Lecture 1: Introduction to NLP

Content

- What is NLP?
- Overview of numerous applications of NLP
- Overview of heuristics, machine learning, and deep learning approaches
- Commonly used NLP pre-processing paradigms

What is NLP?

- NLP, or Natural Language Processing, in Data Science, refers to the field of study and techniques used to enable computers to understand, interpret, and generate human language in a way that is both meaningful and useful. It's a branch of artificial intelligence and linguistics that focuses on the interaction between computers and humans through natural language.
- In data science, NLP techniques are used to analyze, process, and extract insights from large amounts of text data. This can involve tasks such as sentiment analysis, named entity recognition, text classification, language translation, summarization, and more. NLP plays a crucial role in various applications such as chatbots, virtual assistants, information retrieval, social media analysis, and sentiment analysis for customer feedback.

Overview of numerous applications of NLP

- **Email platforms –>** Gmail, Outlook, etc.,
    - use NLP extensively to provide a range of product features, such as spam classification, priority inbox, calendar event extraction, auto-complete, etc.
- **Voice-based assistants, –>** Apple Siri, Google Assistant, Microsoft Cortana, and Amazon Alexa
    - rely on a range of NLP techniques to interact with the user, understand user commands, and respond accordingly.

- **Modern search engines –> Google and Bing**
    - query understanding, query expansion, question answering, information retrieval, and ranking and grouping of the results, to name a few.
- **Machine translation services –> Google Translate, Bing Microsoft Translator, and Amazon Translate**

Overview of heuristics, machine learning, and deep learning

Heuristics-Based NLP

Similar to other early AI systems, early attempts at designing NLP systems were based on building rules for the task at hand. This required that the developers had some expertise in the domain to formulate rules that could be incorporated into a program. Such systems also required resources like dictionaries and thesauruses, typically compiled and digitized over a period of time. An example of designing rules to solve an NLP problem using such resources is lexicon-based sentiment analysis. It uses counts of positive and negative words in the text to deduce the sentiment of the text.

Machine Learning for NLP

Machine learning techniques are applied to textual data just as they're used on other forms of data, such as images, speech, and structured data. Supervised machine learning techniques such as classification and regression methods are heavily used for various NLP tasks. As an example, an NLP classification task would be to classify news articles into a set of news topics like sports or politics. On the other hand, regression techniques, which give a numeric prediction, can be used to estimate the price of a stock based on processing the social media discussion about that stock. Similarly, unsupervised clustering algorithms can be used to club together text documents.

- Naive Bayes
- SVM
- Logistic Regression
- Hidden Markov Model
- Conditional Random Fields(CRFs)

Deep Learning for NLP

We briefly touched on a couple of popular machine learning methods that are used heavily in various NLP tasks. In the last few years, we have seen a huge surge in using neural networks to deal with complex, unstructured data. Language is inherently complex and unstructured. Therefore, we need models with better representation and learning capability to understand and solve language tasks. Here are a few popular deep neural network architectures that have become the status quo in NLP.

- Recurrent Neural Network
- LSTM - Long Short Term Memory
- Convolutional Neural Network
- Transformers
- Autoencoders

Commonly used NLP pre-processing paradigms

- **Tokenization:** splitting large text samples into words / subwords

```
# Tokenization using nltk
from nltk.tokenize import word_tokenize
tokens = word_tokenize(text)
```

- **Regular Expressions:** Used for extracting, finding (or) replacing patterns in text

```
import re
# Extract all alphanumeric characters from text
tokens = re.findall("[\w']+", text)
```

- **Stemming:** Reduces words into their root words, using a rule based system. The root word is not necessarily a meaningful word.

```
from nltk.stem PorterStemmer
stemmer = PorterStemmer()
stemmed_tokens = [stemmer.stem(token) for token in text]
```

- **Lemmatization:** Reduces words into their root words, using a dictionary. Makes sure that root words actually exist in english

```
from nltk.stem WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [lemmatizer.lemmatize(token) for token in text]
```

- **Removing Stopwords:** Removing commonly occuring words like “a”, “an”, “the”, ... which do not contribute much to semantic meaning of text

```
import nltk
nltk.download("stopwords")
stopwords_eng = stopwords.words('english')
filtered_tokens = [token for token in text if not token in stopwords_eng]
```