Lab 10 ML: Weathermood Sentiment

Software Studio Datalab, CS, NTHU 2023

Gitlab

lab-weathermood-toxicity-detection



Outline

1. Sentiment analysis

- 2. TensorFlow.js
- 3. Weathermood toxicity detection

Sentiment analysis

- Natural language processing (NLP)
- Input = a piece of text or sentence
- Output = the score for the opinions and sentiments within



Sentiment analysis



Outline

- 1. Sentiment analysis
- 2. TensorFlow.js
- 3. Weathermood toxicity detection

Stand upon the shoulder of Google

TensorFlow: TensorFlow.js

- Develop ML models in JavaScript, and use ML directly in the browser or in Node.js
- tensorflow.org/js [link]



TensorFlow.js

TensorFlow.js models

Explore pre-trained models to add computer vision, natural language processing (NLP), and other common ML tasks to your web and browser-based applications.



Unlock new real-time experiences in the browser.

hands, and body with models from MediaPipe and beyond, optimized for JavaScript and Node.js.

Toxicity classifier

The toxicity model detects whether text contains toxic content

- Input = a sentence
- Output = whether the input contains toxic content

| text | identity attack | insult | obscene | severe toxicity | sexual explicit | threat | toxicity |
|---|--------------------|--------|---------|--------------------|--------------------|--------|----------|
| We're dudes on computers, moron. You are quite astonishingly stupid. | false | true | false | false | false | false | true |
| Please stop. If you continue to vandalize Wikipedia, as you did to Kmart, you will be blocked from editing. | false | false | false | false | false | false | false |
| I respect your point of view, and when this discussion originated on 8th April I would have tended to agree with you. | false | false | false | false | false | false | false |

Enter text below and click 'Classify' to add it to the table.



Weathermood w/ sentiment



How to use a pre-trained model

Installation

- npm install
- npm install @tensorflow/tfjs @tensorflow-models/toxicity

Usage, no key needed

- To import in npm

```
require('@tensorflow/tfjs');
const toxicity = require('@tensorflow-models/toxicity');
```

- Or as a standalone script tag

```
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"></script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/toxicity"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script><
```

```
// The minimum prediction confidence.
const threshold = 0.9;
// Load the model. Users optionally pass in a threshold and an array of
// labels to include.
toxicity.load(threshold).then(model => {
    const sentences = ['you suck'];
    model.classify(sentences).then(predictions => {
        console.log(predictions);
        /*
    });
});
```

```
▼ Array(7) 🚺
 > 0: {label: 'identity_attack', results: Array(1)}
 ▶ 1: {label: 'insult', results: Array(1)}
 ▼ 2:
     label: "obscene"
   results: Array(1)
    ▼ 0:
        match: false
      probabilities: Float32Array(2) ③
          0: 0.9990488886833191
          1: 0.0009510404779575765
        buffer: ArrayBuffer(8) (1)
          byteLength: 8
          byte0ffset: 0
          length: 2
          Symbol(Symbol.toStringTag): "Float32Array"
        [[Prototype]]: TypedArray
      > [[Prototype]]: Object
       length: 1
     > [[Prototype]]: Array(0)
   > [[Prototype]]: Object
 > 3: {label: 'severe_toxicity', results: Array(1)}
 4: {label: 'sexual_explicit', results: Array(1)}
 > 5: {label: 'threat', results: Array(1)}
 ▶ 6: {label: 'toxicity', results: Array(1)}
   length: 7
```



| text | identity attack | insult | obscene | severe toxicity | sexual explicit | threat | toxicity |
|---|--------------------|--------|---------|--------------------|--------------------|--------|----------|
| We're dudes on computers, moron. You are quite astonishingly stupid. | false | true | false | false | false | false | true |
| Please stop. If you continue to vandalize Wikipedia, as you did to Kmart, you will be blocked from editing. | false | false | false | false | false | false | false |
| I respect your point of view, and when this discussion originated on 8th April I would have tended to agree with you. | false | false | false | false | false | false | false |

Outline

- 1. Sentiment analysis
- 2. TensorFlow.js
- 3. Weathermood toxicity detection

lab-weathermood-toxicity-detection

Three types of mood

| identity attack | insult | obscene | severe toxicity | sexual explicit | threat | toxicity | emoji |
|--------------------|--------|---------|--------------------|--------------------|--------|----------|-------------|
| true | true | false | true | false | true | true | e sad |
| false | false | false | false | false | false | false | ee happy |
| false | false | true | false | true | false | false | fear |

lab-weathermood-toxicity-detection

Toxicity prediction workflow in weathermood

- 1. User triggers the post button, calling handlePost()
- 2. Call predict() from <u>api/toxicity-classifier</u> to start the text classification
- After getting the result, createPost() from api/posts was called to create a post with mood
- Each PostItem will render text and mood, mood is represented through emoji by calling <u>getMoodIcon()</u>

