Natural language processing: Sentiment analysis

What is natural language processing?

- Performing various tasks where the input is natural language
  - In a language that is commonly used by humans: English, Chinese, etc.
  - We usually assume that the language had been translated into writing
  - Understanding language from sound or signals are separate problems (e.g. speach understanding)
- Computers have long communicated in languages designed for communicating with them
  - Programming languages like Python, C or Java
  - This is not NLP!
- As language is the primary way in which humans communicate, NLP is very important!
  - The Turing test is an NLP problem!

The problem: Sentiment analysis

- My problem: given a text on a topic, identify whether it expresses a positive or negative sentiment
  - For instance, the review of a product or a movie.

Examples of reviews

First review:

> “This movie is a masterpiece. Every scene is beautifully shot, the performances are outstanding, and the story kept me completely engaged from start to finish. I left the theater amazed and couldn’t stop thinking about it afterward. Easily one of the best films I’ve seen in years.”

Second review:

> “This movie was terrible. The plot made no sense, the acting was painfully bad, and the pacing was so slow it felt endless. I was bored and frustrated the entire time and honestly wished it had ended much sooner.”

What makes it a sentiment analysis problem?

- **Inputs**:
  - A short text providing a review, expressing a political opinion etc.
- **Outputs**:
  - positive / negative
  - favorable / unfavorable
  - democrat / republican etc.
  - possibly, sliding scale

How do I know that it works?

- This is basically a classification or regression problem, so we are measuring it like any other problem like this
- **accuracy:** percentage of correctly classified inputs
    - Out of 100 examples I got right 90 --> 90% accuracy
- **false negatives**, **false positives** can all be considered
- What differentiates sentiment analysis is the **problem domain**.

An easy solution: dictionary matching

- The easiest way to perform sentiment analysis is to match **keywords** in the text.
- Associate some words with positive sentiment:
  - good, great, awesome, perfect, amazed, excited
- and others with negative
  - bad, awful, bored
- count how many words of each in the review, and calculate the difference
- does this work? yes, to some degree
  - it can be useful, for instance, for hate speech detection
  - easy to program, very fast, not quite "Artificial Intelligence"

But then...

> “This was a wonderfully long movie with an impressively simple story that repeated itself beautifully. The actors delivered consistently calm performances, giving every scene the same relaxing tone. By the end, I felt fully rested, as if the film had thoughtfully ensured there was nothing left to process or remember.”

Dictionary-based approaches cannot detect irony, satire, indirect speach, many figures of speech etc.

Neural network-based approaches

- Starting from 2010, a number of approaches started to use neural networks and deep learning for sentiment analysis
-

Let us read a paper

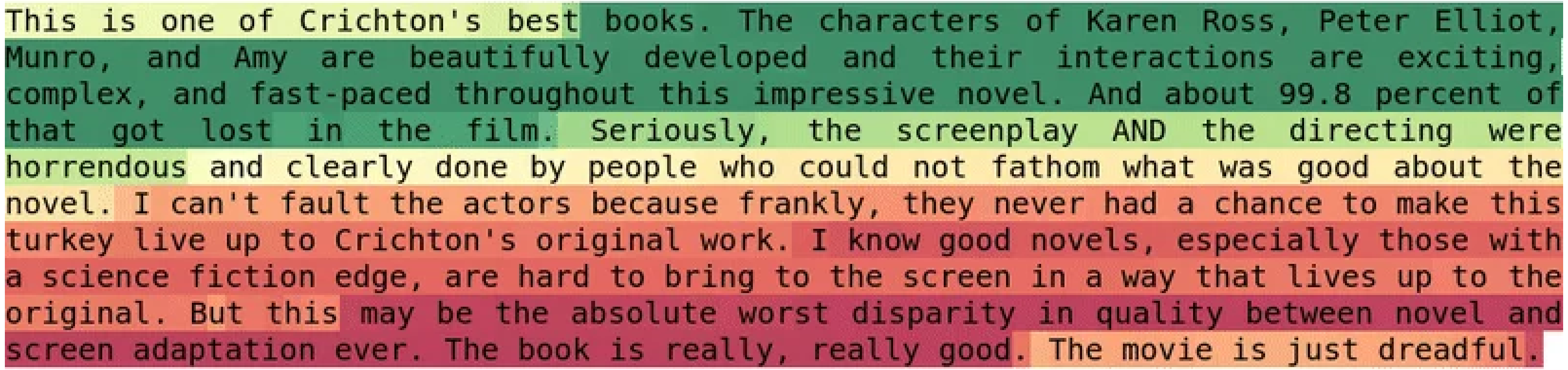
- "Learning to Generate Reviews and Discovering Sentiment"
  Alec Radford, Rafal Jozefowicz, Ilya Sutskever
  2017
- Where can I read it?
  - https://arxiv.org/pdf/1704.01444
  - https://openai.com/index/unsupervised-sentiment-neuron/
- How big the impact? How many citations it has?
  - http://scholar.google.com
- Who are the authors? What lab came from?
  - first author, last author

Couple of things to remember from this paper

- **What technology they used**: multiplicative LSTM (long-short term memory)
- **What they trained on?**: Amazon reviews (82 million reviews)
- **Biggest claim**: State of the art performance on the Stanford Sentiment Treebank bendmark
  - This is important, but usually not the most interesting thing about a paper, because the performance will be beaten by next year...
- **Most interesting thing**: The unsupervised sentiment neuron
  - What surprised me most about this paper?

Unsupervised sentiment neuron

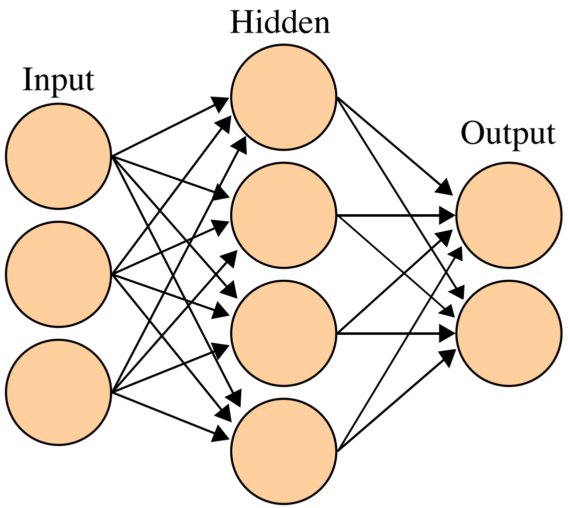
- The researchers when trained with L1 regularization, and analyzing the network, the researchers found a **single neuron** that corresponded to the sentiment!
  - They did not train for this!
  - It is known that L1 regularization creates sparse representations...
  - This is very, very sparse!
- The neuron tracks the evolution of sentiment letter-by-letter



An aside: neurons and weights

Neural networks and weights

- We have already seen neural networks in this class - for instance in the image classification lecture
  - We even trained them.



Source: Cburnett, CC BY-SA 3.0, via Wikimedia Commons

Neural networks and weights

- Neural networks are made of layers of neurons that are connected to each other
  - These connections have weights that are learned during training
  - The weights are just numbers that determine how much influence one neuron has on another: eg. +0.5 or -0.2
- After training, the weights are fixed, and they determine how the network processes inputs to produce outputs
  - We save the neural network by saving its structure and its weights

Interpretability of neural networks

- Neural networks are often considered "black boxes" because they have many parameters (weights) that are not easily interpretable
  - We don't know what each neuron does, or what each weight means, in a human-understandable way
- It is very rare that we can find a single neuron that corresponds to a specific concept, like sentiment
  - This is what makes the "unsupervised sentiment neuron" so interesting

More asides: open source and open weights

Open source models in the context of AI

- You might have heard about "open source models" in software
- Closed source: eg. Microsoft Windows: you can use it if you buy it, but you cannot see the code or modify it
- Open source: eg. Linux: you can see the code, modify it, and use it for free
    - Companies might still make money by providing support, hosting, or custom versions of open source software

What can you do with open source software?

- There are different licenses for open source software
  - GPL: if you modify the code and distribute it, you must also share your modifications under the same license
  - MIT: you can do whatever you want with the code, as long as you include the original license and copyright notice
  - Apache: similar to MIT, but also includes a patent license
- You can use it and **modify** it, but you cannot claim that you wrote the original code
  - Depending on the license, you might have to share your modifications if you distribute them
  - People do this *all the time*

Open source models in the context of AI

- What does it mean for an AI model to be open source?
- **Closed source**: you can use the model through an API, but you cannot see its architecture or weights, and you cannot download it to run it on your own hardware
  - Examples: OpenAI's GPT-5, Anthropic's Claude, Google's Gemini
- **Open weights**: you can see the model's architecture and weights, you can download it and run it on your own hardware, and you can modify it (with caveats)
  - Examples: Meta/Facebook's LLaMA, OpenAI's GPT-oss, Chinese models like Gwen, Kimi, Deepseek, GLM, etc.

API access vs local models

- API access
  - the model runs on the provider's servers
  - you send your request on the web and receive the response
  - you are using the computer resources of the provider, and pay for it (usually based on usage)
- Local models
  - you download the model and run it on your own hardware
  - you are using your own computer resources

How expensive is it to run a local model?

- Very small models like Bert (110M parameters) can run on a regular laptop (or even a smartphone)
- Models up to 7B parameters can run on a single high-end GPU (like an NVIDIA RTX 4090)
- Larger models like LLaMA 13B or 65B require multiple GPUs or specialized hardware

Open weights is not quite the same as open source

- Open source implies that you can modify the code
- Companies that share the model weights usually:
  - Do not share the training code or the training data
  - The licence might restrict how you can modify or use the model
- People sometimes do minor modifications to the model (like fine-tuning it on a specific task)
- Training the model from scratch is usually not feasible for most people, because it requires a lot of computational resources and data

Hugging Face and the model hub

- Hugging Face is a company that provides tools and infrastructure for working with AI models, especially in the NLP domain
- The Hugging Face Model Hub is a platform where researchers and developers can share their pre-trained models, including their architecture and weights
- You can find models for various tasks, including sentiment analysis, text generation, translation, etc
- They have a Python library called `transformers` that makes it easy to download and use these models in your own code
- Usually, you can use the models for free, but you might have to pay for the computational resources if you run them on a cloud service

Back to sentiment analysis!

Can I run large language models on my laptop?

- There are two types of models:
  - **Encoder-only models**: these are usually smaller and can run on a laptop. They are good for tasks like sentiment analysis, classification, etc.
    - Examples: BERT, DistillBERT, RoBERTa, etc.
  - **Decoder-only models** (LLMs)**: these are usually larger and require more computational resources. They are good for tasks like text generation, question answering, etc.
    - Examples: GPT-3, LLaMA, etc.
- For sentiment analysis, we can use an encoder-only model that is small enough to run on a laptop, and it can still perform well on the task.

Applications of sentiment analysis

- Product reviews: companies can analyze customer feedback to improve their products and services.
- Social media monitoring: organizations can track public opinion about their brand, products, or services.
- Political analysis: sentiment analysis can be used to gauge public opinion on political candidates, policies, or events.
- Market research: businesses can analyze consumer sentiment to inform marketing strategies and product development.

Pitfalls and dangers: Political applications

- Sentiment analysis can be used to manipulate public opinion by identifying and targeting specific groups with tailored messages.
- It can be used to spread misinformation or propaganda by amplifying certain sentiments or suppressing others.
- It can lead to privacy concerns if used to analyze individuals' opinions without their consent.
- It can contribute to polarization by reinforcing existing biases and creating echo chambers.

Try it out: movie review sentiment