Natural Language Processing (NLP)

The datasets in the early days of Machine Learning were mainly numeric.

This simplified an already difficult problem because numeric data is easily amenable to manipulations by a computer.

But the quantity of non-numeric data, such as Image and Text, is increasing rapidly thanks to the omnipresence of Internet connected devices.

Natural Languge Processing is the set of techniques that facilitate using *unstructured* text as raw material.

What are some typical tasks

- Is a document *relevant*
- Does the document express a Positive or Negative sentiment
- Summarization of the document

We will provide a brief introduction, based on Neural Network technqiues.

That is not to discount more "classical" methods for NLP

- Part of speech tagging
- Stemming
- Lemmatization

All of these are potentially useful as pre-processing steps for Deep Learning.

However, if our data sets are big enough, it may be counter-productive to preprocess.

- We may introduce our own biases
- "Let the data speak for itself"

Notation

We define a finite set $\mathbf{V}$ to be the universe of possible tokens that make up text

- We denote the items as *tokens* rather than *words*
- Tokens may involve non-words such as puncuation (e.g., exclamation) and special characters (emoji)
- Tokens may consist of sub-words "un-important"

We denote the j^{th} token in vocabulary $\mathbf{V}$ as $\mathbf{V}_j$.

Text $\mathbf{w}$ is a *sequence* of tokens

$$\mathbf{w}_{(1)}, \ldots, \mathbf{w}_{(n_{\mathbf{w}})}$$

using parenthesized subscripts to index into sequences, as usual.

$$\mathbf{w}_{(t)} \in \mathbf{V} \text{ for all } \mathbf{w}_{(t)} \in \mathbf{w}$$

Two non-word tokens are used to denote the start and end of the sequence of tokens in text.

- $\mathbf{w}_{(0)} = \langle \mathrm{START} \rangle$
- $\mathbf{w}_{(n_{\mathbf{w}}+1)} = \langle \mathrm{END} \rangle$

What is special about NLP ?

We begin the study of NLP by discussing issues that are particular to text (NLP_Tokens.ipynb)

Sentiment classification notebook on Colab : simple model

Let's illustrate the traditional summarization of the sequence by some code for the following Classification task

- Input: Movie review, as sequence of characters
- Label: Positive/Negative

NLP notebook: examine the data (https://colab.research.google.com/github/kenperry-public/ML_Spring_2020/blob/master/Keras_examples_imdb_cnn.ipynb#scrollTo=shHO2I

NLP notebook: simple model (https://colab.research.google.com/github/kenperry-public/ML_Spring_2020/blob/master/Keras_examples_imdb_cnn.ipynb#scrollTo=QtvUFJ

Using Convolutions for Text

n-grams via convolution (NLP_Convolution.ipynb)

Sentiment classification notebook on Colab : Convolutional n-grams

Let's augment or simple Sentiment Classification notebook with neural n-grams.

NLP notebook: neural n-grams (https://colab.research.google.com/github/kenperry-public/ML_Spring_2020/blob/master/Keras_examples_imdb_cnn.ipynb#scrollTo=LPChvdı

Embeddings

Embeddings (NLP_Embeddings.ipynb)

Learning embeddings notebook Colab: Sentiment Classification

[NLP notebook: learned embeddings (https://colab.research.google.com/github/kenperry-public/ML_Spring_2020/blob/master/Keras_examples_imdb_cnn.ipynb#scrollTo=f5XrUD](https://colab.research.google.com/github/kenperry-public/ML_Spring_2020/blob/master/Keras_examples_imdb_cnn.ipynb#scrollTo=f5XrUD)

Conclusion

Machine Learning on textual data is becoming an increasingly important source of insight in many domains.

NLP brought together many elements of the knowledge we gained in this course

- One Hot Encoding of tokens
- Reducing variable length sequence inputs to fixed length (RNN)
- Classical ML (Classification)

So, as useful as this lecture was in its own right, we hope it re-inforces everything you've learned throughout this course.

In [ ]:
```
print("Done")
```