

Movie Sentiment Analysis

- ❑ Problem: Given text for a movie detect sentiment?
 - Not given the numeric score
- ❑ Requires deep understanding of text

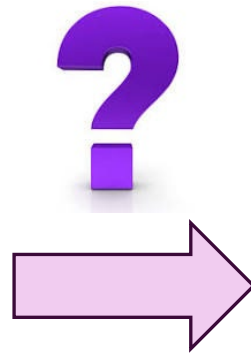
< The Lord of the Rings: The Rings of... ...



★ 2

Bland and mediocre

I would have loved this to be a really good sword and sorcery series. Its barely even average. And when you think how much money Amazon have spent on...



“Negative review”

IMDB Review Data

- ❑ Dataset from 2011 classic paper
- ❑ 50,000 reviews taken from IMBD database
- ❑ Available from Kaggle

Learning Word Vectors for Sentiment Analysis

**Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang,
Andrew Y. Ng, and Christopher Potts**
Stanford University
Stanford, CA 94305

[amaas, rdaly, ptpham, yuze, ang, cgpotts]@stanford.edu

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("lakshmi25npathi/imdb-dataset-of-50k-movie-reviews")
```

Example Reviews

As a another reviewer states Hanna's War is an outstanding film about an outstanding person, Hanna "Anniko" Senesh, who would become the Jewish Joan Of Arc. Unfortunately I diverge in opinion not agreeing that Miss Detmers as the lead is too beautiful to be taken seriously as a resistance fighter. In truth for me her performance is not held back by her beauty but makes it all the more stark in the terror of the sadistic brutality as a resistor she faces. Maruschka Detmers performance is brave, poignant, heartfelt or understood, and totally believable. In other words for me "In the zone." from the opening credits. If you would like to learn about the suffering of someone else for something they believe in and be impressively entertained give Hanna's War with Maruschka Detmers a try. My hat is off also to Ellen Burstyn as Hanna's mother a much well known and famous actress who could have made effort to walk off with the film. In that it is a team effort perhaps of two actress' but not an All About Eve situation.

Example Positive Review

This film is so bad - dialogues, story, actors and actresses - everything! - that it's hard to imagine that we'll see a worse movie this year or in the following years. "Love's Brother" (set in Australia among Italian immigrants) has nothing but shallow clichés about Italian culture to offer, and it is quite telling that even the Italians from and in Italy speak ENGLISH in the film. The message of the film - ugly people have to marry ugly people, beautiful people have to marry beautiful people - is truly discomfoting. Giovanni Ribisi is quite good in films like 'Suburbia' or 'Lost in Translation', but here his pseudo-Italian accent is hard to bear. See this film at your own risk. Trash as trash can!

Example Negative Review

Downloading a Pre-Trained LLM

- ❑ Use MiniLMv2:
 - From Microsoft 2021
 - Very compact, easy to use

- ❑ Download from Hugging Face

**MINILMv2: Multi-Head Self-Attention Relation Distillation
for Compressing Pretrained Transformers**

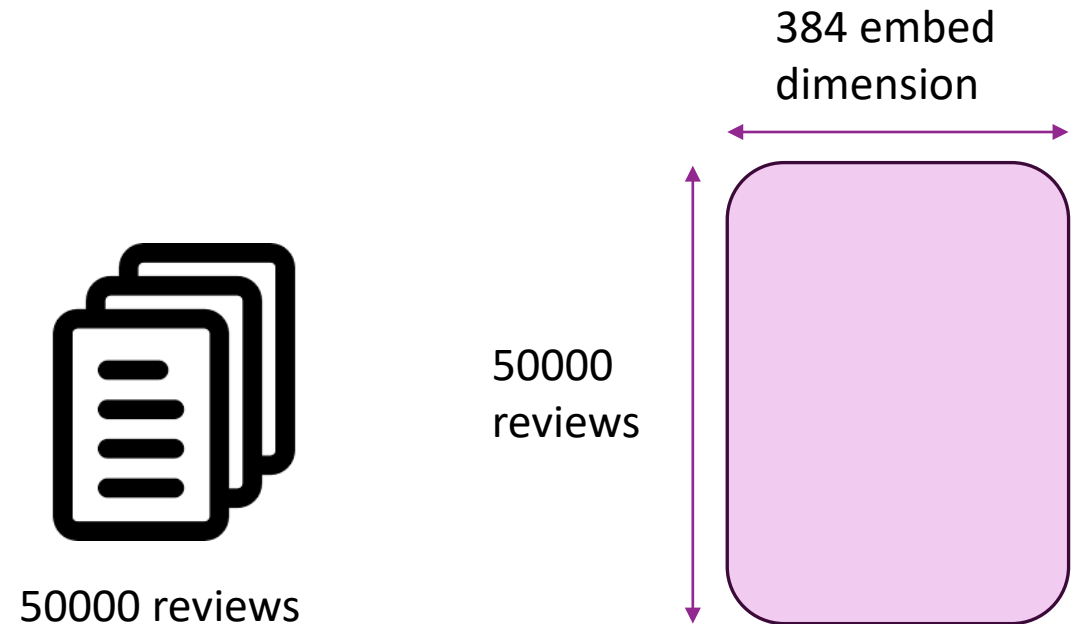
Wenhui Wang Hangbo Bao Shaohan Huang Li Dong Furu Wei*
Microsoft Research
{wenwan,t-habao,shaohanh,lidong1,fuwei}@microsoft.com

```
from sentence_transformers import SentenceTransformer
```

```
model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')  
model = model.to(device)
```

Encoding the Reviews

- ❑ Encode the reviews with the model
- ❑ On Google Colab free T4 GPU
 - About 2 mins for 50,000 reviews
 - With CPU it is too long
- ❑ Each embedding is dimension 384
- ❑ Output a 50000×384 matrix



```
review_embeddings = model.encode(reviews, convert_to_tensor=True, show_progress_bar=True)
print(f"Shape of review embeddings: {review_embeddings.shape}")
print(f"First review embedding (partial):\n{review_embeddings[0][:5]}")
```

Training a Classifier

- ❑ Used a simple neural network
 - Input = 384 (embedding dimension)
 - 50 hidden units
 - Binary output (positive or negative)

```
class SentimentClassifier(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(SentimentClassifier, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(hidden_size, num_classes)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        out = self.fc1(x)
        out = self.relu(out)
        out = self.fc2(out)
        out = self.sigmoid(out)
        return out

# Determine input size from the embeddings
input_size = review_embeddings.shape[1]
hidden_size = 200
num_classes = 1 # For binary classification
```

Classification Results

□ Not the best, but it works

