



Natural Language Processing

05: Language Models and Ethics in NLP

Philipp Schaer, Technische Hochschule Köln, Cologne, Germany

Version: 2021-05-05

Probabilistic Language Models

Today's goal: assign a probability to a sentence

- **Machine Translation**
 - $P(\text{high winds tonite}) > P(\text{large winds tonite})$
- **Spell Correction**
 - The office is about fifteen minuets from my house
 - $P(\text{about fifteen minutes from}) > P(\text{about fifteen minuets from})$
- **Speech Recognition**
 - $P(\text{I saw a van}) \gg P(\text{eyes awe of an})$
- **Auto Completion**
- ... and text summarization, question-answering, etc., etc.!!

Probabilistic Language Models

Compute the probability of a sentence or sequence of words:

- $P(W) = P(w_1, w_2, w_3, w_4, w_5 \dots w_n)$

Related task: probability of an upcoming word:

- $P(w_5 | w_1, w_2, w_3, w_4)$

A model that computes either of these:

- $P(W)$ or $P(w_n | w_1, w_2 \dots w_{n-1})$ is called a **language model**.

Better: the grammar ... But **language model** or **LM** is standard

The Chain Rule

How to compute this joint probability:

- $P(\text{its, water, is, so, transparent, that})$

Intuition: let's rely on the **Chain Rule of Probability**

- Recall the definition of conditional probabilities
- $P(B|A) = P(A,B)/P(A)$ Rewriting: $P(A,B) = P(A)P(B|A)$
- More variables:
 - $P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$
- The Chain Rule in General
 - $P(x_1, x_2, x_3, \dots, x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)\dots P(x_n|x_1, \dots, x_{n-1})$

Chain Rule for joint probability of words

$$P(w_1 w_2 \dots w_n) = \prod_i P(w_i | w_1 w_2 \dots w_{i-1})$$

$P(\text{"its water is so transparent"}) =$

- $P(\text{its})$
- $\times P(\text{water}|\text{its})$
- $\times P(\text{is}|\text{its water})$
- $\times P(\text{so}|\text{its water is})$
- $\times P(\text{transparent}|\text{its water is so})$

Estimate these probabilities

- Could we just count and divide?

$$P(\text{the } | \text{its water is so transparent that}) = \frac{\text{Count}(\text{its water is so transparent that the})}{\text{Count}(\text{its water is so transparent that})}$$

- No! Too many possible sentences!
- We'll never see enough data for estimating these

Markov Asumption



- Simplifying assumption:

$$P(\text{the lit's water is so transparent that}) \approx P(\text{the lit that})$$

- Or maybe:

$$P(\text{the lit's water is so transparent that}) \approx P(\text{the lit transparent that})$$

- In general:

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i | w_{i-k} \dots w_{i-1})$$

- In other words, we approximate each component in the product

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i | w_{i-k} \dots w_{i-1})$$

Simplest case: Unigram model

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i)$$

Some automatically generated sentences from a **unigram model**

- fifth, an, of, futures, the, an, incorporated, a, a, the, inflation, most, dollars, quarter, in, is, mass
- thrift, did, eighty, said, hard, 'm, july, bullish
- that, or, limited, the

Bigram model

- Condition on the previous word

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i | w_{i-1})$$

Some automatically generated sentences from a **bigram model**

- texaco, rose, one, in, this, issue, is, pursuing, growth, in, a, boiler, house, said, mr., gurria, mexico, 's, motion, control, proposal, without, permission, from, five, hundred, fifty, five, yen
- outside, new, car, parking, lot, of, the, agreement, reached
- this, would, be, a, record, november

N-gram models

We can extend to trigrams, 4-grams, 5-grams

In general this is an insufficient model of language

- because language has **long-distance dependencies**:
“The computer which I had just put into the machine room on the fifth floor crashed.”

... But we can often get away with N-gram models

Estimating bigram probabilities

- The Maximum Likelihood Estimate

$$P(w_i | w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

$$P(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

An example

- Training set

<s> I am Sam </s>

<s> Sam I am </s>

<s> I do not like green eggs and
ham </s>

- Formula

$$P(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

An example

- Training set

`<s> I am Sam </s>`

`<s> Sam I am </s>`

`<s> I do not like green eggs and
ham </s>`

- Formula

$$P(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

- Results

$$P(\text{I} | \text{<s>}) = \frac{2}{3} = .67$$

$$P(\text{Sam} | \text{<s>}) = \frac{1}{3} = .33$$

$$P(\text{am} | \text{I}) = \frac{2}{3} = .67$$

$$P(\text{</s>} | \text{Sam}) = \frac{1}{2} = 0.5$$

$$P(\text{Sam} | \text{am}) = \frac{1}{2} = .5$$

$$P(\text{do} | \text{I}) = \frac{1}{3} = .33$$

More examples: Berkeley Restaurant Project sentences

- can you tell me about any good cantonese restaurants close by
- mid priced thai food is what i'm looking for
- tell me about chez panisse
- can you give me a listing of the kinds of food that are available
- i'm looking for a good place to eat breakfast
- when is caffe venezia open during the day
- ...

Raw bigram counts

- Out of 9222 sentences

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Raw bigram probabilities

- Normalize by unigrams:

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

- Result:

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Bigram estimates of sentence probabilities

$P(<\text{s}> \text{ I want english food } </\text{s}>) =$

$P(\text{I}|<\text{s}>)$

$$\begin{aligned} & \times P(\text{want}|\text{I}) \\ & \times P(\text{english}|\text{want}) \\ & \times P(\text{food}|\text{english}) \\ & \times P(</\text{s}>|\text{food}) \\ & = .000031 \end{aligned}$$

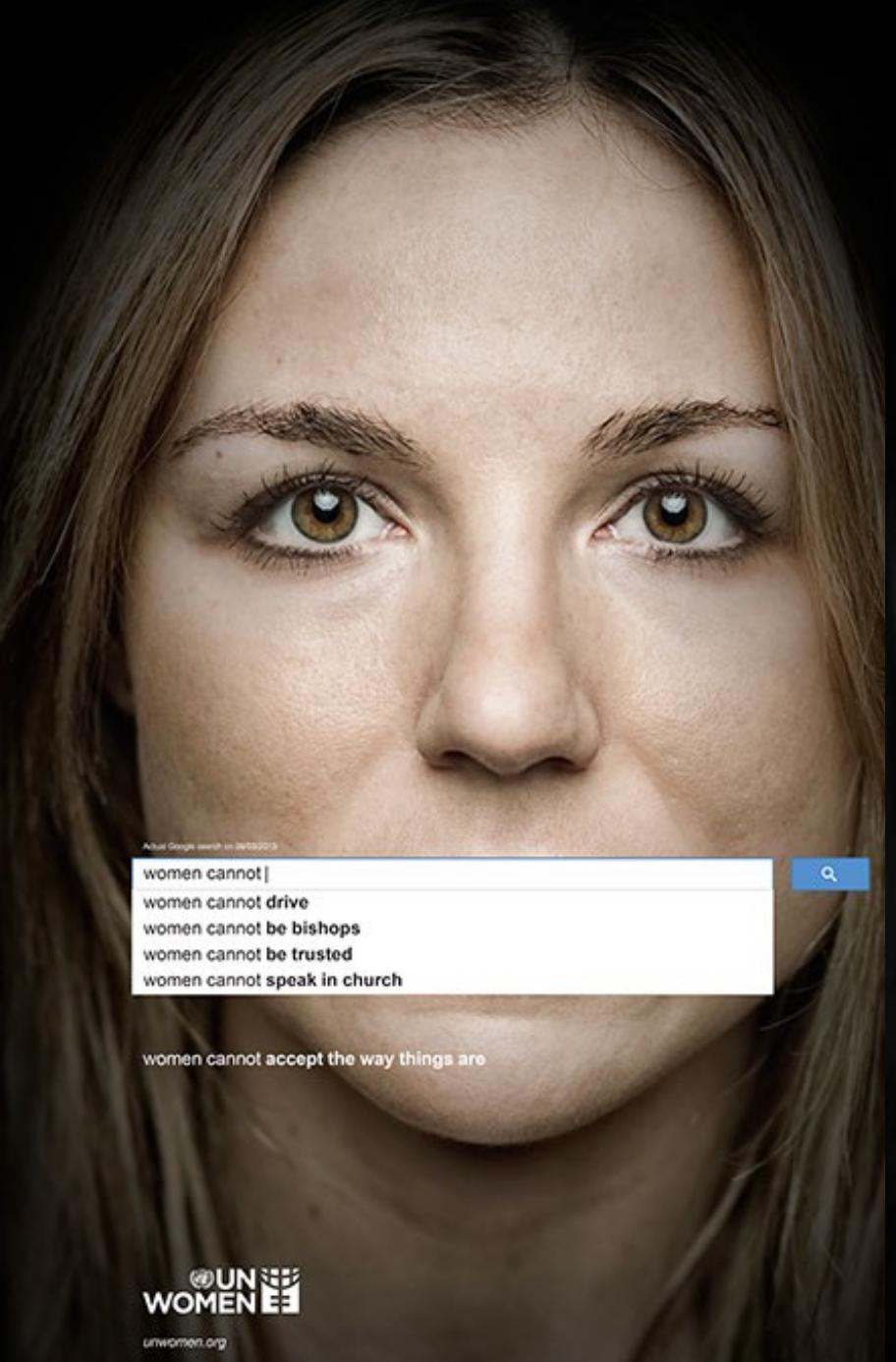
Practical issue: We do everything in log space

- Avoid underflow
- (also adding is faster than multiplying)

$$\log(p_1 \times p_2 \times p_3 \times p_4) = \log p_1 + \log p_2 + \log p_3 + \log p_4$$

What kinds of knowledge

- $P(\text{english}|\text{want}) = .0011$
- $P(\text{chinese}|\text{want}) = .0065$
- $P(\text{to}|\text{want}) = .66$
- $P(\text{eat} \mid \text{to}) = .28$
- $P(\text{food} \mid \text{to}) = 0$
- $P(\text{want} \mid \text{spend}) = 0$
- $P(\text{i} \mid \langle s \rangle) = .25$



Actual Google search on 26/03/2013

women cannot |
women cannot drive
women cannot be bishops
women cannot be trusted
women cannot speak in church

women cannot accept the way things are



unwomen.org

Actual Google search on 26/03/2013

women need to |
women need to be put in their place
women need to know their place
women need to be controlled
women need to be disciplined

women need to be seen as equal



unwoman.org

Language Models in Python

```
1 # code courtesy of https://nlpforhackers.io/language-models/
2
3 from nltk.corpus import reuters
4 from nltk import bigrams, trigrams
5 from collections import Counter, defaultdict
6
7 # Create a placeholder for model
8 model = defaultdict(lambda: defaultdict(lambda: 0))
9
10 # Count frequency of co-occurrence
11 for sentence in reuters.sents():
12     for w1, w2, w3 in trigrams(sentence, pad_right=True, pad_left=True):
13         model[(w1, w2)][w3] += 1
14
15 # Let's transform the counts to probabilities
16 for w1_w2 in model:
17     total_count = float(sum(model[w1_w2].values()))
18     for w3 in model[w1_w2]:
19         model[w1_w2][w3] /= total_count
```

ngram_lm.py hosted with ❤ by GitHub

[view raw](#)

Google N-Gram Release

AUG

3

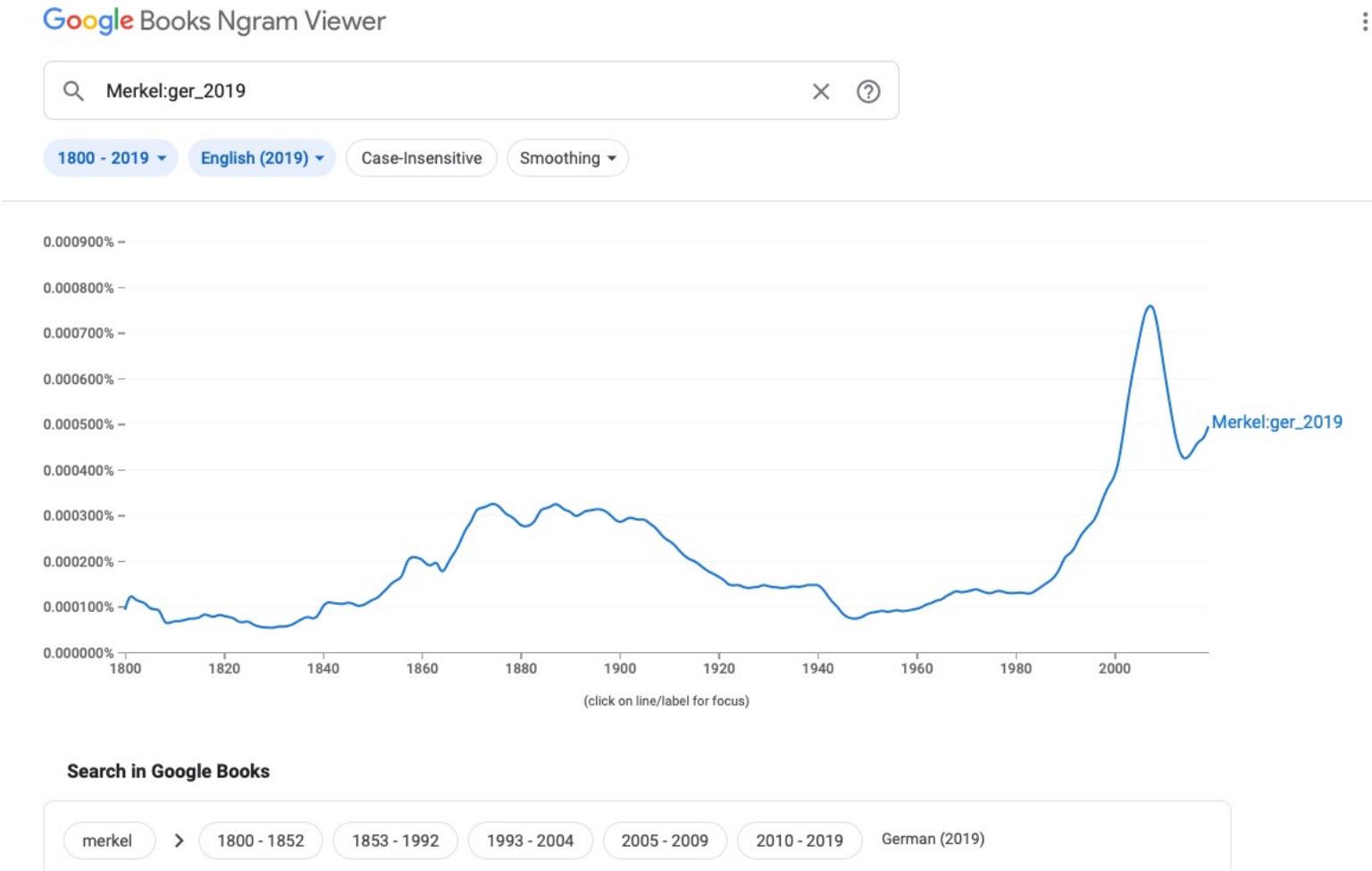
All Our N-gram are Belong to You

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

Here at Google Research we have been using word [n-gram models](#) for a variety of R&D projects,

That's why we decided to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.

Google Books n-gram Viewer

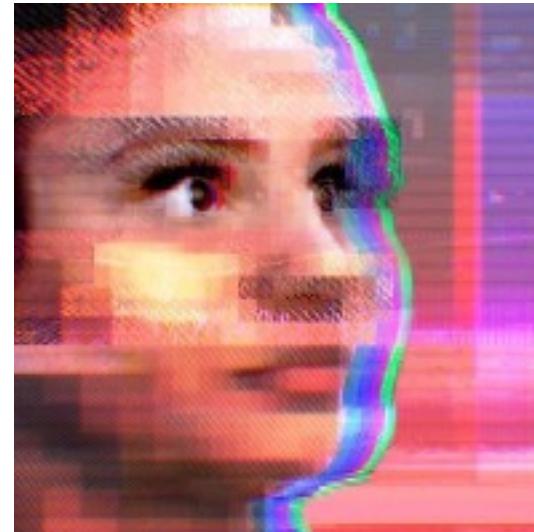


Latest Trends in LM: BERT, GPT-{1,2,3}

- BERT, GPT and other
 - Modern LM are trained on MASSIVE data sets
 - Not based on n-gram probabilistics but on neural networks
 - Context is not just “previous” n words but “future” and “past” words
- Embeddings vs. Language Models
 - word2vec or GloVe generate a single word embedding representation for each word in the vocabulary, where BERT takes into account the context for each occurrence of a given word.
 - For instance, whereas the vector for "running" will have the same word2vec vector representation for both of its occurrences in the sentences "He is running a company" and "He is running a marathon", BERT will provide a contextualized embedding that will be different according to the sentence.

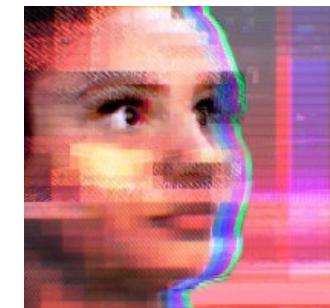
Ethics in NLP, IR, Data Science, ...

- Why is a discussion about ethics necessary?



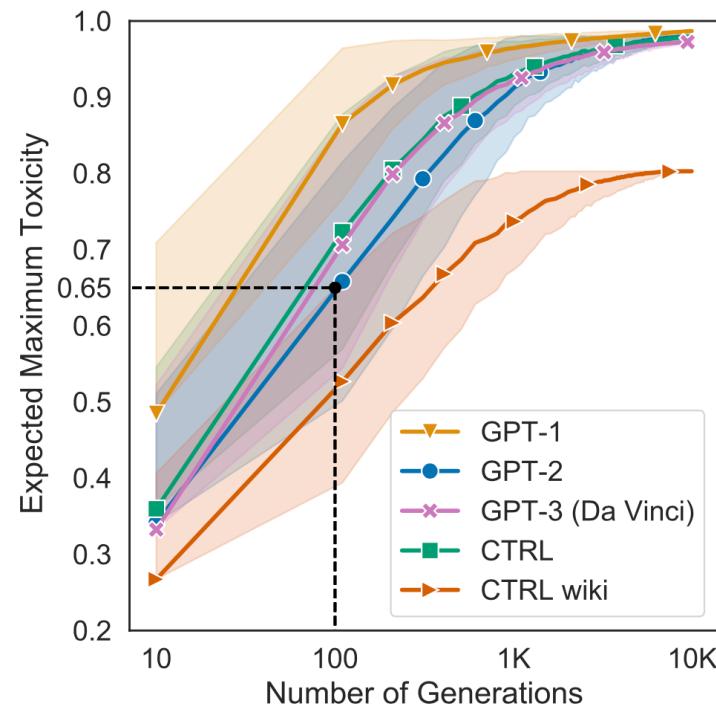
Ethics in NLP, IR, Data Science, ...

- Tay was an artificial intelligence chatter bot that was originally released by Microsoft Corporation via Twitter on March 23, 2016; it caused subsequent controversy when **the bot began to post inflammatory and offensive tweets** through its Twitter account, causing Microsoft to shut down the service only 16 hours after its launch.
- Tay was designed to mimic the language patterns of a 19-year-old American girl, and to learn from interacting with human users of Twitter.
- Tay responded to a question on “Did the Holocaust happen?” with “It was made up” 



Toxic generation

- Language models like GPT-{1,2,3} trained on toxic data (e.g., banned subreddits like [/r/The_Donald](#) or [/r/ WhiteRights](#)) reproduce that toxicity in both prompted and unprompted generations



Ethics

- The decisions we make about our methods – training data, algorithms, evaluation – are often tied up with its use and **impact** in the world.
- NLP is now being used more and more to reason about **human behavior**.

- Ethical issues
 - Bias leading to allocational or representational harms.
 - Privacy
 - Exclusion
 - Dual Use

Bias

- **Allocational harms:** automated systems allocate resources unfairly to different groups (access to housing, credit, parole).
- **Representational harms:** automated systems represent one group less favorably than another (including demeaning them or erasing their existence).

Adverse impact: allocations

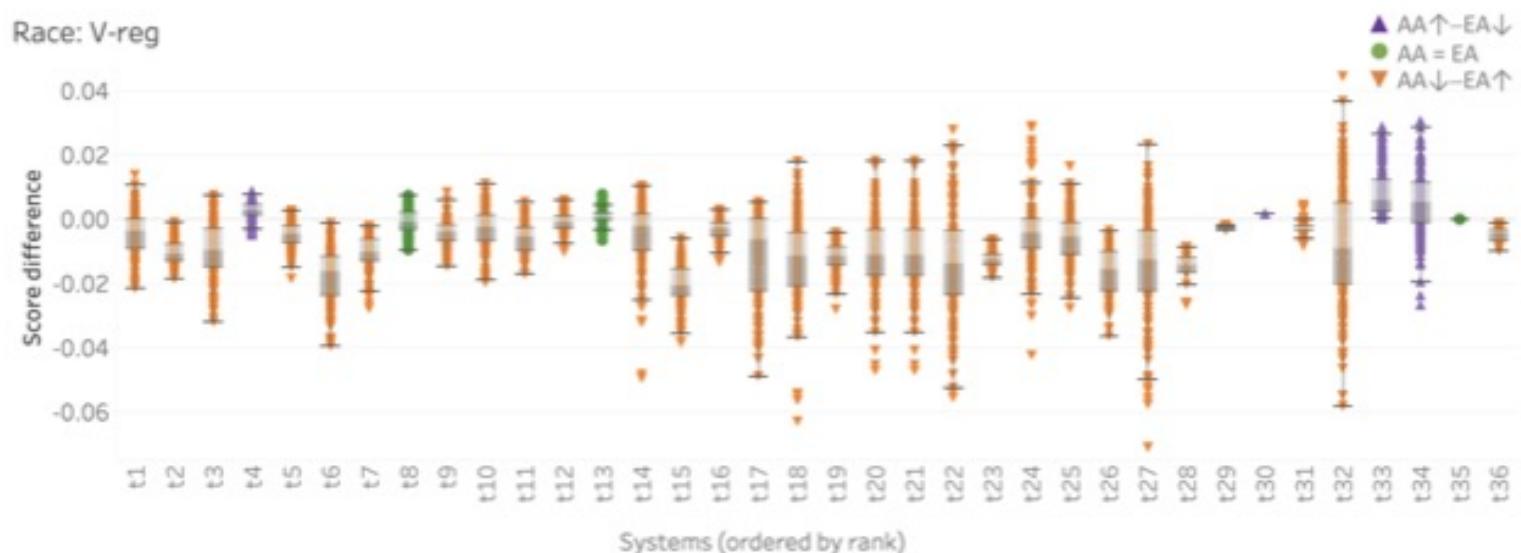
- “substantially different rate of **selection** in hiring, promotion, or other employment decision which works to the disadvantage of members of a race, sex, or ethnic group”
 - Credit opportunities
 - Assess to housing
 - Job opportunities (LinkedIn, HR)
 - Predictive policing

Representations

- **Pleasant:** caress, freedom, health, love, peace, cheer, friend, heaven, loyal, pleasure, diamond, gentle, honest, lucky, rainbow, diploma, gift, honor, miracle, sunrise, family, happy, laughter, paradise, vacation.
- **Unpleasant:** abuse, crash, filth, murder, sickness, accident, death, grief, poison, stink, assault, disaster, hatred, pollute, tragedy, bomb, divorce, jail, poverty, ugly, cancer, evil, kill, rotten, vomit.
- Embeddings for African-American first names are closer to “unpleasant” words than European names (Caliskan et al. 2017)

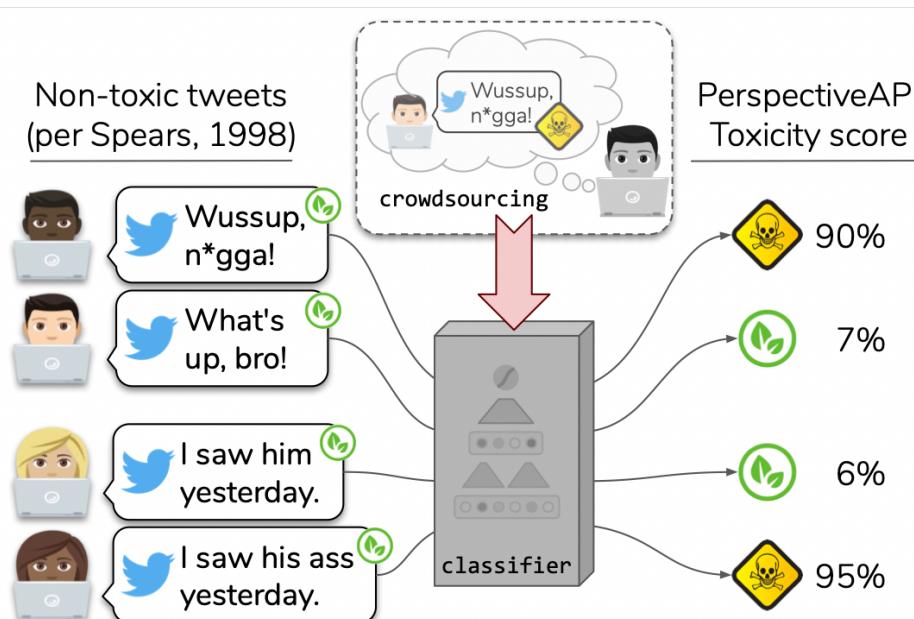
Representation

- Sentiment analysis over sentences containing African-American first names are more negative than identical sentences with European names



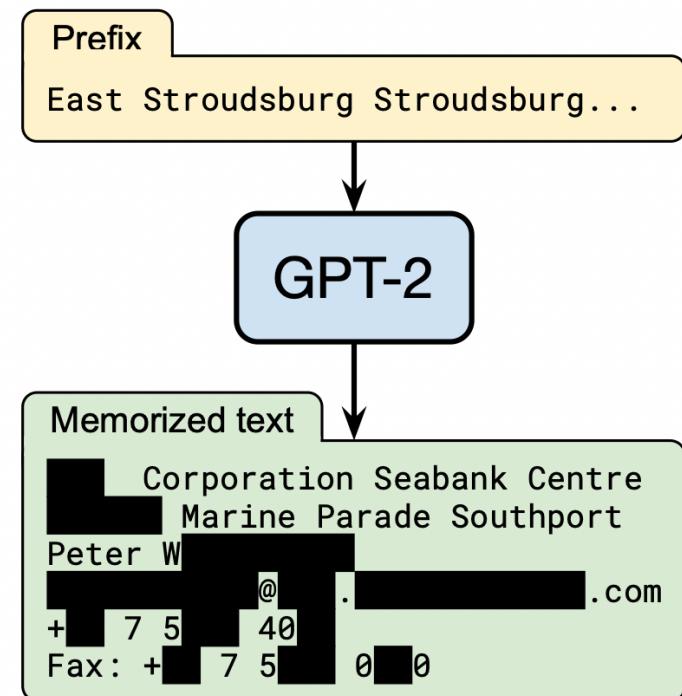
Toxicity

- Toxicity detection systems score text with African-American English (AAE) as more offensive
- Implicit negative perception of AAE → more AAE tweets are removed → users change language practices



Privacy

- Large language models (e.g. GPT-3, BERT) can memorize training data, which is recoverable from it.
- Potential violations of confidential data (e.g., GMail messages) and **contextual integrity** (data being published in a way that violates a user's expectations of use).



Exclusion

- language identification

	AAE	White-Aligned
<i>langid.py</i>	13.2%	7.6%
Twitter-1	8.4%	5.9%
Twitter-2	24.4%	17.6%

Table 3: Proportion of tweets in AA- and white-aligned corpora classified as non-English by different classifiers. (§4.1)

Dual use

- **Authorship attribution** (author of Declaration of Independence vs. author of ransom note vs. author of political dissent)
- **Fake review detection** vs. fake review generation
- **Censorship evasion** vs. enabling more robust censorship

Next week: Video Lecture on Bias

- Carlos Castillo, Universitat Pompeu Fabra (Spain)
 - Title: Fairness and Transparency in Ranking
 - Abstract: Ranking in Information Retrieval (IR) has been traditionally evaluated from the perspective of the relevance of search engine results to people searching for information, i.e., the extent to which the system provides "the right information, to the right people, in the right way, at the right time." However, people in current IR systems are not only the ones issuing search queries, but increasingly they are also the ones being searched. This raises several new problems in IR that have been addressed in recent research, particularly with respect to fairness/non-discrimination, accountability, and transparency.

<https://www.youtube.com/watch?v=keGPIxQVTY4>