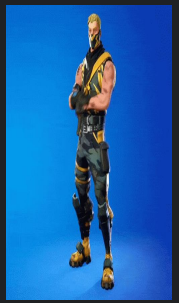
Starting Soon...



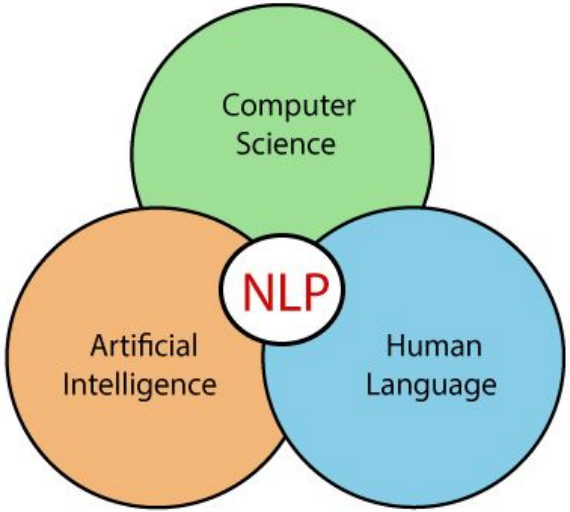
Google Developer Student Clubs

NLP for Beginners:

Build your own summarization model for the Talk Tuah Podcast

Benjamin Tsang & Taha Sarfraz

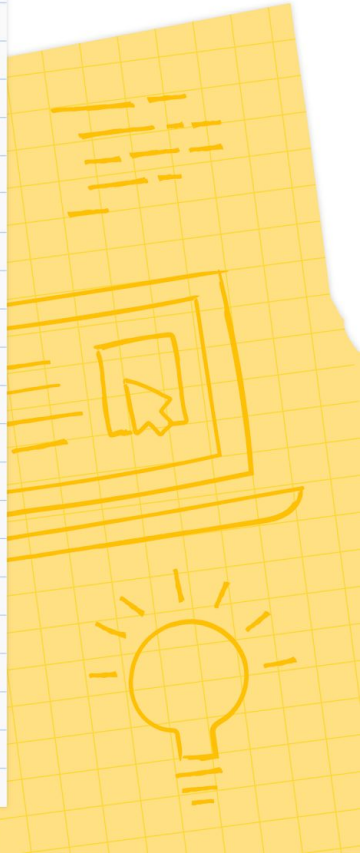
What is NLP?



Google Developer Student Clubs

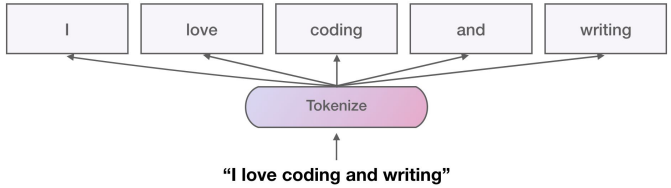
Building a simple extractive summarization model

- Open Google Colab Notebook
- Make a copy
- Upload Talk Tuah transcript
- Follow along, run code for the model
- We will explain concepts at every step!



Tokenization

- Breaking down text into smaller units - **tokens**
- Sentence, word, subword (coding → “code” + “ing”), character
- Breaking down into smaller pieces to analyze



Feature Extraction

- Word Frequency
- Term Frequency - Inverse Document Frequency (tf-idf)
    - Accounts for the fact that some words are common in every document
- Sentence length
- Positional Importance (starting, ending sentences usually are more important)

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

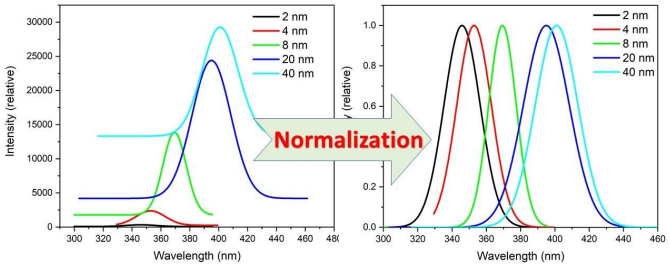
$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

Sentence Selection

- Normalize, so different scales are comparable
- Combine the different scores
- Select the top N sentences based on combined score.
- This gives us our summary!

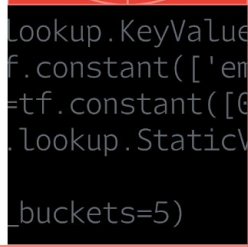



```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

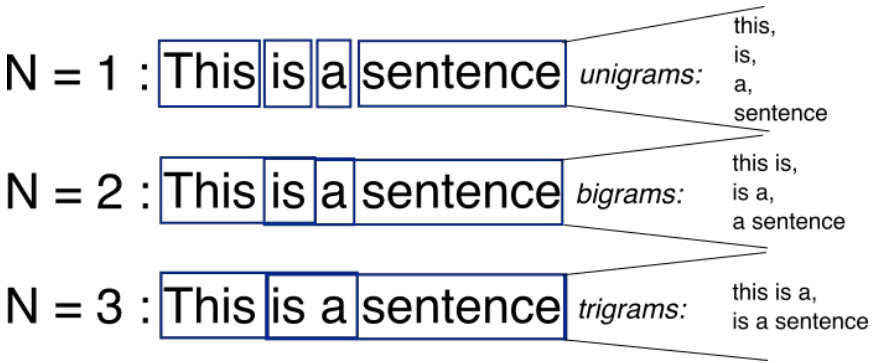
Post-processing

- Remove redundancies
- Grammar and punctuation checks
- Simplify or refine overly complex sentences



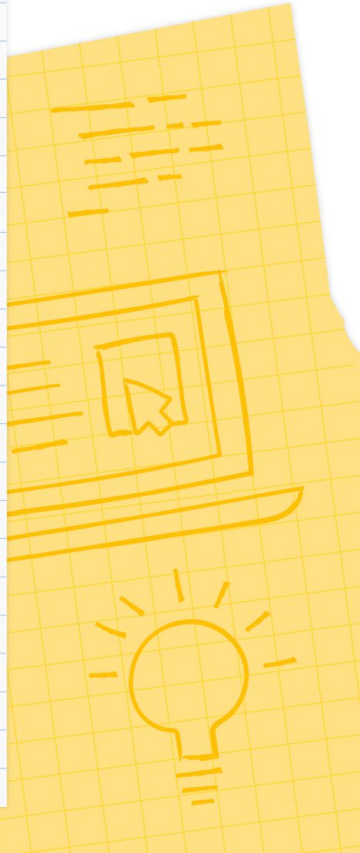
Evaluating the extractive summarization model

- Compression ratio
- Precision
- Recall
- ROUGE-N (overlap of n-grams)
- Intuition



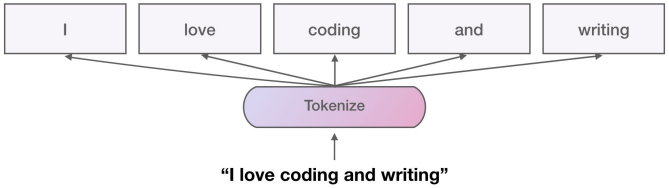
Limitations of extractive summarization models

- Relies on existing sentences, no rephrasing
- Could be less coherent
- Lack of understanding
- Bias towards sentence position and length
- Potentially meaningless sentences



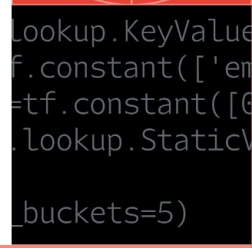
Hugging Face

- What is hugging face
  - Pretrained models that you can use.
  - platform for NLP and machine learning (transformers, datasets, spaces, etc.).
- Why is it important
  - Open-source library powering NLP
  - Common models like BART, T5, or Pegasus.



Use Cases and Applications

- Where Text Summarization is Used:
  - Media: Summarizing news articles or podcasts.
  - Customer Service: Condensing support conversations.
  - Legal: Creating summaries of long legal documents.
  - Education: Summarizing lecture notes or textbooks



Example:

```
from transformers import pipeline

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")
text = " "
summary = summarizer(text, max_length=130, min_length=30, do_sample=False)
print(summary[0]['summary_text'])
```

Example (2):

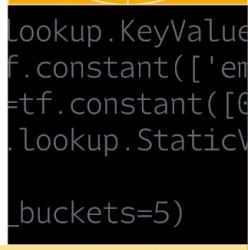
```
summarizer = pipeline("summarization", model="facebook/bart-large-cnn")
```

- Other things you can do w/ the pipeline class
  - Sentiment-analysis
  - Summary
  - poem

Example (2):

```
summary = summarizer(text, max_length=130, min_length=30, do_sample=False)
```

- do_sample
  - do_sample: If False, the model uses greedy decoding (choosing the most probable word at each step).
  - If True, it adds randomness, which can make the output more diverse.



Thank tening!

Google Developer