Adversarial Removal of Demographic Attributes from Text Data

Yanai Elazar and Yoav Goldberg
Bar-Ilan University / NLP Group
November 2, 2018
Text is used for predictions
Motivation

• For example, consider a text classification setup, where we predict:
  • Hiring decisions
  • Mortgages approval
  • Loans rates
This applicant would easily get any NLP job
Motivation

The common implementation:

Input CV → ML Model → Hire / Don’t Hire
The common implementation:

Encode → Representation → Predict

Input CV

Hire

Don’t Hire
Motivation

Amazon built an AI tool to hire people but had to shut it down because it was discriminating against women.

But then we see this...
Motivation

• When deciding on recruiting an applicant based on their writings/CV...
• ...we would like that attributes like the author’s:
  • Gender
  • Race
  • Age
• won’t be part of the decision.
• In some places, this is even illegal
Motivation

• We seek to build models which are:
  • Predictive for some main task (e.g. Hiring decision)

  • Agnostic to irrelevant/protected attributes (e.g. race, gender, …)
Motivation

How do we know we do not condition on some sensitive attribute by mistake?

Input CV

Encode

Representation

Predict

Hire

Don’t Hire
Motivation

If we can predict protected attributes from the representation...

A talented candidate might suffer from demographic discrimination
Motivation

If we **can not** predict protected attributes from the representation...

We don’t condition on these protected attributes and...
A talented candidate won’t suffer from demographic discrimination
In this work:

we do not have access to sensitive tasks like Hiring decisions.

we focus on other tasks, less sensitive
Let's predict... EMOJIS

We use DeepMoji.

DeepMoji is a model for predicting Emojis from tweets

Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm

Bjarke Felbo¹, Alan Mislove², Anders Søgaard³, Iyad Rahwan¹, Sune Lehmann⁴

¹Media Lab, Massachusetts Institute of Technology
²College of Computer and Information Science, Northeastern University
³Department of Computer Science, University of Copenhagen
⁴DTU Compute, Technical University of Denmark
Text classification - Example

Let's predict... EMOJIS

I love mom's cooking

- 😊: 49.1%
- 😍: 8.8%
- ❤️: 3.1%
- 😁: 3.0%
- ❤️: 2.9%

I love how you never reply back..

- 😞: 14.0%
- 😞: 8.3%
- 😞: 6.3%
- 😞: 5.4%
- 😞: 5.1%

I love cruising with my homies

- 😎: 34.0%
- 👌: 6.6%
- 🍾: 5.7%
- 😁: 4.1%
- 🏆: 3.8%

I love messing with yo mind!!

- 😞: 17.2%
- 😈: 11.8%
- 😊: 8.0%
- 😇: 6.4%
- 🦉: 5.3%

I love you and now you're just gone..

- 😢: 39.1%
- 😞: 11.0%
- 😞: 7.3%
- 😞: 5.3%
- 😞: 4.5%

This is shit

- 😞: 7.0%
- 😞: 6.4%
- 😞: 6.0%
- 😞: 6.0%
- 😞: 5.8%

This is the shit

- 🎧: 10.9%
- 🎵: 9.7%
- 🤗: 6.5%
- 😁: 5.7%
- 😁: 4.8%

Deep Moji (Felbo et al., 2017)
Let's predict... EMOJIS

- DeepMoji is a strong and expressive model
- It also creates powerful representations

Deep Moji (Felbo et al., 2017)
Text classification - Example

Let's predict... EMOJIS

- DeepMoji is a strong and expressive model
- It also creates powerful representations
- Achieved several SOTA results on text classification
Text classification - Example

Let's predict... EMOJIS

Does this representation also contain information on sensitive attributes?

Encode → Predict

Race
Gender
Age

Emoji distributions:
- 😞: 7.0%
- 😞: 6.4%
- 😞: 6.0%
- 😞: 6.0%
- 😞: 5.8%
Setup

Task (Emojis)

We take the representation that predict Emojis

DeepMoji (Felbo et al., 2017)

$$h(x)$$

Embeddings

Encoder

Representation

Classifier

I love messing with yo mind

$$x$$
We take the representation that predict Emojis

And use them to predict demographics.

We define: \textit{leakage} = score above a random guess an "Attacker" achieves
Text Leakage – Case Study

- We use DeepMoji encoder, to encode tweets, from 3 datasets, all binary and balanced.

- Each dataset is tied to a different demographic label.

- We then train Attackers to predict these attributes.
Text Leakage – Case Study

The dev-set scores above chance level are quite high

Big Surprise?
Not really.
This is the core idea in Transfer-Learning. We’ve seen its benefits in pretrained embeddings, language models etc.
Text Leakage – Case Study

• Why do we get this major “help” in predicting other attributes than those we trained for?
• One option is the correlation between attributes in the data.

Fair enough. Let’s control for it.
Controlled Setup
New setup

• We use Twitter data

• We focus on sentiment prediction, emoji based

• With *Race*, *Gender* and *Age* as protected attributes

*Blodgett et al., 2016*  *Rangel et al., 2016*  *Rangel et al., 2016*
New setup

Balanced Dataset

Task (Sentiment)

Demographics

50% Male
50% Female

50% Positive
50% Negative
Balanced Training

Training our own encoder on the balanced datasets

Main Task (sentiment)

Classifier

Representation

Encoder

Embeddings

I love messing with yo mind
Balanced Training

And using the Attacker to check for leakage

Protected Attribute (gender) $\text{att}(h(x))$

Trainable

Encoder

Representation $h(x)$

Embeddings

I love messing with yo mind

$\times$
Balanced Training - Leakage

We wanted to see something like this:

But instead...
The Attacker manages to extract a substantial amount of sensitive information

Even in a balanced setup, leakage exists
Our objective

- Create a representation which:
- Is predictive of the main task (e.g. sentiment)
Our objective

• Create a representation which:
  • Is not predictive of protected attribute (e.g. gender, race)
  • Is predictive of the main task (e.g. sentiment)

and

• Is not predictive of protected attribute (e.g. gender, race)
Our objective

• Interesting technical problem – How to *unlearn* something?
• Interesting technical problem – *Can we* *unlearn* something?
Actively Reducing Leakage
Adversarial Setup (Ganin and Lempitsky, 2015)

\[ f(h(x)) \]

Classifer 1 (Main Task)

Encoder

Representation

Embeddings

\[ h(x) \]

\[ \text{gradient reversal layer} \]

Classifier 2 - Adv (Protected Attribute)

\[ \text{Remove stuff from representation} \]

\[ -\lambda \frac{\partial L_{adv}}{adv(h(x))} \]

I love messing with yo mind
Does it work?

“Does it work?”

“I love mom’s cooking”
Does it work?

“Does it work?”

“I love mom’s cooking”

Successfully removed demographics?
Does it work?

During adversary training the demographic information seems to be gone (close to chance)

IS THAT SO?
Does it work? Not so quickly...

When training the Attacker

We can still recover a considerable amount of information
Does it work? Not so quickly...

Consistent across tasks and protected attributes
Does it work? more or less

Well, the adversarial method does help. But not enough
While effective during training, in test time, the adversarial do not remove all the protected information.
Can we make stronger adversaries?
Stronger, Better, Bigger???

Baseline

$f(h(x))$ (Main Task)

Classifier 1

Representation

Encoder

Embeddings

Classifier 2 - Adv (Protected Attribute)

gradient reversal layer

$adv(h(x))$

$I love messing with yo mind$
More Parameters!

Classifiers:
- Classifier 1 (Main Task): $f(h(x))$
- Classifier 2 (Protected Attribute): $adv(h(x))$

Encoder

Representation

Embeddings

Gradient reversal layer

Advantage:
$$\lambda \frac{\partial L_{adv}}{adv(h(x))}$$

Text:
I love messing with yo mind

Equation:
$$ad(h(x))$$
Stronger, Better, Bigger???

Baseline

Classifer 1 (Main Task)

Classification

Representation

Encoder

Embeddings

I love messing with yo mind

Classifier 2 - Adv (Protected Attribute)

gradient reversal layer

\[
\text{adv}(h(x))
\]

\[
\frac{\partial L_{adv}}{\text{adv}(h(x))}
\]
Stronger, Better, Bigger???

Bigger Weight!

Classifier 1 (Main Task)

Classifier 2 - Adv (Protected Attribute)

Gradient reversal layer

Scale the reverse gradients

\[ \frac{\partial L_{adv}}{adv(h(x))} \]

Embeddings

Encoder

Representation

\( h(x) \)

\( f(h(x)) \)

I love messing with yo mind
Stronger, Better, Bigger???

Baseline

Classifier 1 (Main Task)

Representation

Encoder

Embeddings

I love messing with yo mind

Classifier 2 - Adv (Protected Attribute)

gradient reversal layer

\[
f(h(x))
\]

\[
adv(h(x))
\]

\[
\lambda \frac{\partial L_{adv}}{adv(h(x))}
\]
More Adversaries!

Classifier 1 - Main Task

f(h(x))

Classifier 2 - Adv (Protected Attribute)

Classifier 3 - Adv (Protected Attribute)

gradient reversal layer

adv(h(x))

\[ \frac{\partial L_{adv_i}}{adv(h(x))} \]

\[ -\lambda \sum_i \]

I love messing with yo mind
Stronger, Better, Bigger???
Stronger, Better, Bigger???

Better, but still not perfect
Error Analysis
• What are the hard cases, which slip the adversary?
  • We trained the adversarial model 10 times (with random seeds)
  • then, trained the Attacker on each model
  • We collected all examples, which were consistently labeled correctly
Persistent Examples

• What are the hard cases, which slip the adversary?
Persistent Examples

AAE ("non-hispanic blacks")

Enjoy y'all day

_ Naw im cool

My Brew Eatting

My momma Bestfrand died

Tonight was cool

SAE ("non-hispanic whites")

I want to be tan again

Why is it so hot in the house?!

I want to move to california

I wish I was still in Spain

Ahhhh so much homework.

More about the leakage origin can be found in the paper
Summary

● When training a text encoder for some task
  ○ Encoded vectors are also useful for predicting other things (“transfer learning”)
  ○ Including things we did not want to encode (“leakage”)

● It is hard to completely prevent such leakage
  ○ Do not blindly trust adversarial training
  ○ Verify your model using an “Attacker”

Thank you
Wait. I remember this thing called Overfitting

- We still have a problem
  - During training it seems that the information was removed
  - But the Attacker tells us another story
- Everything we reported was on the dev-set
- Is it possible that we just overfitted on the training-set?
Wait. I remember this thing called Overfitting

● “Adversary overfitting”:
  ○ Memorizing the training data
  ○ By removing all its sensitive information
  ○ While leaking in test time
Wait. I remember this thing called Overfitting

We trained on 90% on the “overfitted” training set, and tested the remaining 10%

Training Set

new Train

90%

10%

new Dev

It is more than that
Few words about fairness

• Throughout this work, we aimed in achieving zero leakage, or in other words: *fairness by blindness*

• Many other definitions for “fairness” (>20)

• With 3 popular
  
  • *Demographic parity*
  
  • *Equality of Odds*
  
  • *Equality of Opportunity*

  In the paper, we prove that in our setup (balanced data) these definitions are identical.