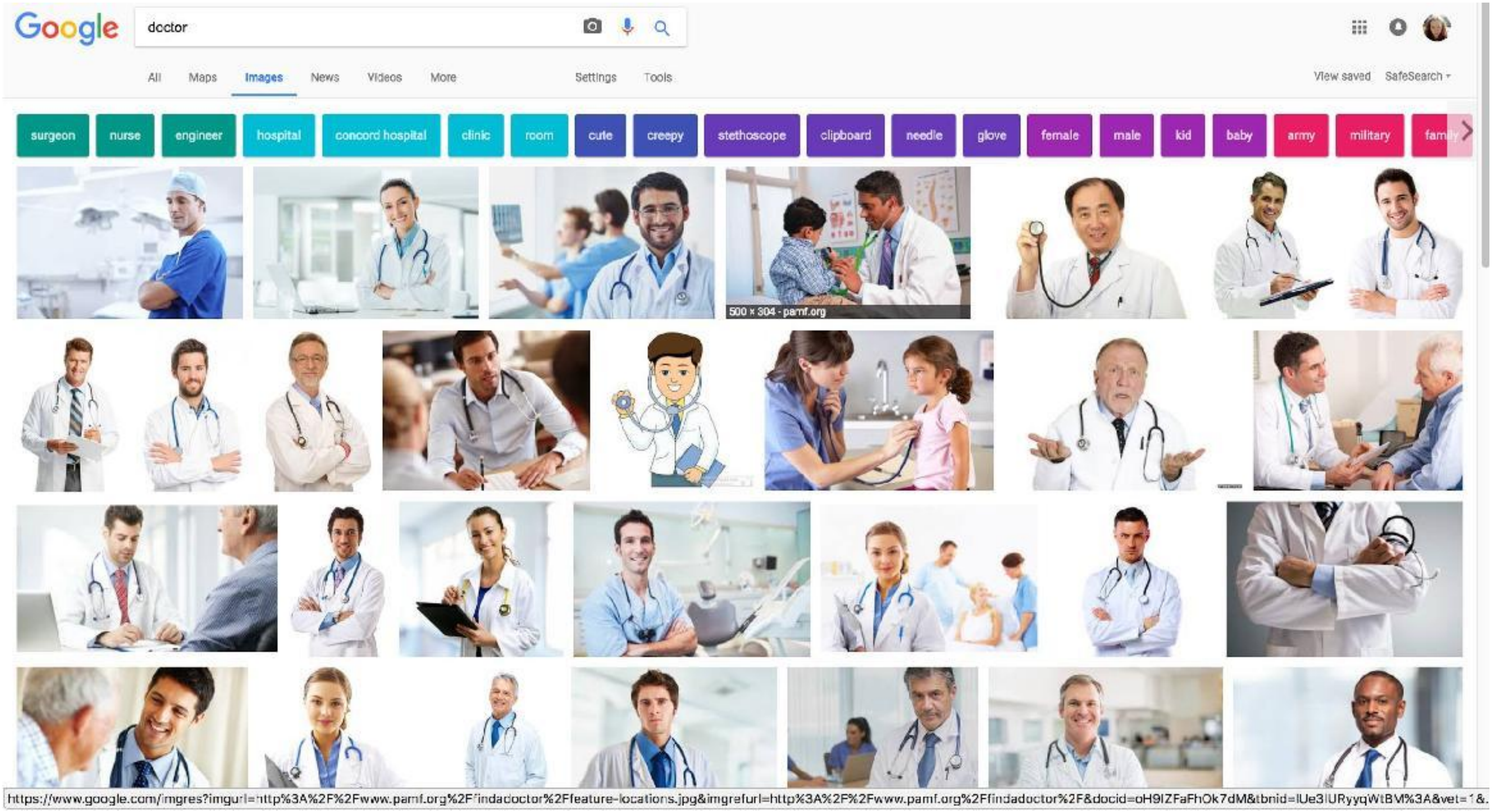
- June 2016: web search query “three black teenagers”





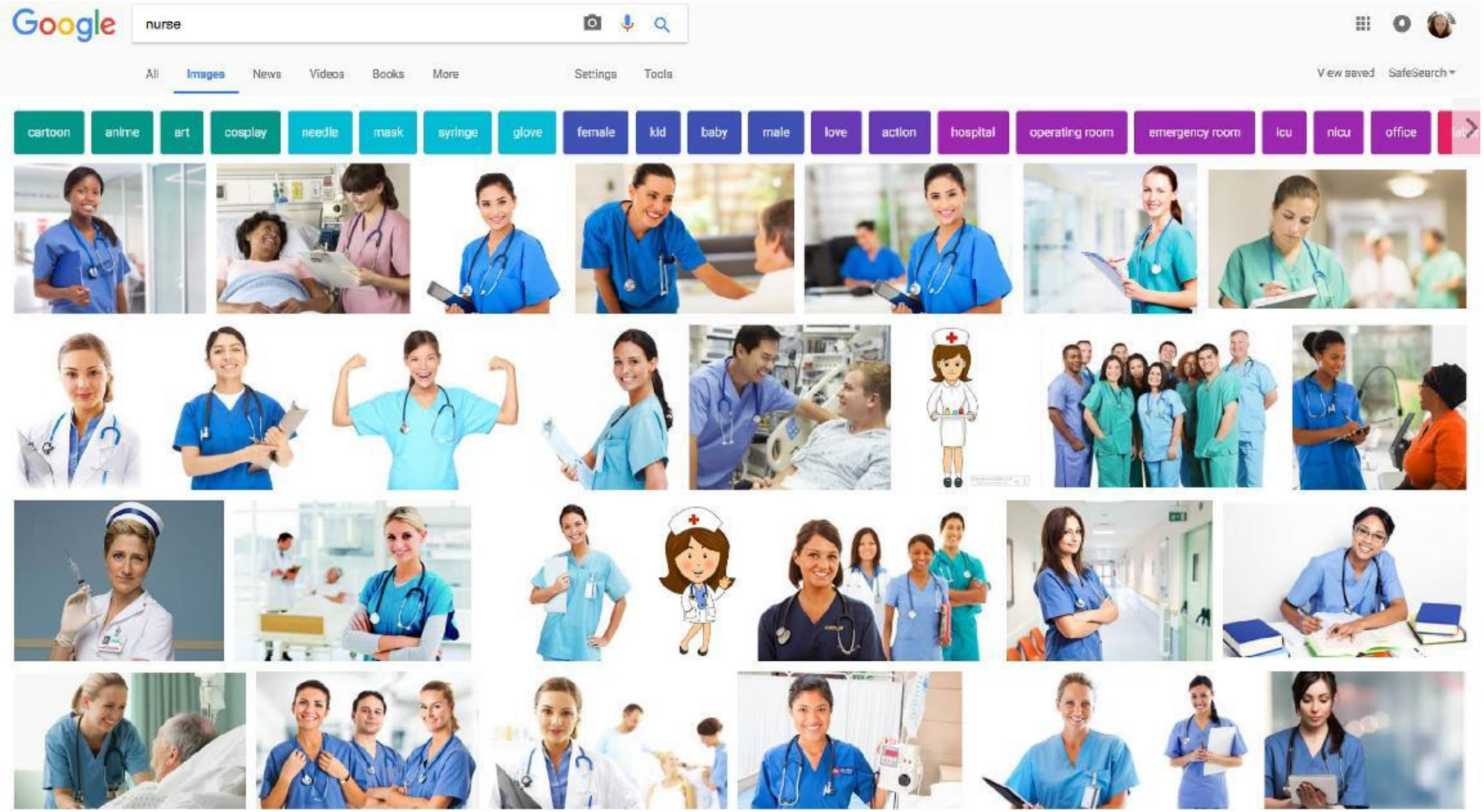
Gender/Race/Age Stereotypes

- June 2017: image search query “Doctor”



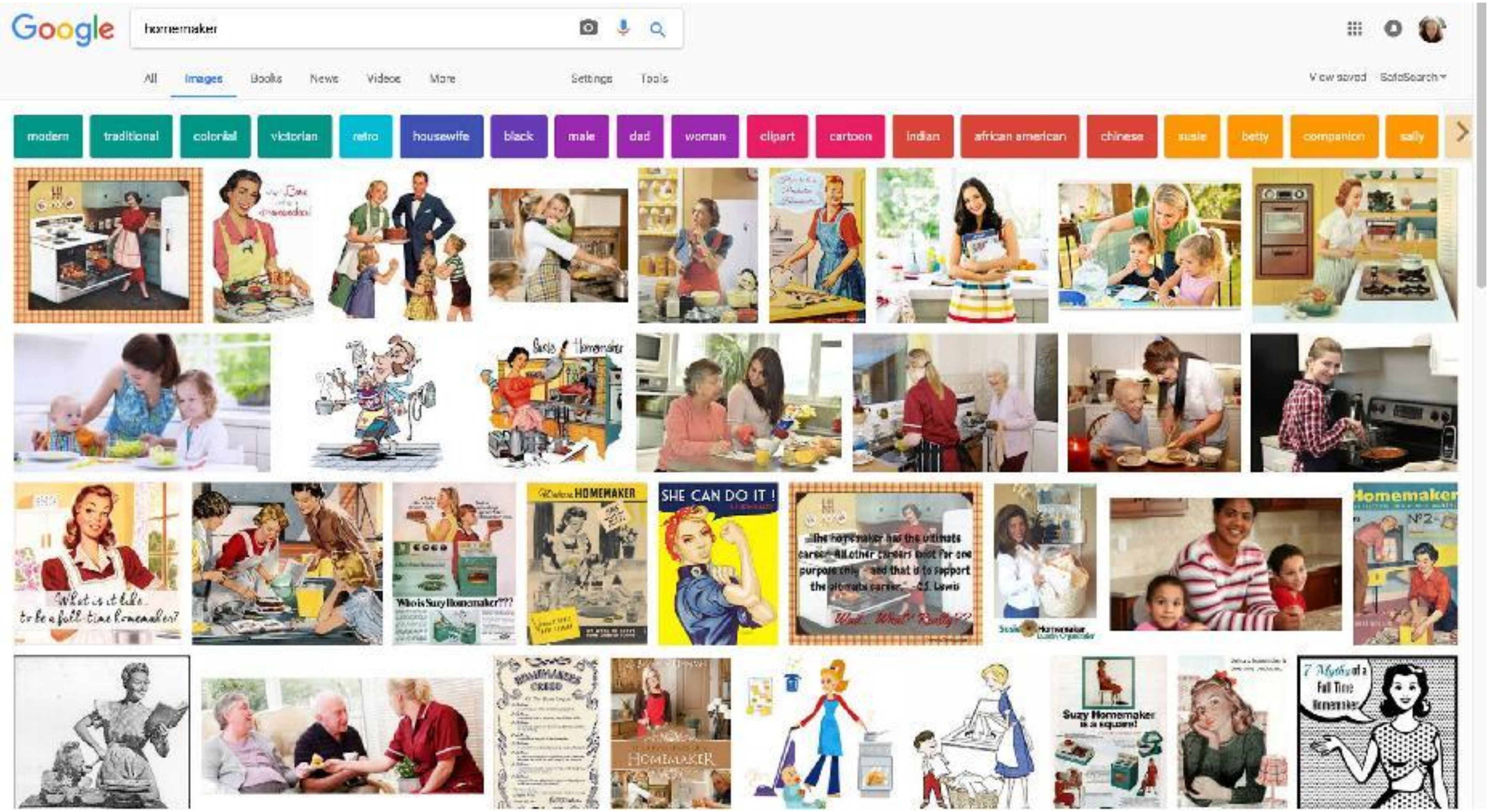
Gender/Race/Age Stereotypes

- June 2017: image search query "Nurse"



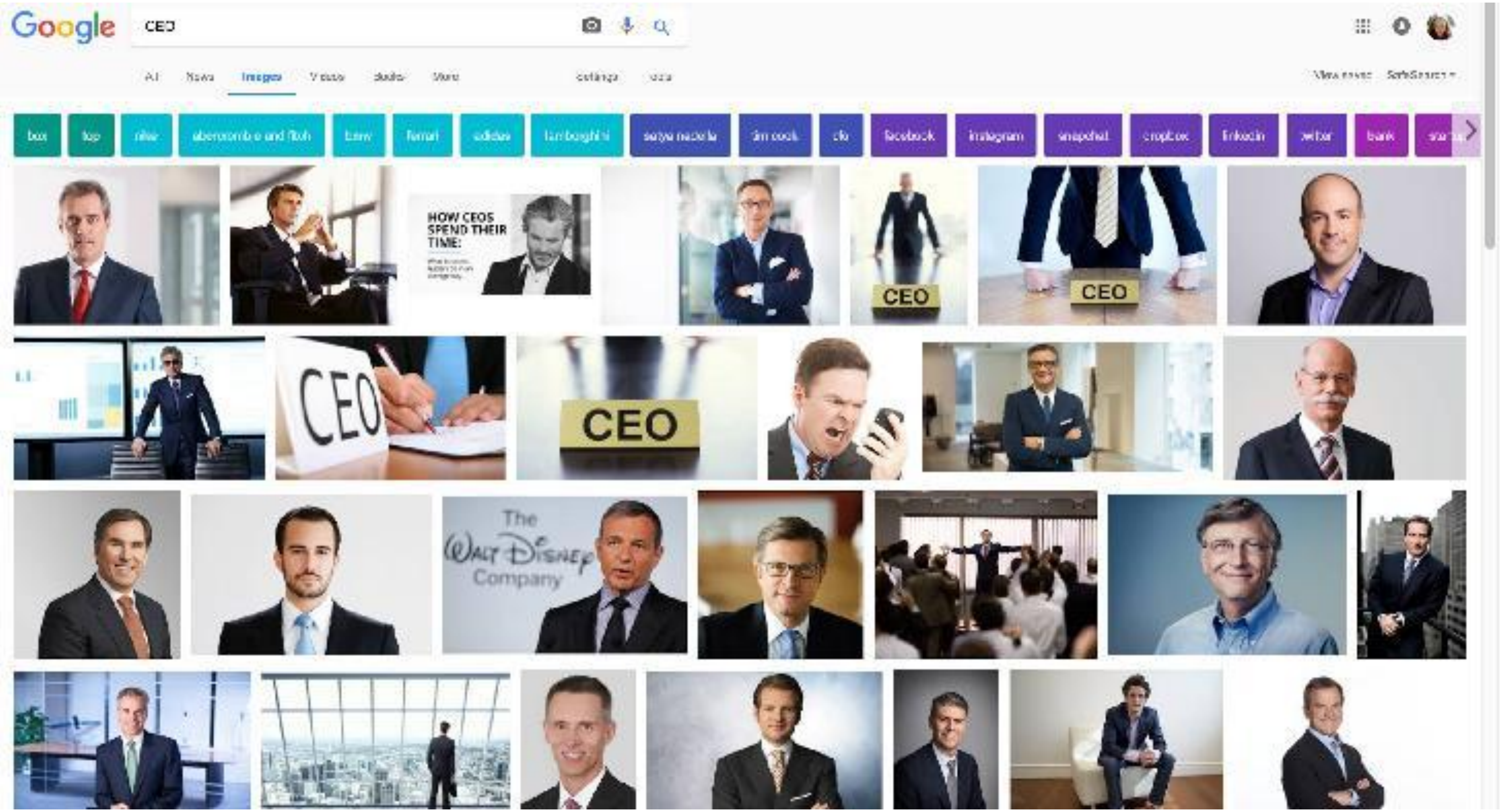
Gender/Race/Age Stereotypes

- June 2017: image search query “Homemaker”



Gender/Race/Age Stereotypes

- June 2017: image search query “CEO”



Consequence: models are biased

Gender Biases on the Web

- The dominant class is often portrayed and perceived as relatively more professional (Kay, Matuszek, and Munson 2015)
- Males are over-represented in the reporting of web-based news articles (Jia, Lansdall-Welfare, and Cristianini 2015)
- Males are over-represented in twitter conversations (Garcia, Weber, and Garimella 2014)
- Biographical articles about women on Wikipedia disproportionately discuss romantic relationships or family-related issues (Wagner et al. 2015)
- IMDB reviews written by women are perceived as less useful (Otterbacher 2013)

Biased NLP Technologies

- Bias in word embeddings (Bolukbasi et al. 2017; Caliskan et al. 2017; Garg et al. 2018)
- Bias in Language ID (Blodgett & O'Connor. 2017; Jurgens et al. 2017)
- Bias in Visual Semantic Role Labeling (Zhao et al. 2017)
- Bias in Natural Language Inference (Rudinger et al. 2017)
- Bias in Coreference Resolution (At NAACL: Rudinger et al. 2018; Zhao et al. 2018)
- Bias in Automated Essay Scoring (At NAACL: Amorim et al. 2018)

Sources of Human Biases in Machine Learning

- Bias in data and sampling
- Optimizing towards a biased objective
- Inductive bias
- Bias amplification in learned models

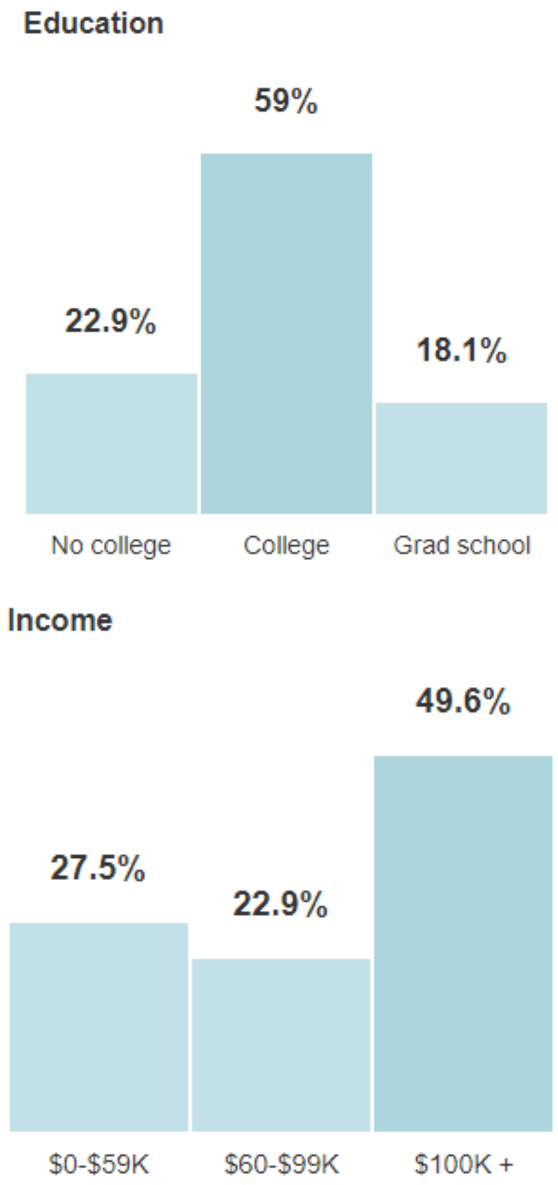
Sources of Human Biases in Machine Learning

- **Bias in data and sampling**
- Optimizing towards a biased objective
- Inductive bias
- Bias amplification in learned models

Types of Sampling Bias in Naturalistic Data

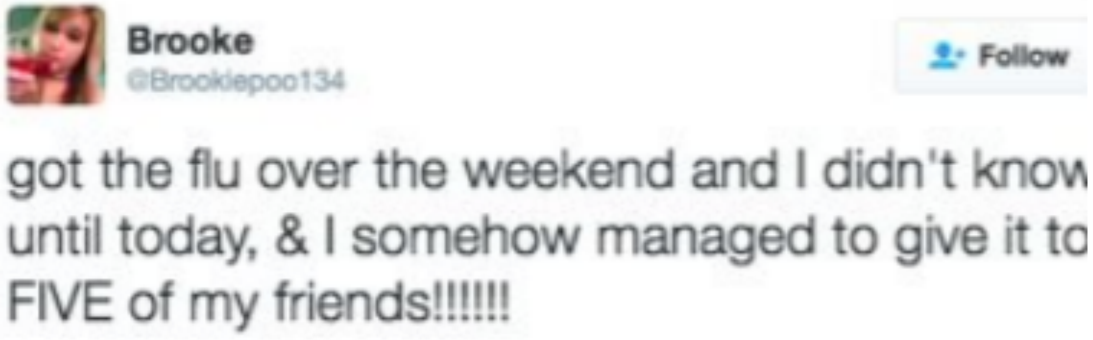
- Self-Selection Bias
  - Who decides to post reviews on Yelp and why?
    Who posts on Twitter and why?
- Reporting Bias
  - People do not necessarily talk about things in the world in proportion to their empirical distributions
    (Gordon and Van Durme 2013)
- Proprietary System Bias
  - What results does Twitter return for a particular query of interest and why? Is it possible to know?
- Community / Dialect / Socioeconomic Biases
  - What linguistic communities are over- or under-represented?
    leads to community-specific model performance (Jorgensen et al. 2015)

US Demographics of Yelp Users



Example: Bias in Language Identification

- Most applications employ off-the-shelf LID systems which are highly accurate



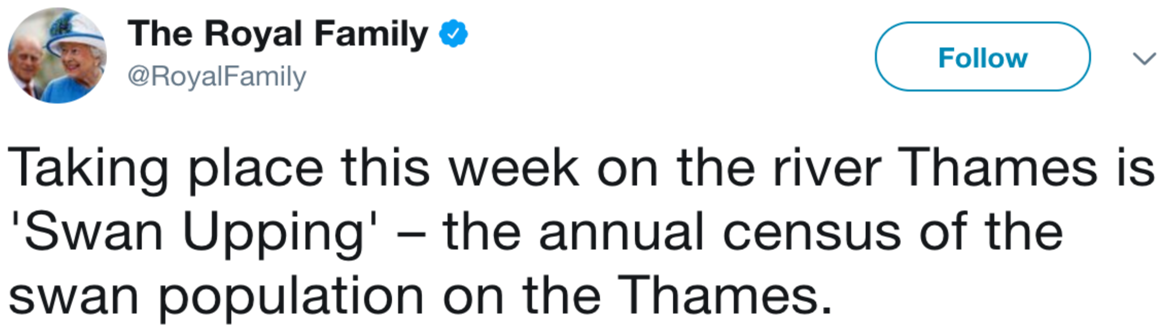
Brooke
@Brookiepoo134

got the flu over the weekend and I didn't know until today, & I somehow managed to give it to FIVE of my friends!!!!!!

*Slides on LID by David Jurgens
(Jurgens et al. ACL’17)

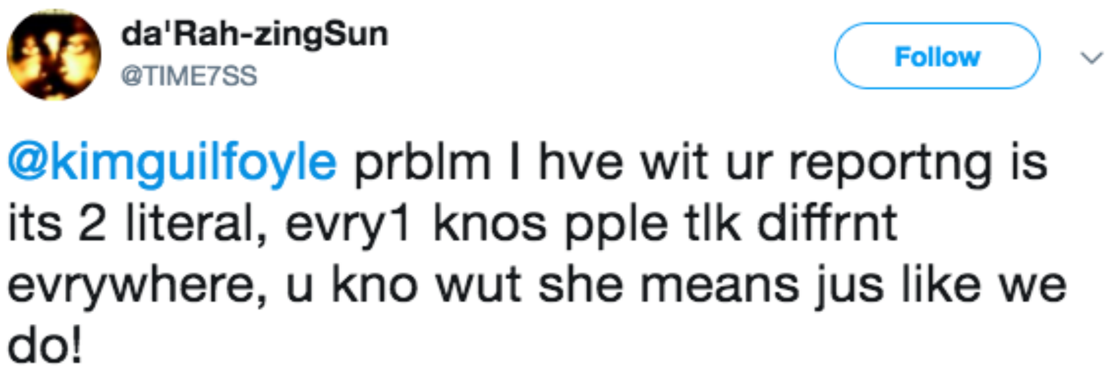
McNamee, P., “Language identification: *a solved problem* suitable for undergraduate instruction” Journal of Computing Sciences in Colleges 20(3) 2005.

“This paper describes […] how even the most simple of these methods **using data obtained from the World Wide Web achieve accuracy approaching 100%** on a test suite comprised of ten European languages”



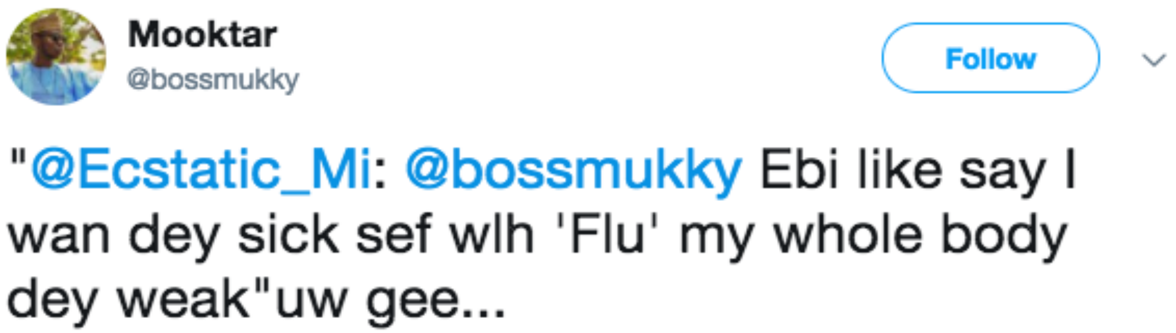
The Royal Family
@RoyalFamily

Taking place this week on the river Thames is 'Swan Upping' – the annual census of the swan population on the Thames.



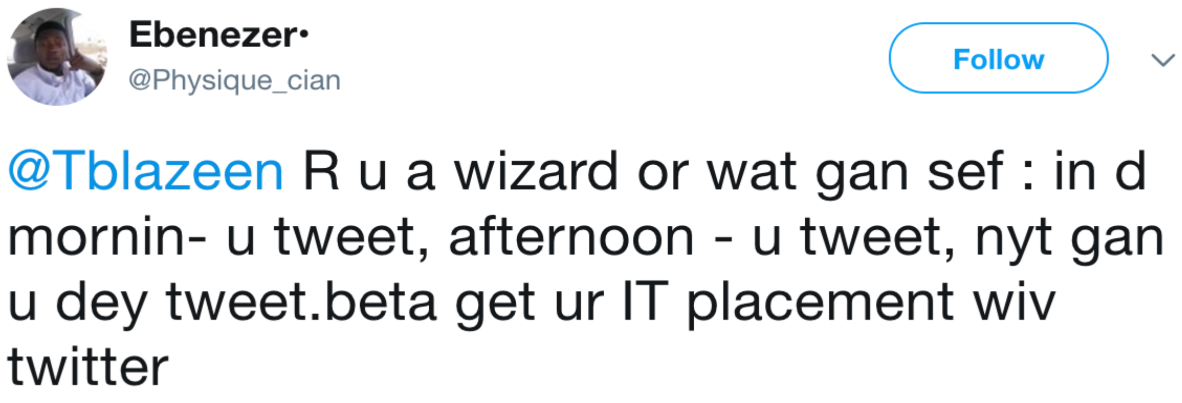
da'Rah-zingSun
@TIME7SS

@kimguilfoyle prblm I hve wit ur reportng is its 2 literal, evry1 knos pple tlk diffrnt evrywhere, u kno wut she means jus like we do!



Mooktar
@bossmukky

"@Ecstatic_Mi: @bossmukky Ebi like say I wan dey sick sef wlh 'Flu' my whole body dey weak"uw gee...

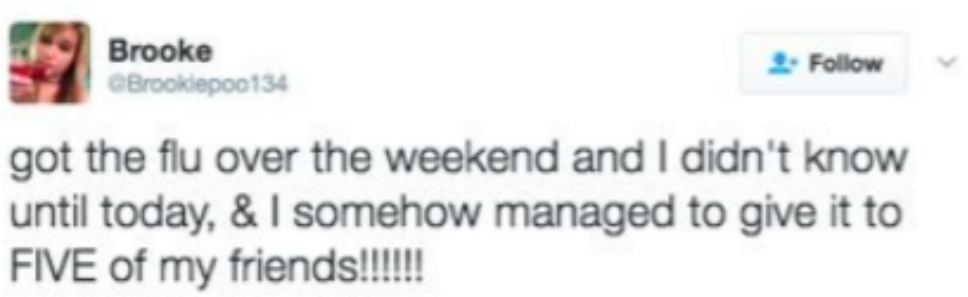


Ebenezer•
@Physique_cian

@Tblazeen R u a wizard or wat gan sef : in d mornin- u tweet, afternoon - u tweet, nyt gan u dey tweet.beta get ur IT placement wiv twitter

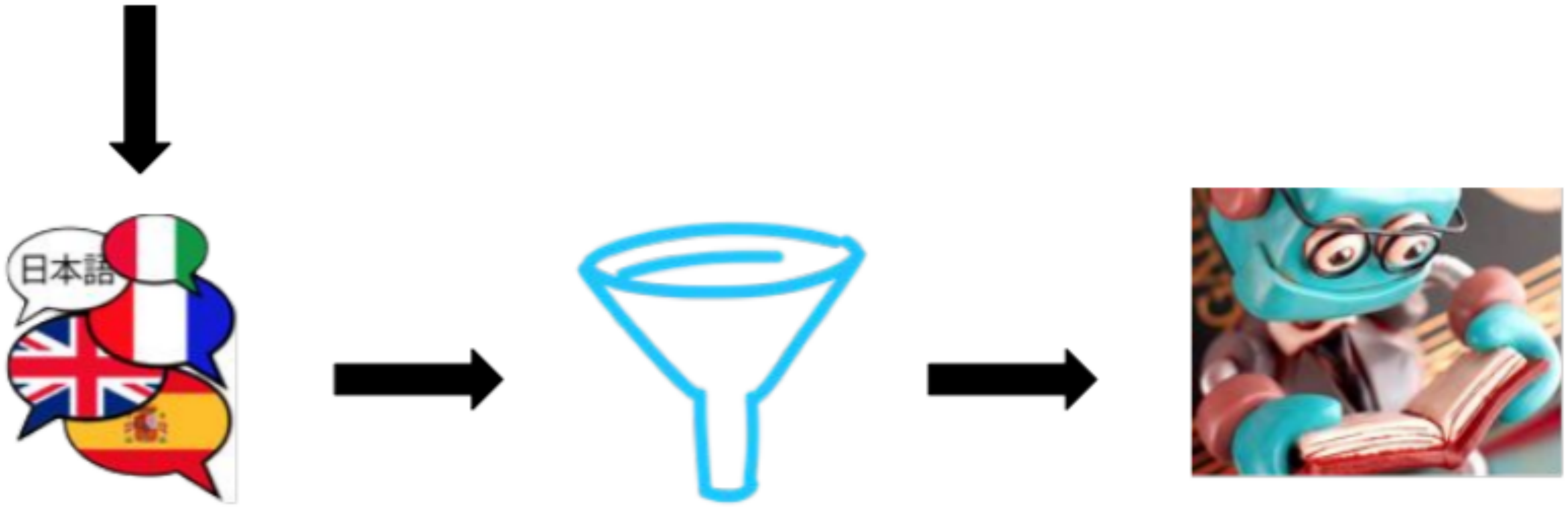
- Language identification degrades significantly on African American Vernacular English (Blodgett et al. 2016) **Su-Lin Blodgett is a UMass NLP PhD student!**

LID Usage Example: Health Monitoring



Brooke
@Brookiepoo134

got the flu over the weekend and I didn't know until today, & I somehow managed to give it to FIVE of my friends!!!!!!

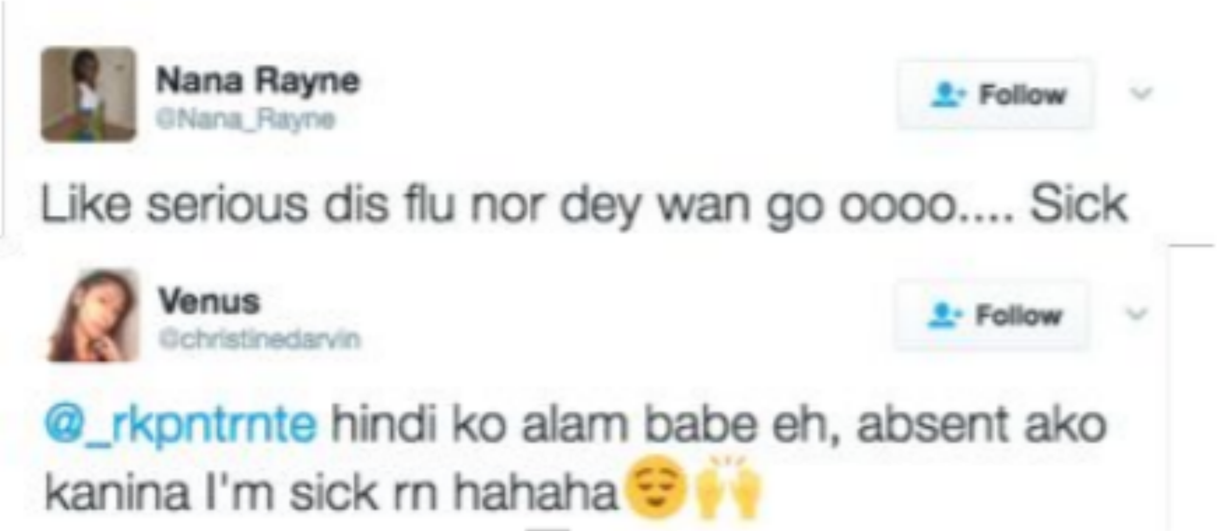


Language Detection

Keyword Filter
"flu", "sick"

Analytics
Which symptoms?
Are they hungover?

LID Usage Example: Health Monitoring



Nana Rayne
@Nana_Rayne

Follow

Like serious dis flu nor dey wan go oooo.... Sick

Venus
@christinedarvin

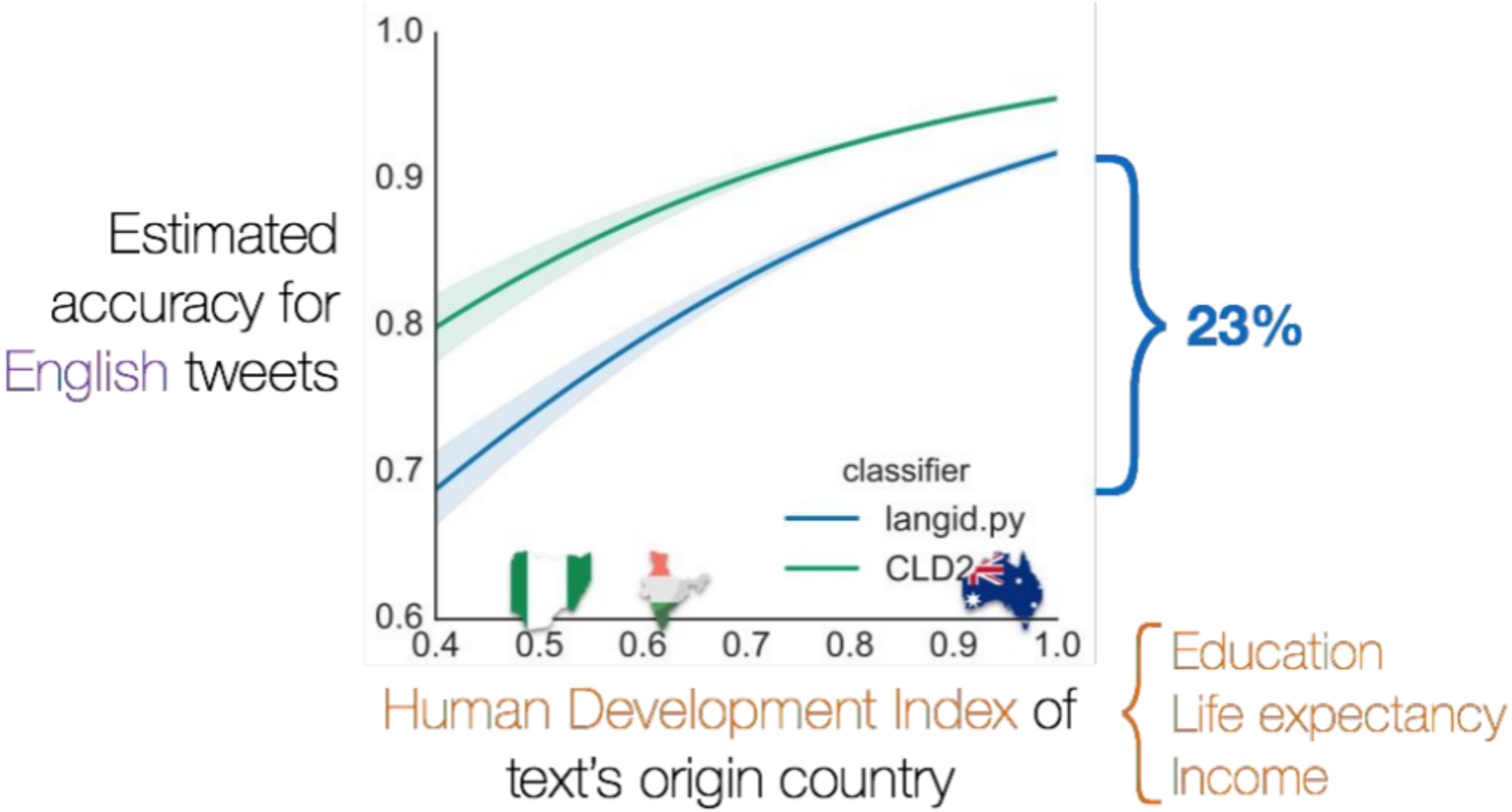
Follow

@_rkpntrnte hindi ko alam babe eh, absent ako kanina I'm sick rn hahaha😌🙌

Language Detection

Socioeconomic Bias in Language Identification

- Off-the-shelf LID systems under-represent populations in less-developed countries



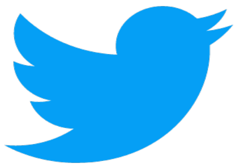
Jurgens et al. ACL’17

Better Social Representation through Network-based Sampling

- Re-sampling from strategically-diverse corpora

Topical

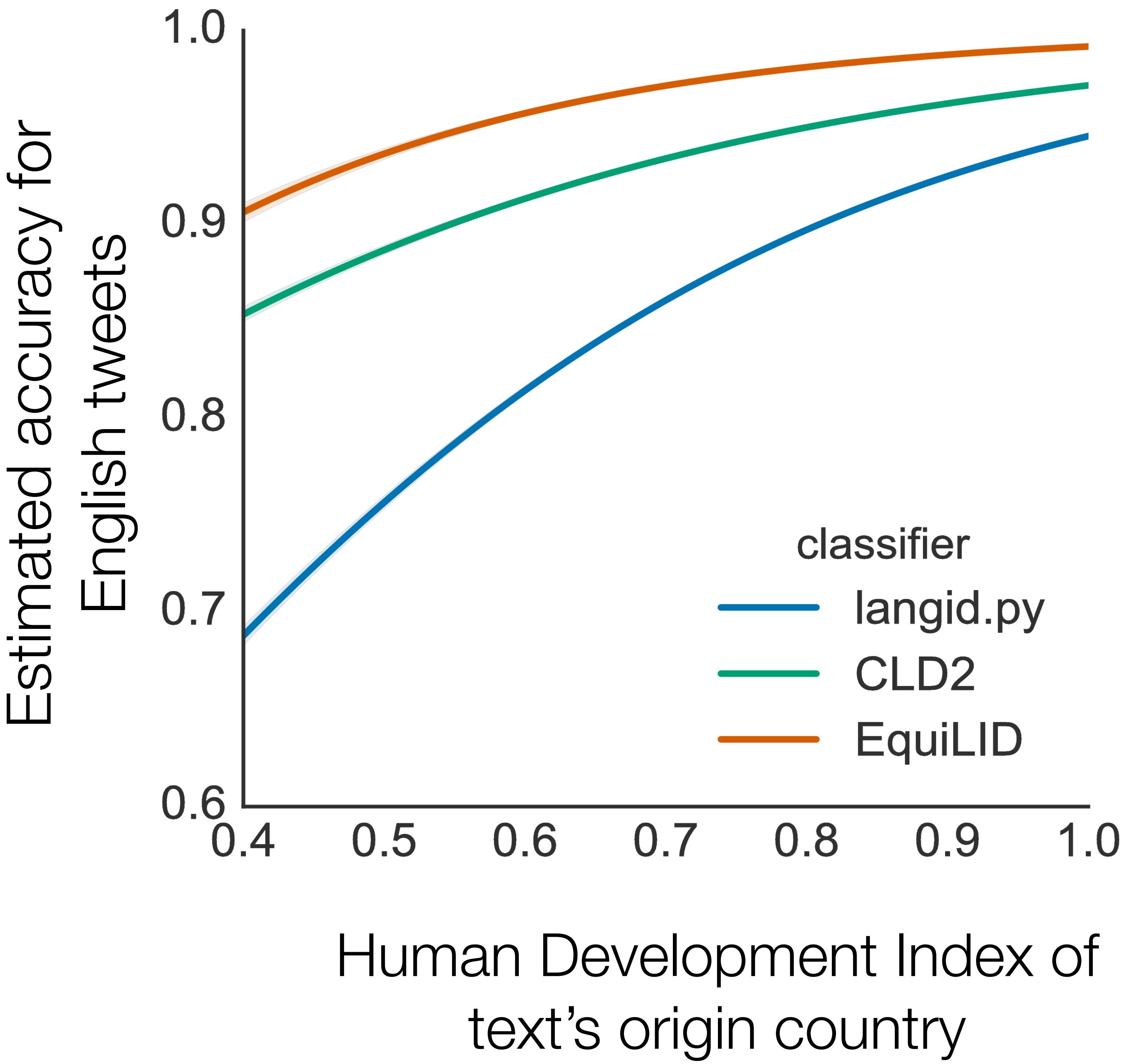
Geographic



Social

Jurgens et al. ACL’1

Multilingual



Jurgens et al. ACL'17

Sources of Human Biases in Machine Learning

- Bias in data and sampling
- **Optimizing towards a biased objective**
- Inductive bias
- Bias amplification in learned models

Optimizing Towards a Biased Objective

- Northpointe vs ProPublica

COMPAS

Optimizing Towards a Biased Objective

“what is the probability that this person will commit a serious crime in the future, as a function of the sentence you give them now?”

Optimizing Towards a Biased Objective

“what is the probability that this person will commit a serious crime in the future, as a function of the sentence you give them now?”

- COMPAS system
  - balanced training data about people of all races
  - race was *not* one of the input features
- Objective function
  - labels for “who will commit a crime” are unobtainable
  - a proxy for the real, unobtainable data: “who is more likely to be *convicted*”

what are some issues with this proxy objective?

Sources of Human Biases in Machine Learning

- Bias in data and sampling
- Optimizing towards a biased objective
- **Inductive bias**
- Bias amplification in learned models

what is inductive bias?

- the assumptions used by our model. examples:
    - recurrent neural networks for NLP assume that the sequential ordering of words is meaningful
    - features in discriminative models are assumed to be useful to map inputs to outputs

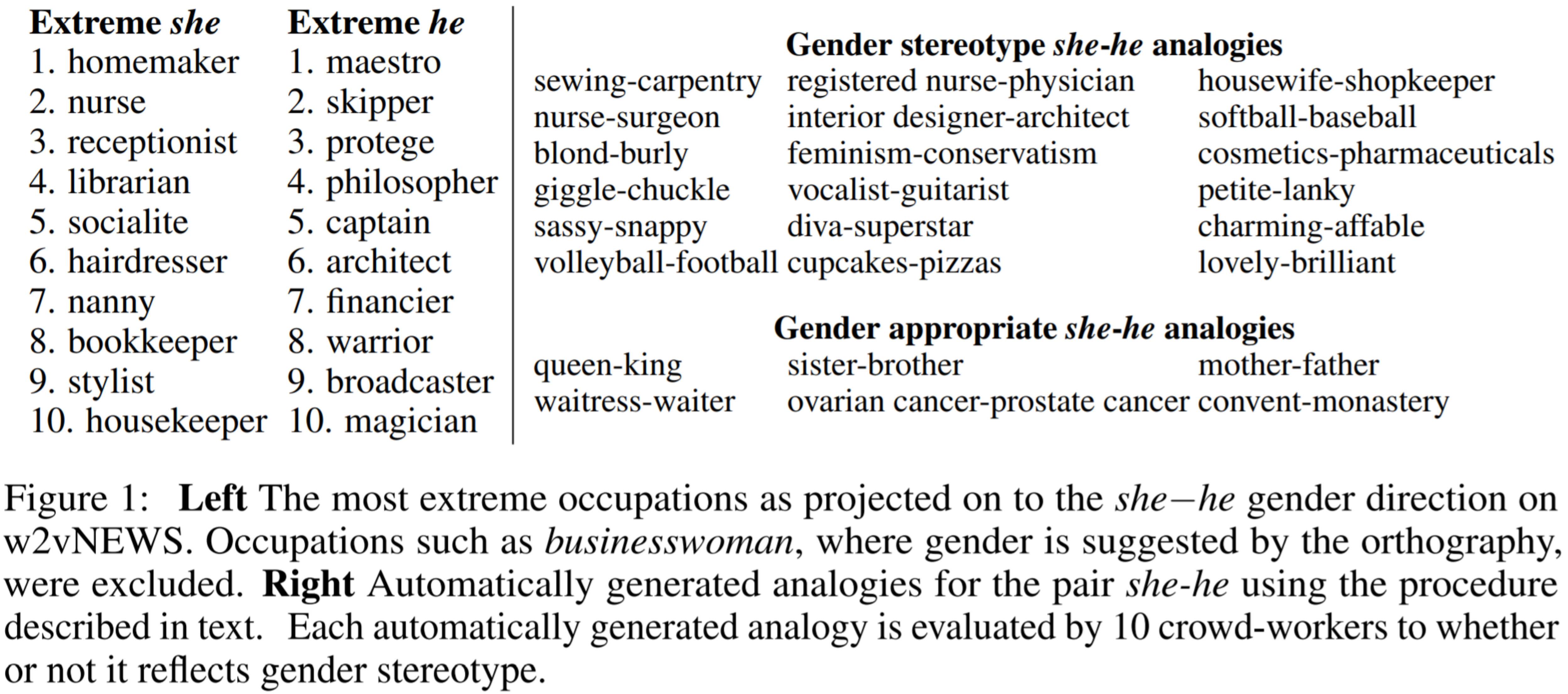
Bias in Word Embeddings

1. Caliskan, A., Bryson, J. J. and Narayanan, A. (2017) **Semantics derived automatically from language corpora contain human-like biases.** *Science*

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

Biases in Embeddings: Another Take

$$\min cos(he - she,\ x - y)\ s.t.\ ||x - y||_2 < \delta$$



Extreme *she*	Extreme *he*
1. homemaker	1. maestro
2. nurse	2. skipper
3. receptionist	3. protege
4. librarian	4. philosopher
5. socialite	5. captain
6. hairdresser	6. architect
7. nanny	7. financier
8. bookkeeper	8. warrior
9. stylist	9. broadcaster
10. housekeeper	10. magician

Gender stereotype *she*-*he* analogies

sewing-carpentry	registered 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	lovely-brilliant

Gender appropriate *she*-*he* analogies

queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Figure 1: **Left** The most extreme occupations as projected on to the *she*−*he* gender direction on w2vNEWS. Occupations such as *businesswoman*, where gender is suggested by the orthography, were excluded. **Right** Automatically generated analogies for the pair *she*-*he* using the procedure described in text. Each automatically generated analogy is evaluated by 10 crowd-workers to whether or not it reflects gender stereotype.

Towards Debiasing

1. Identify gender subspace: B

Gender Subspace

$\overrightarrow{\text{she}}-\overrightarrow{\text{he}}$

$\overrightarrow{\text{her}}-\overrightarrow{\text{his}}$

$\overrightarrow{\text{woman}}-\overrightarrow{\text{man}}$

$\overrightarrow{\text{Mary}}-\overrightarrow{\text{John}}$

$\overrightarrow{\text{herself}}-\overrightarrow{\text{himself}}$

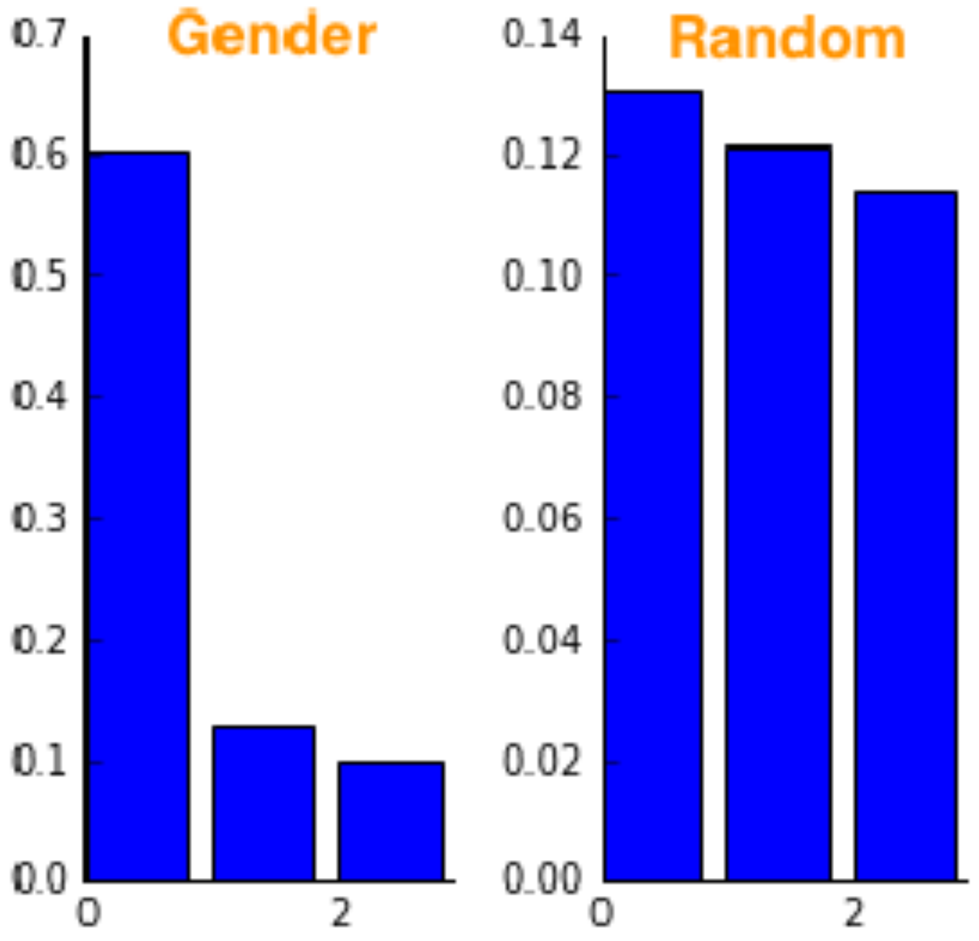
$\overrightarrow{\text{daughter}}-\overrightarrow{\text{son}}$

$\overrightarrow{\text{mother}}-\overrightarrow{\text{father}}$

$\overrightarrow{\text{gal}}-\overrightarrow{\text{guy}}$

$\overrightarrow{\text{girl}}-\overrightarrow{\text{boy}}$

$\overrightarrow{\text{female}}-\overrightarrow{\text{male}}$

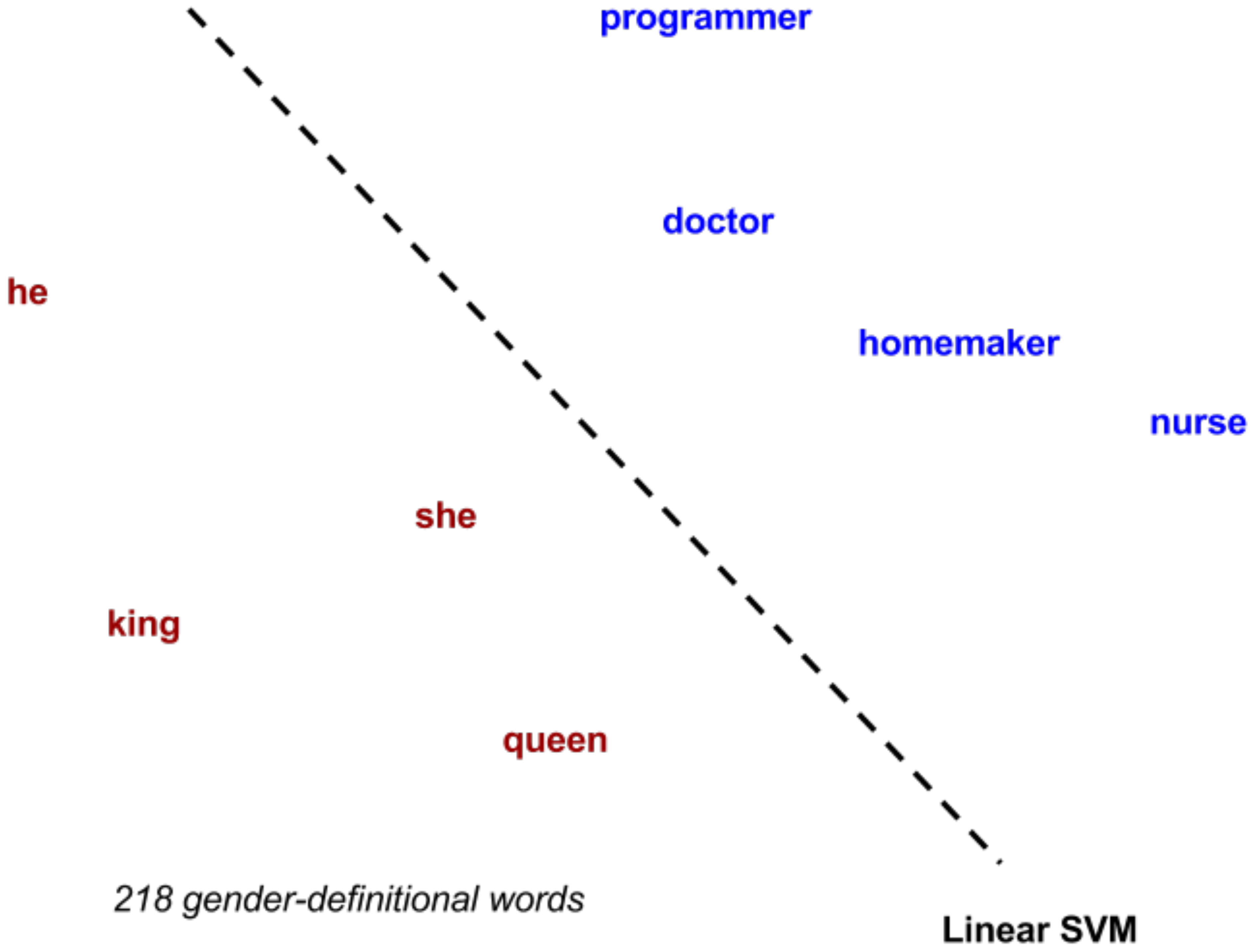


The top PC captures the gender subspace

Towards Debiasing

1. Identify gender subspace: B
2. **Identify gender-definitional (S) and gender-neutral words (N)**

Gender-definitional vs. Gender-neutral Words



Towards Debiasing

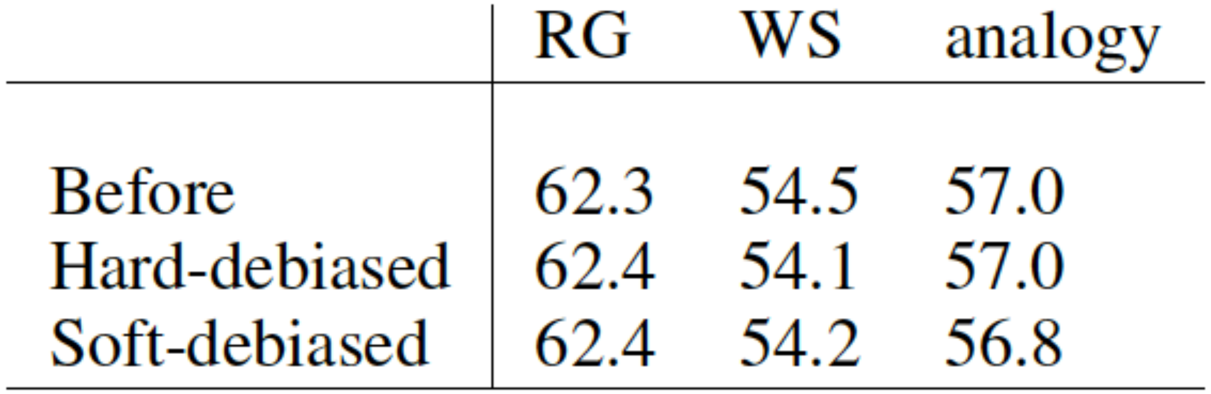
1. Identify gender subspace: B
2. Identify gender-definitional (S) and gender-neutral words (N)
3. Apply transform matrix (T) to the embedding matrix (W) such that
    a. Project away the gender subspace B from the gender-neutral words N
    b. But, ensure the transformation doesn't change the embeddings too much

$$min_T \underbrace{||(TW)^T(TW) - W^TW||_F^2}_{\text{Don't modify embeddings too much}} + \lambda \underbrace{||(TN)^T(TB)||_F^2}_{\text{Minimize gender component}}$$

T - the desired debiasing transformation　　B - biased space
W - embedding matrix
N - embedding matrix of gender neutral words

Does Debiasing Reduce Utility?

The performance does not degrade after debiasing



	RG	WS	analogy
Before	62.3	54.5	57.0
Hard-debiased	62.4	54.1	57.0
Soft-debiased	62.4	54.2	56.8

RG: Synonymy; Rubenstein & Goodenough (1965)
WS: Word Similarity

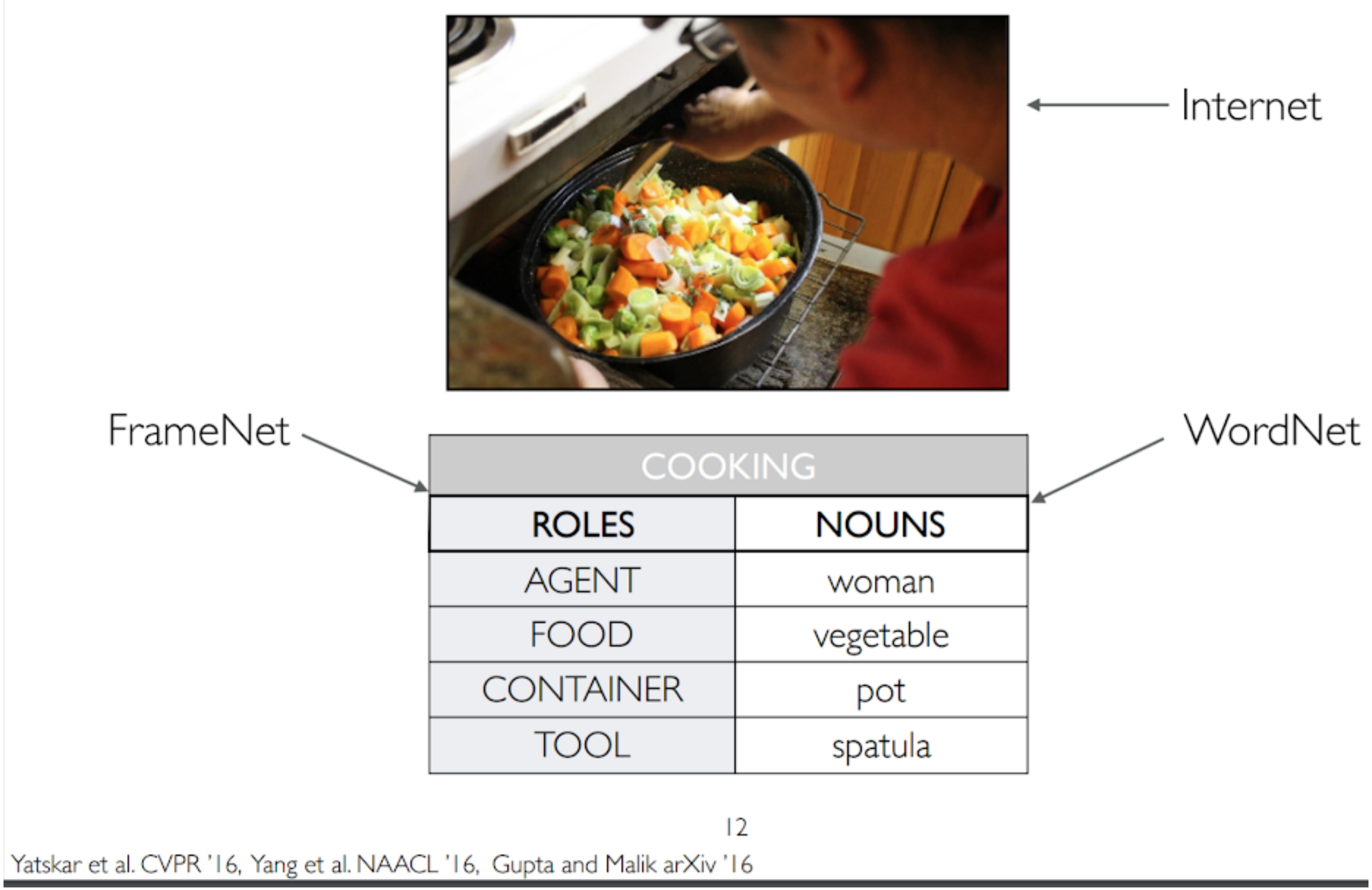
Sources of Human Biases in Machine Learning

- Bias in data and sampling
- Optimizing towards a biased objective
- Inductive bias
- **Bias amplification in learned models**

Bias Amplification

Zhao, J., Wang, T., Yatskar, M., Ordonez, V and Chang, M.-W. (2017) **Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraint.** *EMNLP*

imSitu Visual Semantic Role Labeling (vSRL)

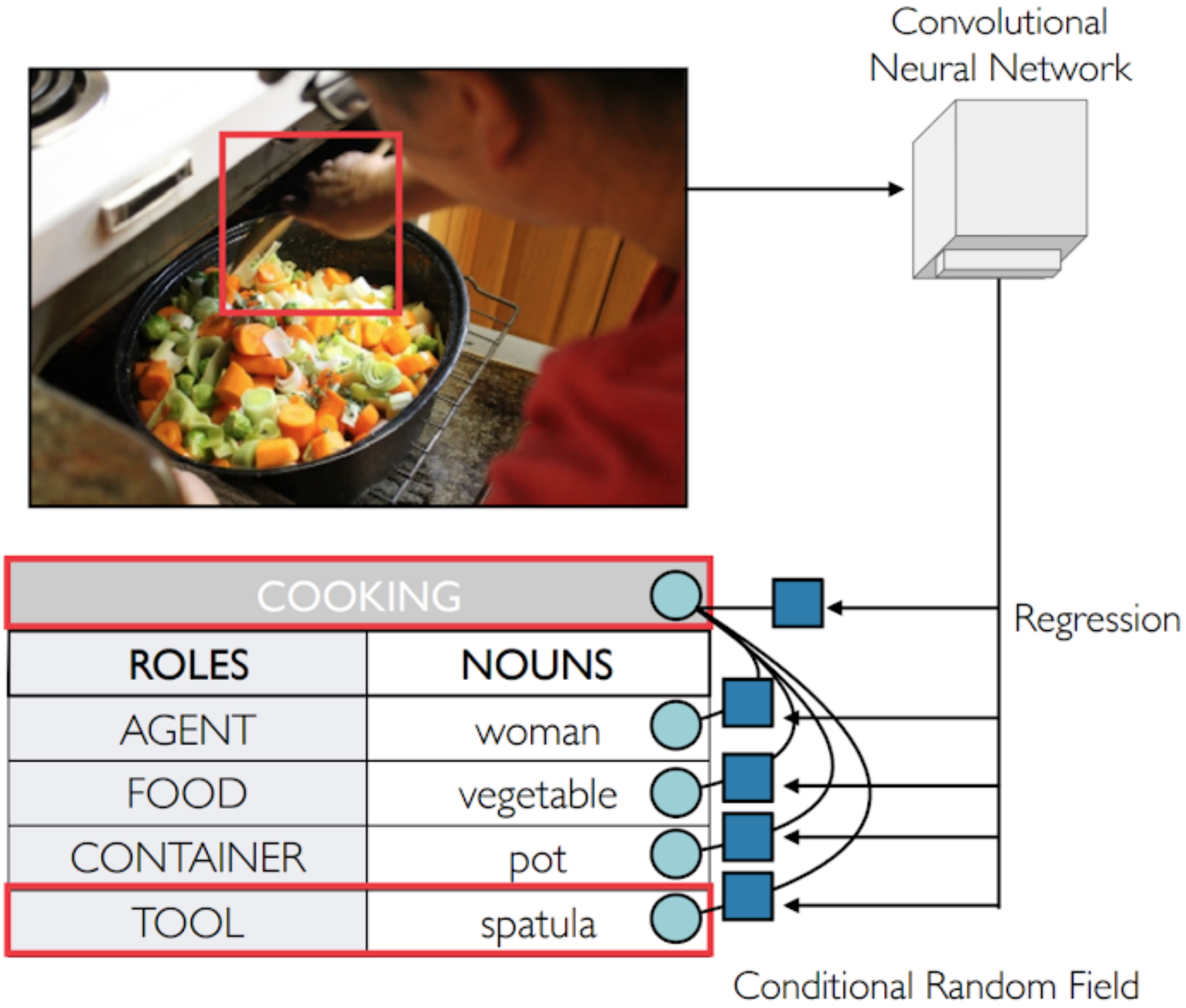


COOKING	
ROLES	**NOUNS**
AGENT	woman
FOOD	vegetable
CONTAINER	pot
TOOL	spatula

Yatskar et al. CVPR '16, Yang et al. NAACL '16, Gupta and Malik arXiv '16

Slides by Mark Yatskar https://homes.cs.washington.edu/~my89/talks/ZWYOC17_slide.pdf

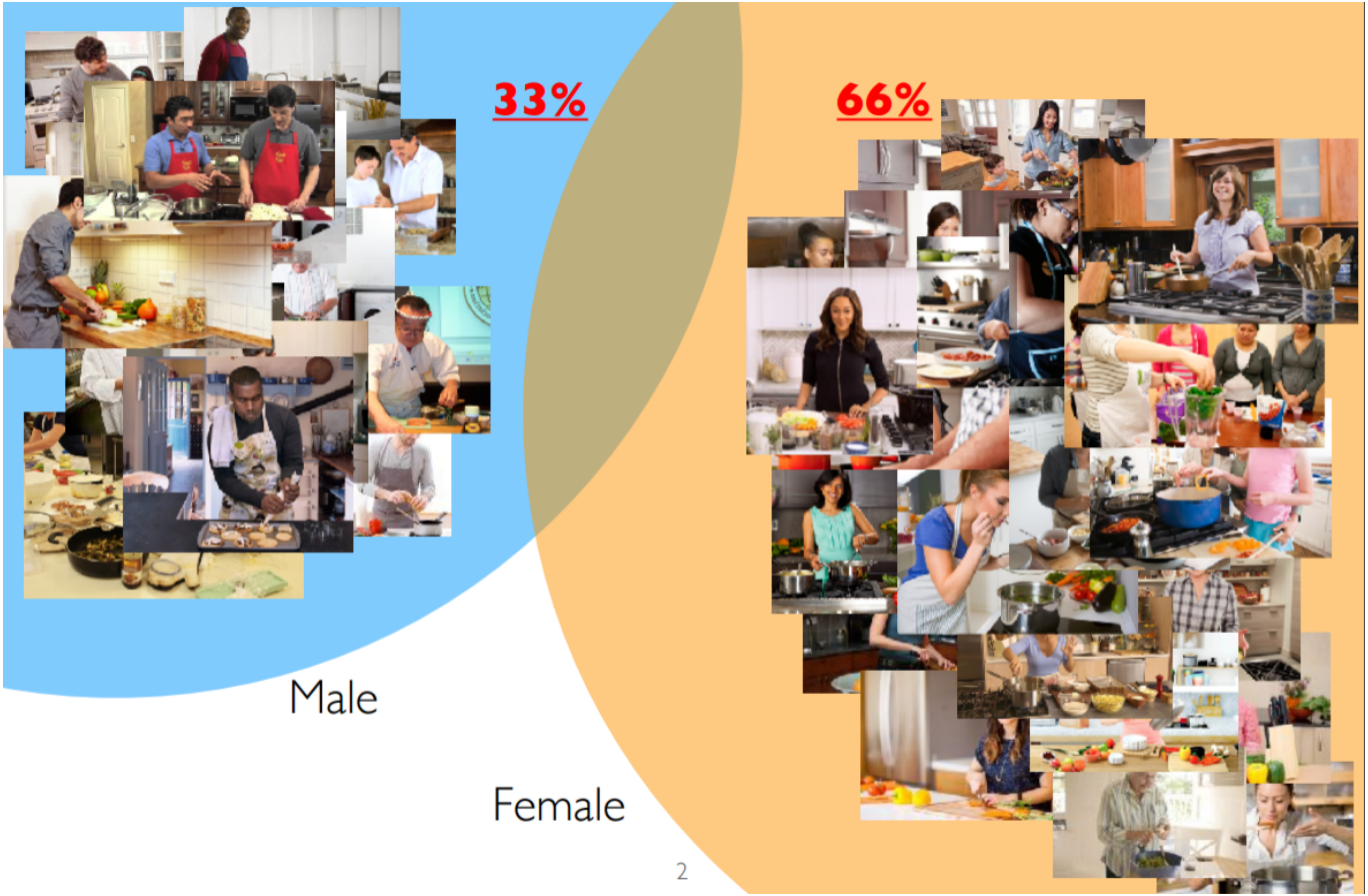
imSitu Visual Semantic Role Labeling (vSRL)



COOKING	
ROLES	NOUNS
AGENT	woman
FOOD	vegetable
CONTAINER	pot
TOOL	spatula

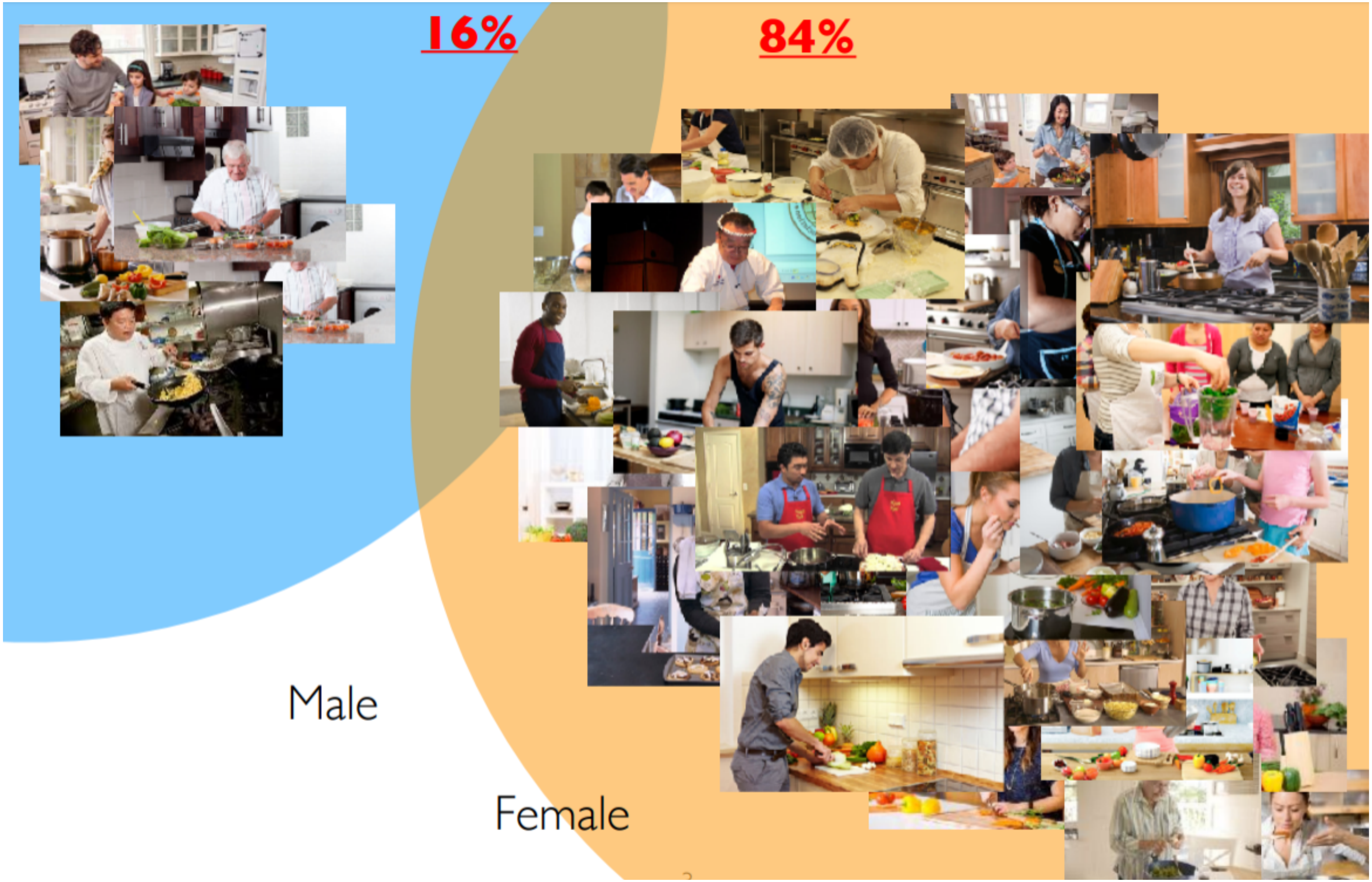
by Mark Yatskar

Dataset Gender Bias



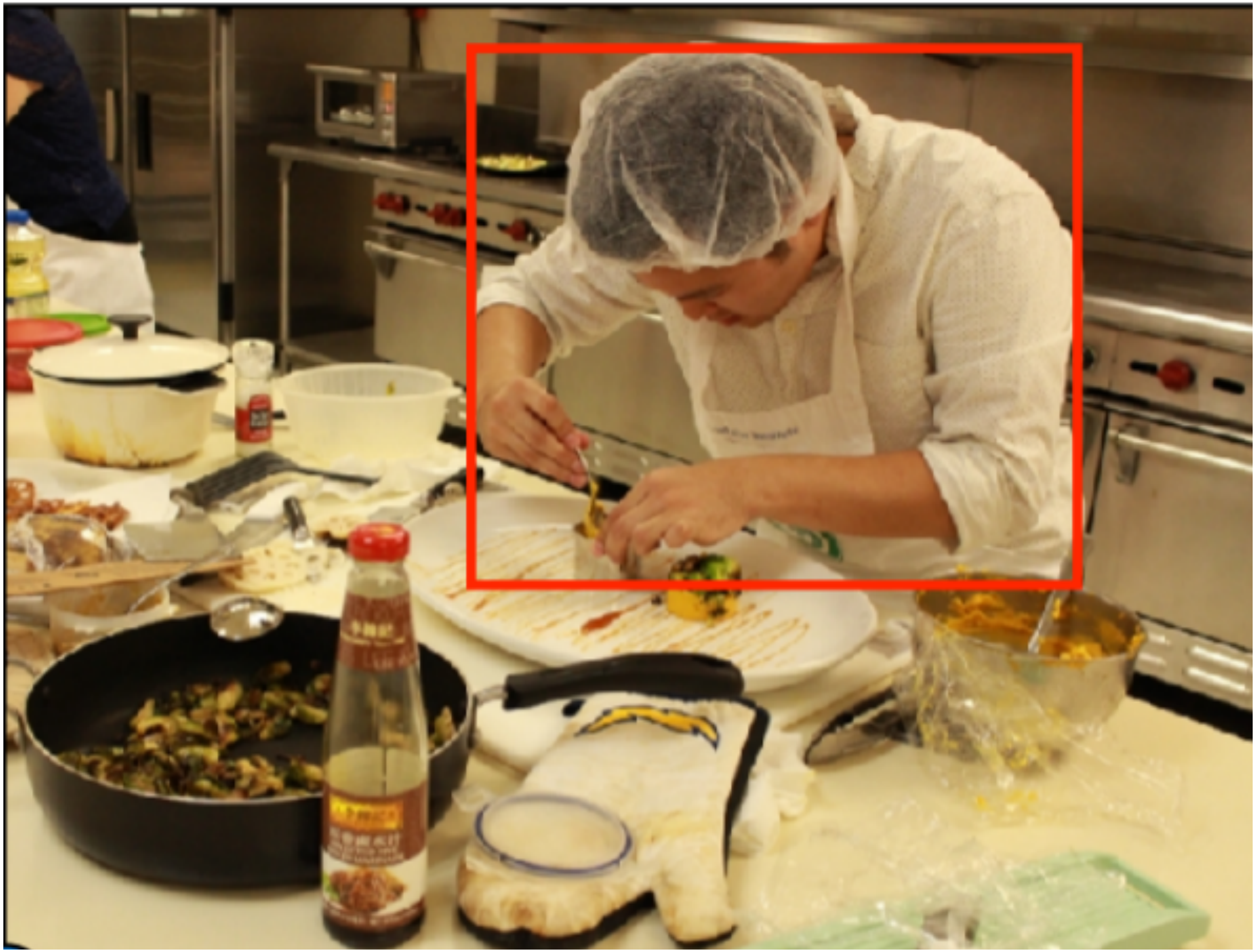
by Mark Yatskar

Model Bias After Training



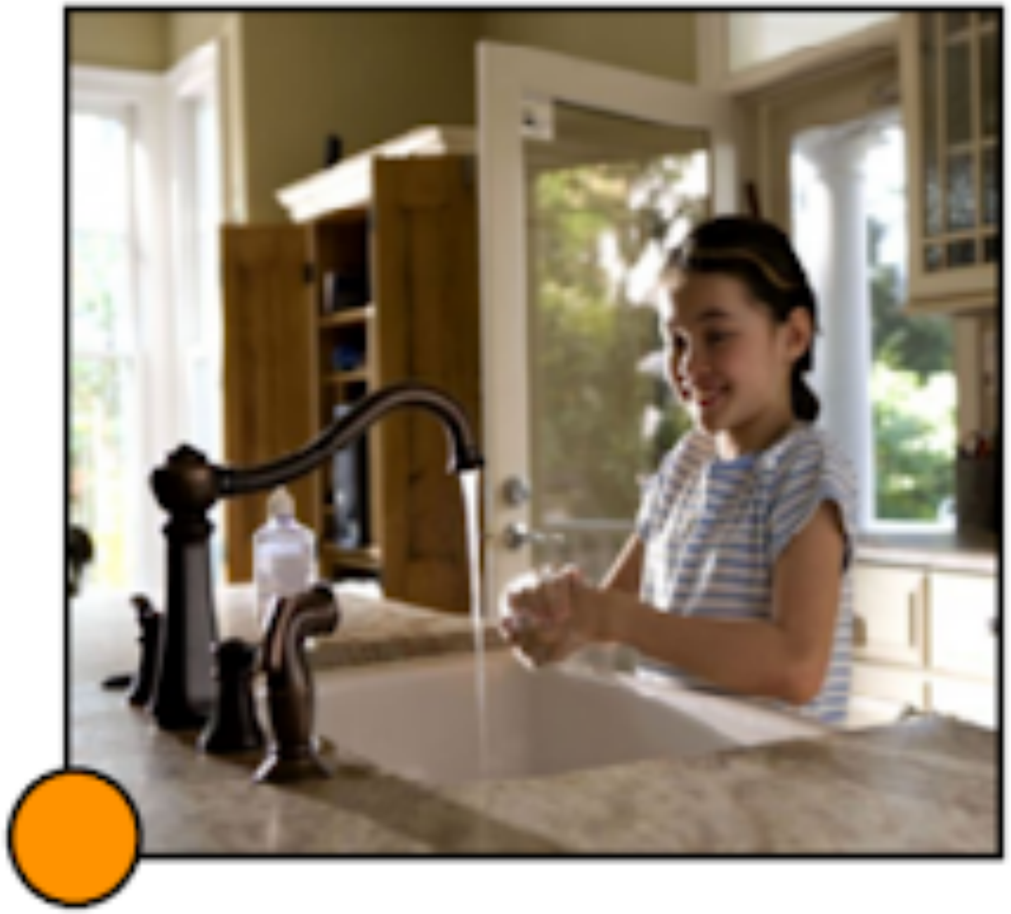
by Mark Yatskar

Why does this happen?

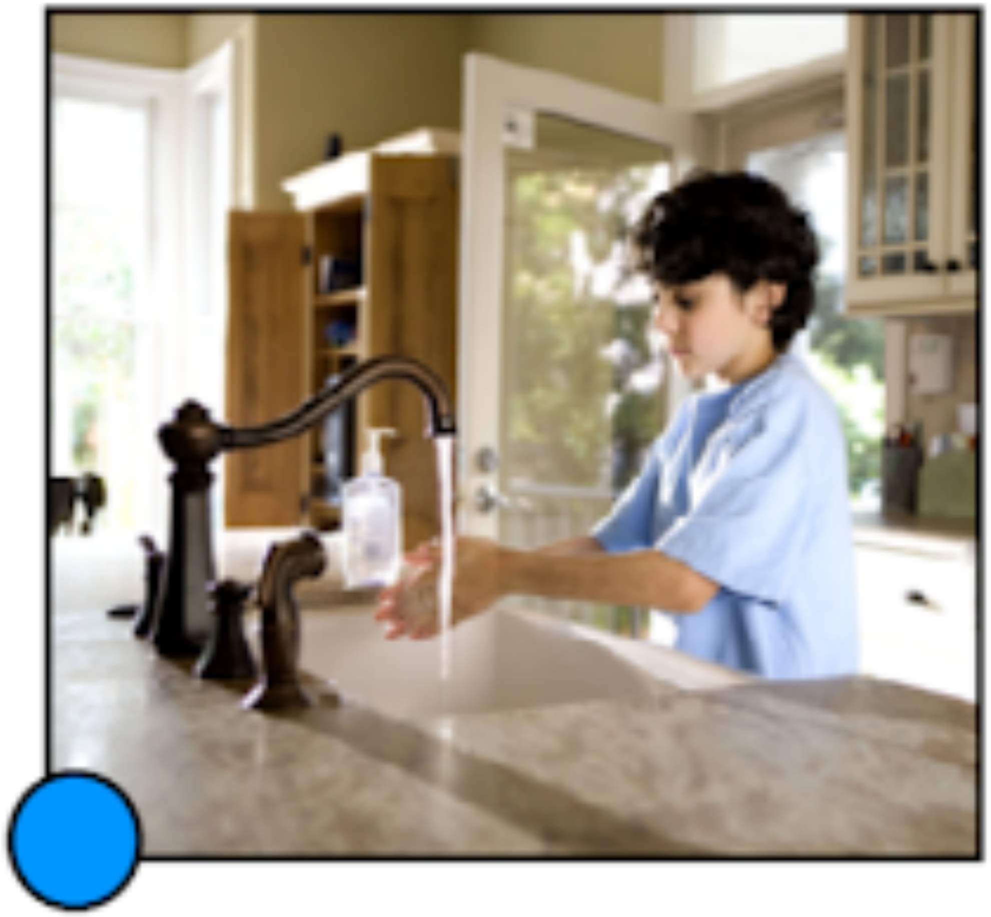


by Mark Yatskar

Algorithmic Bias



woman cooking



man fixing faucet

by Mark Yatskar

Quantifying Dataset Bias

$$bias(activity, gender) = \frac{cooc(activity, gender)}{\Sigma_{gender' \in G} cooc(activity, gender')}$$

$b(o,g)$

by Mark Yatskar

Quantifying Dataset Bias

Training Gender Ratio (◆ verb)

Training Set

◆ cooking

🟠 woman

🔵 man

◆ COOKING

ROLES	NOUNS
🟠 AGENT	woman
FOOD	stir-fry

◆ COOKING

ROLES	NOUNS
🔵 AGENT	man
FOOD	noodle

$$\frac{\#(\blacklozenge\ \text{cooking}\ ,\ \text{man})}{\#(\blacklozenge\ \text{cooking}\ ,\ \text{man}) + \#(\blacklozenge\ \text{cooking}\ ,\ \text{woman})} = 1/3$$

by Mark Yatskar

Quantifying Dataset Bias: Dev Set

Predicted Gender Ratio (◆ verb)

Development Set

◆ cooking

🟠 woman

🔵 man

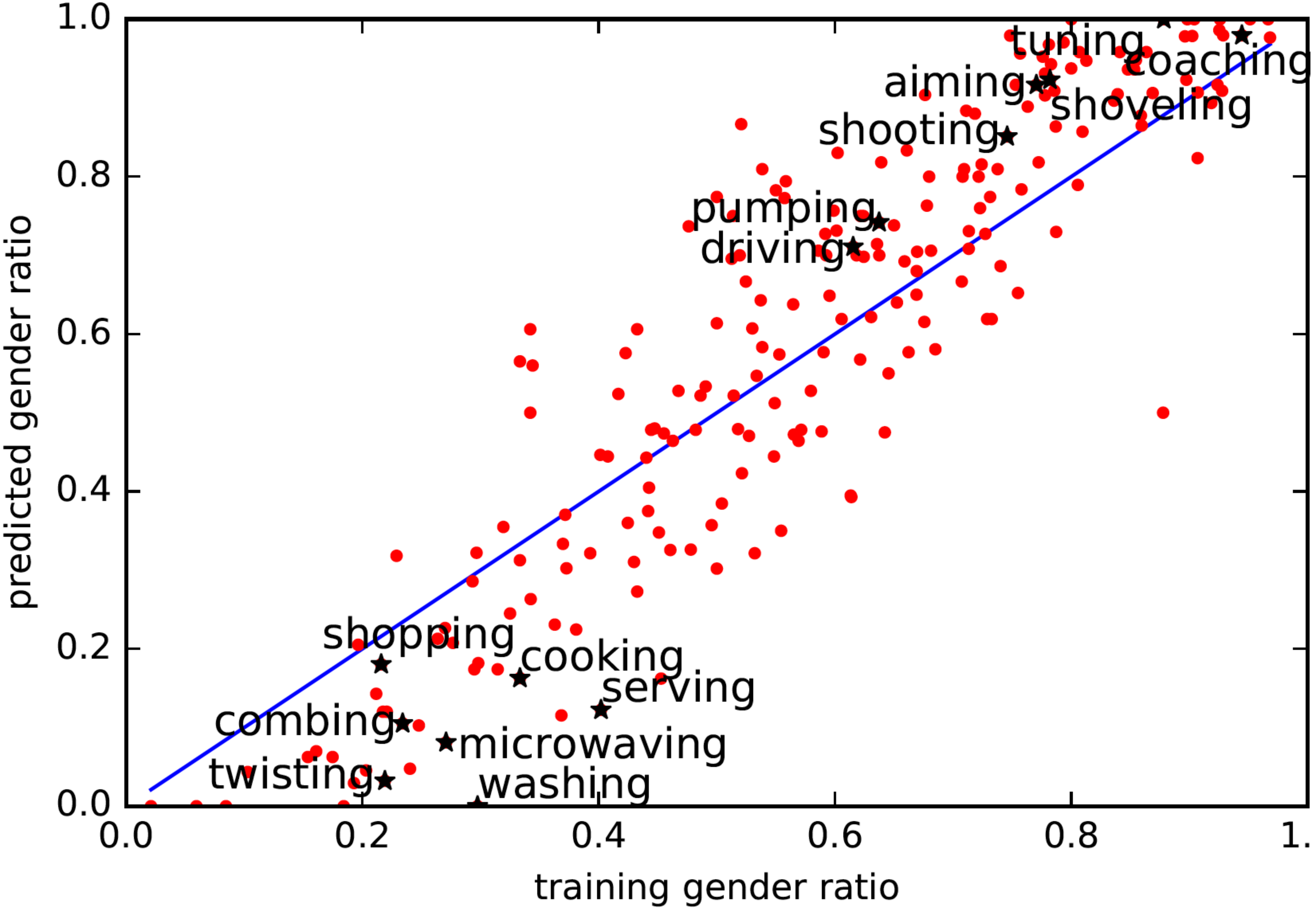
◆ COOKING	
ROLES	NOUNS
🟠 AGENT	woman
FOOD	stir-fry

◆ COOKING	
ROLES	NOUNS
🔵 AGENT	man
FOOD	noodle

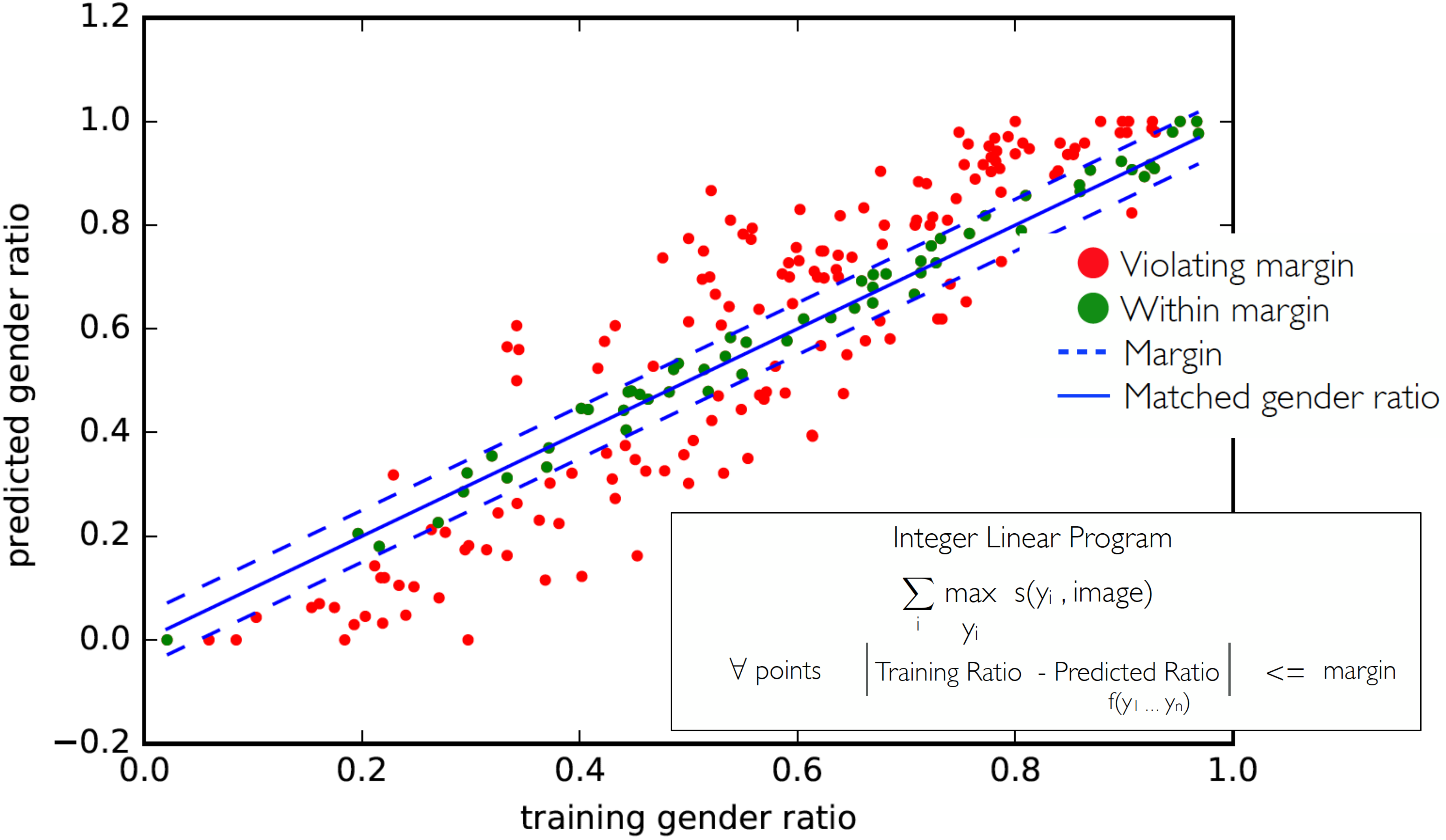
$$\frac{\#(\text{◆ cooking},\ \text{🔵 man})}{\#(\text{◆ cooking},\ \text{🔵 man}) + \#(\text{◆ cooking},\ \text{🟠 woman})} = 1/6$$

by Mark Yatskar

Model Bias Amplification



Reducing Bias Amplification (RBA)



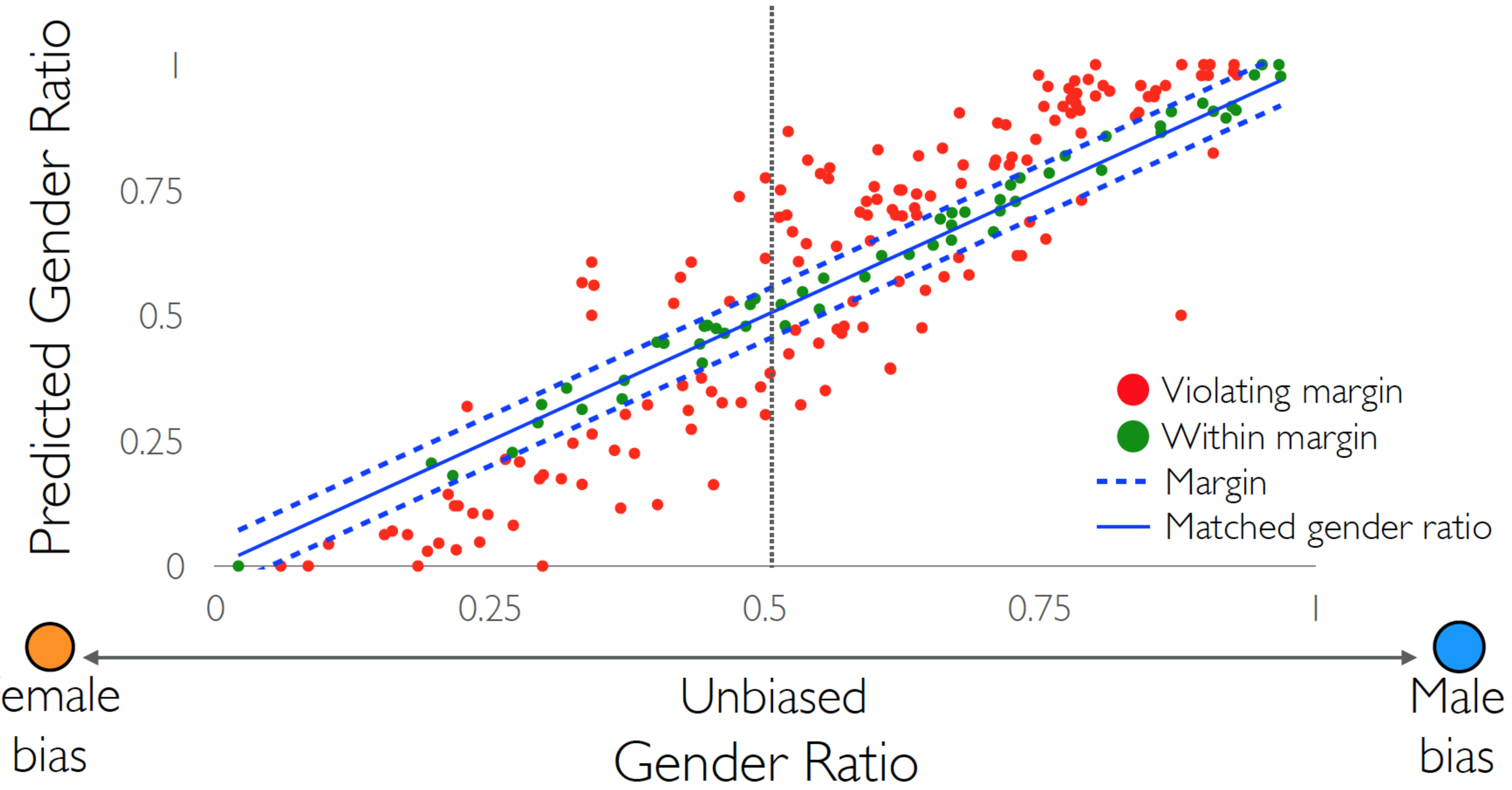
Integer Linear Program

$$\sum_{i} \max_{y_i} \; s(y_i\,, \text{image})$$

$$\forall \text{ points} \quad \left| \text{Training Ratio} - \underset{f(y_1 \ldots y_n)}{\text{Predicted Ratio}} \right| <= \text{margin}$$

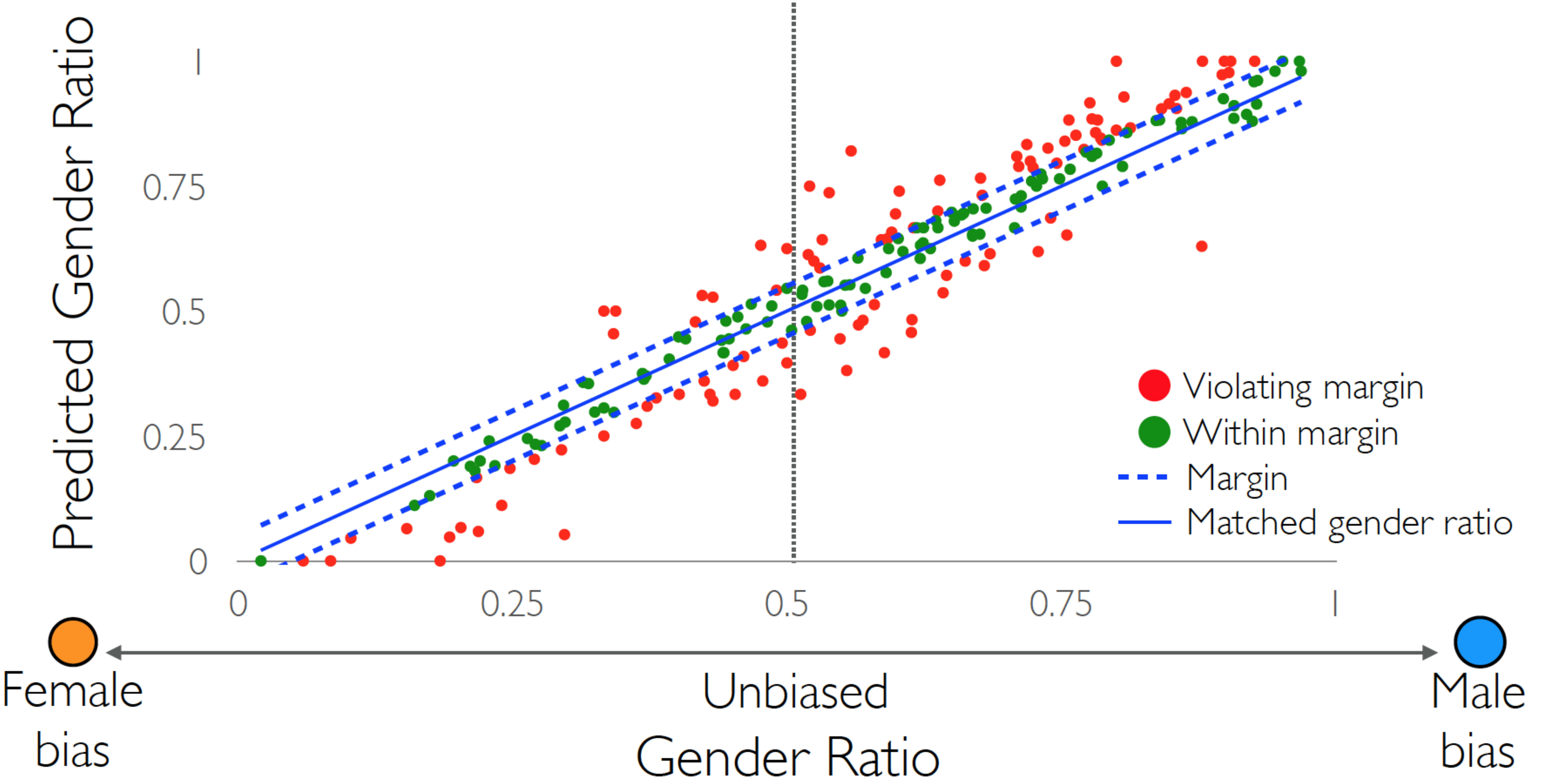
Results

imSitu Verb Violation: 72.6% .050 |bias↑| 24.07 acc.



Results

imSitu Verb	Violation: 72.6%	.050 |bias↑|	24.07 acc.
w/ RBA	Violation: 50.5%	.024 |bias↑|	23.97 acc.



Discussion

- Applications that are built from online data, generated by people, learn also real-world stereotypes
- Should our ML models represent the “real world”?
- Or should we artificially skew data distribution?
- If we modify our data, what are guiding principles on what our models should or shouldn't learn?

Considerations for Debiasing Data and Models

- Ethical considerations
  - Preventing discrimination in AI-based technologies
    - in consumer products and services
    - in diagnostics, in medical systems
    - in parole decisions
    - in mortgage lending, credit scores, and other financial decisions
    - in educational applications
    - in search → access to information and knowledge
- Practical considerations
  - Improving performance particularly where our model's accuracy is lower

exercise!