Want to leverage advanced NLP to calculate sentiment?

Can't be bothered building a model from scratch?

Transformers allows you to easily leverage a pre-trained BERT neural network to do exactly that!

In this video we'll go through how to get up and running with Hugging Face Transformers and BERT to be able to calculate sentiment. We'll run the model using a single prompt but also leverage BeautifulSoup to scrape reviews from Yelp to be able to calculate sentiment on a larger scale.

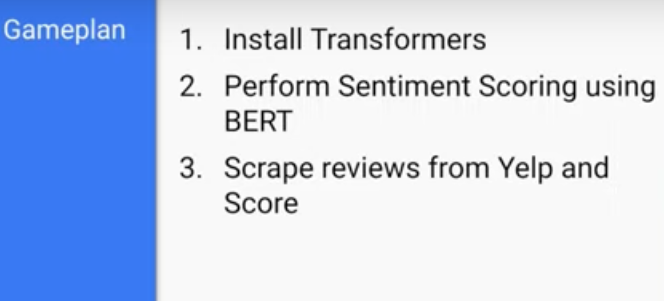
In this video you'll learn how to:

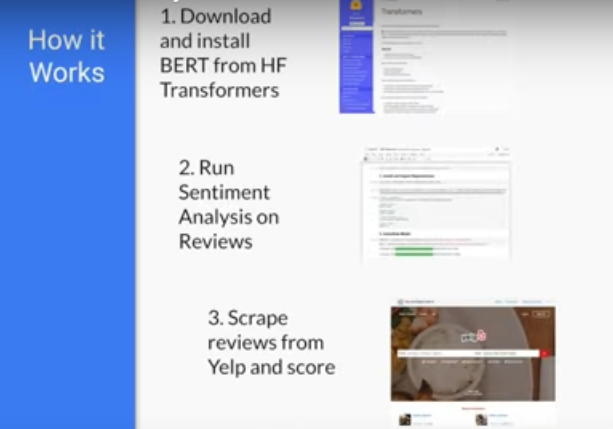
1. Install Transformers

2. Perform Sentiment Scoring using BERT and Python

3. Scrape reviews from Yelp and Calculate their Sentiment

Get the code for this tutorial: <https://github.com/nicknochnack/BERTS>...



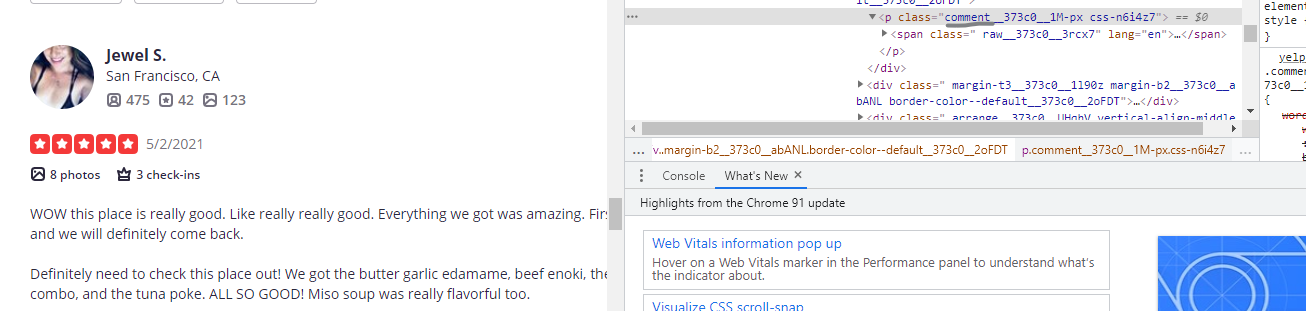


* Install Pytorch for the specific OS
* Install Transformer model from HuggingFace Transformers
  + Transformers to download **nlptown/bert-base-multilingual-uncased-sentiment**
    - This a bert-base-multilingual-uncased model finetuned for sentiment analysis on product reviews in six languages: English, Dutch, German, French, Spanish and Italian. It predicts the sentiment of the review as a number of stars (between 1 and 5).
  + requests : Is for scraping the reviews in Yelp site
  + beautifulsoup4 : Work through the soup that will be returned from the Yelp page and helps to extract the data that we actually need
  + pandas : helps to structure our data into a format to make it easy to work with
  + numpy : For additional Data Transformation processes
* Import
  + AutoTokenizer : Tokenizer converts a string to numbers and we can pass it to NLP model
  + AutoModelForSequenceClassification : Gives the architecture for transformers to load in our NLP model
  + Torch : For PyTorch to use argmax function to extract highest sequence results
  + Request : to scrape data
  + Beautifulsoup: Traverse DOM data and extract the reviews back from yelp page
  + Re : Creates a Regex function in order to extract the specific comments that we want
* Instantiate the model:
  + tokenizer = AutoTokenizer.from\_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
    - Create the tokenizer
  + model = AutoModelForSequenceClassification.from\_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
    - Load the Model
* Encode and Calculate Sentiment
  + tokens = tokenizer.encode('It was good but couldve been better. Great', return\_tensors='pt')
    - pass the sentence and then pass the keyword argument which is converting the tenors to pytorch(pt)
    - Gives the encoded string
  + Pass the token to the model = result
  + Result will give Logits output

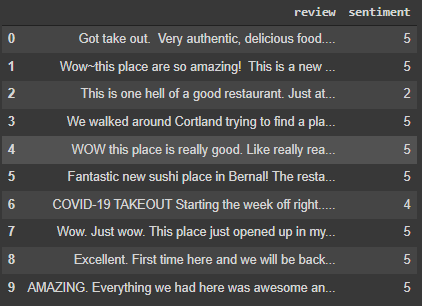
tensor([[-2.7768, -1.2353, 1.4419, 1.9804, 0.4584]],

grad\_fn=<AddmmBackward>)

* + - Number values represents the probability of that particular class being the sentiment in order 🡪 4 -3-2-1-0 🡪 in Pytorch
    - To convert these numbers to useful information we use **torch.argmax(result.logits)** to give the sentiment score 🡪 4
* Collect Reviews
  + Collect reviews from yelp – web scraping
  + All the reviews are called by the class name “**Comments**”



* + Use Regex to extract the classes with comments extension
  + r = requests.get('https://www.yelp.com/biz/social-brew-cafe-pyrmont')
    - Requests to extract all the texts from the web link
  + soup = BeautifulSoup(r.text, 'html.parser')
    - pass the r.text to beautifulsoup and setting to html.parser
  + regex = re.compile('.\*comment.\*')
    - Extract the specific components from the web link
  + results = soup.find\_all('p', {'class':regex})
    - find\_all to find paragraph tag and review contents through regex specified
    - s
  + reviews = [result.text for result in results]



Colab: <https://colab.research.google.com/drive/13ZQXmGL4U5RqIJyy4Qs-d99anJfHFXTc?authuser=1#scrollTo=rwnDN64yW13n>

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

