Berkeley Engineering | BerkeleyHaas

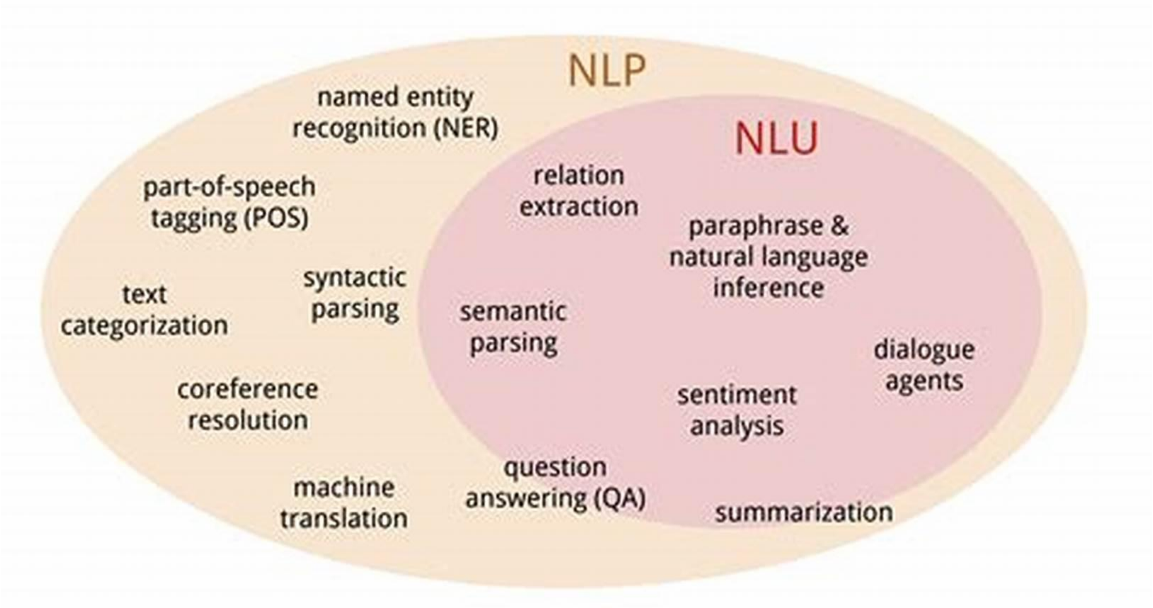
PROFESSIONAL CERTIFICATE IN MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Office Hour #18 with Matilde D’Amelio

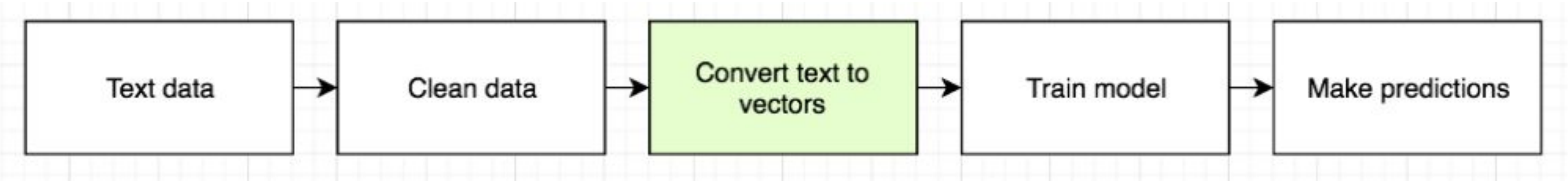
Natural Language Processing

Natural language processing strives to build machines that understand and respond to text or voice data—and respond with text or speech of their own—in much the same way humans do.

Natural language understanding (NLU) focuses on machine reading comprehension through grammar and context, enabling it to determine the intended meaning of a sentence.
Natural language generation (NLG) focuses on text generation, or the construction of text in English or other languages, by a machine and based on a given dataset.



Natural Language Processing



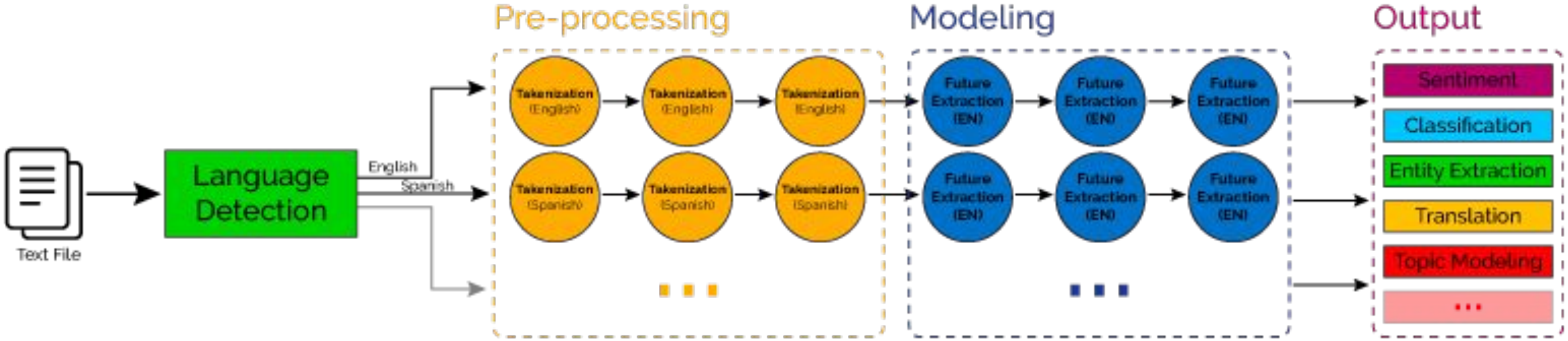
Text data

Clean data

Convert text to vectors

Train model

Make predictions



Text File

Language Detection

English

Spanish

Pre-processing

Takenization (English)

Takenization (English)

Takenization (English)

Takenization (Spanish)

Takenization (Spanish)

Takenization (Spanish)

...

Modeling

Future Extraction (EN)

Future Extraction (EN)

Future Extraction (EN)

Future Extraction (EN)

Future Extraction (EN)

Future Extraction (EN)

...

Output

Sentiment

Classification

Entity Extraction

Translation

Topic Modeling

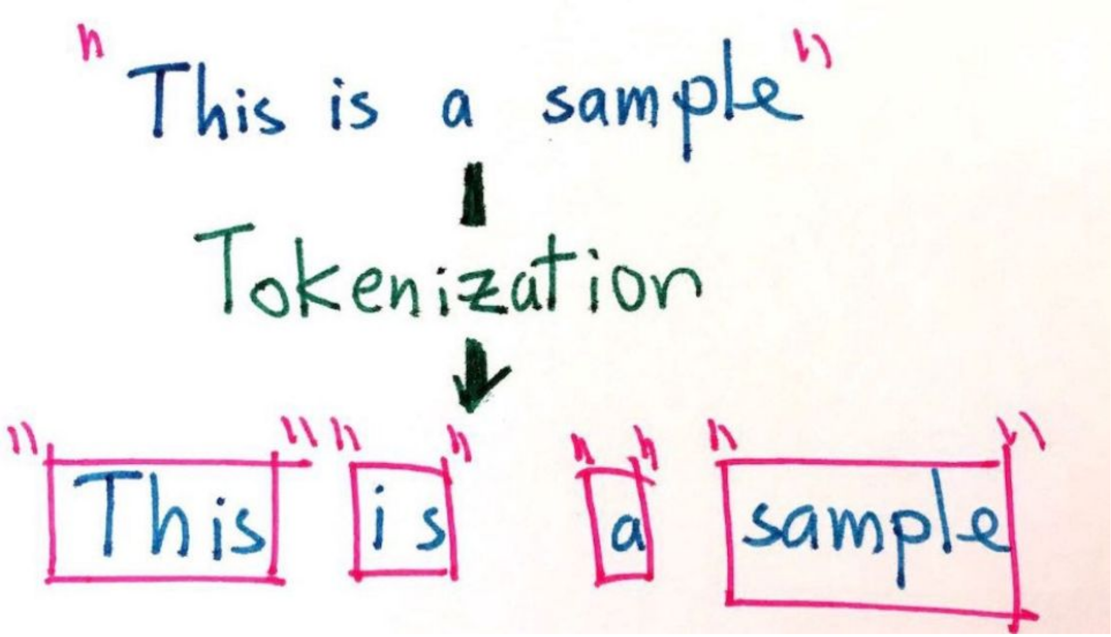
...

Tokenization

`Tokenization` is the process of breaking down the given text in natural language processing into the smallest unit in a sentence called a token. Punctuation marks, words, and numbers can be considered tokens.

Why do we need `Tokenization`? We may want to find the **frequencies** of the words in the entire text by dividing the given text into tokens. Then, models can be made on these frequencies. Or we may want to **tag tokens by word type**.

The NLTK library is used in Python



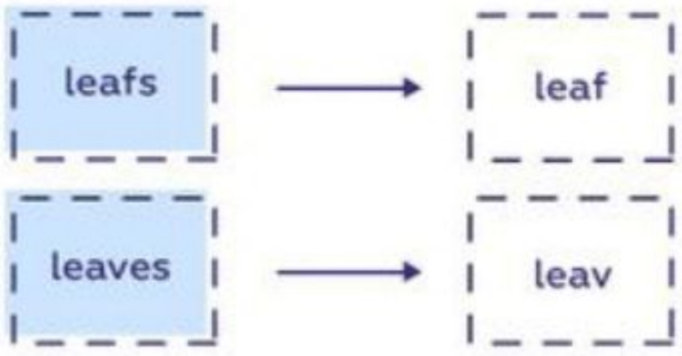
Stemming

Stemming is the process of finding the root of words. *With stemming, words are reduced to their word stems. A word stem need not be the same root as a dictionary-based morphological root, it just is an equal to or* ***smaller form of the word****.*

When you are breaking down words with stemming, you can sometimes see that finding roots is erroneous and absurd. Because Stemming works rule-based, it cuts the suffixes in words according to a certain rule. This reveals inconsistencies regarding stemming. Overstemming and understemming.

Overstemming occurs when words are over-truncated. In such cases, the meaning of the word may be distorted or have no meaning.

Understemming occurs when two words are stemmed from the same root that is not of different stems.

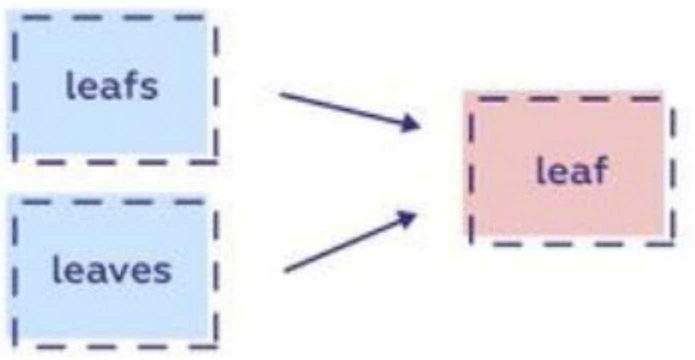


Lemmatization

Lemmatization is the process of finding the form of the related word in the **dictionary**. It is different from Stemming. It involves longer processes to calculate than Stemming.

The aim of lemmatization, like stemming, is to reduce inflectional forms to a ***common base form****. As opposed to stemming, lemmatization does not simply chop off inflections. Instead, it uses lexical knowledge bases to get the correct base forms of words.*

`NLTK` *provides* `WordNetLemmatizer` *class which is a thin wrapper around the* `wordnet` *corpus. This class uses* `morphy()` *function to the* `WordNet CorpusReader` *class to find a* `lemma`.

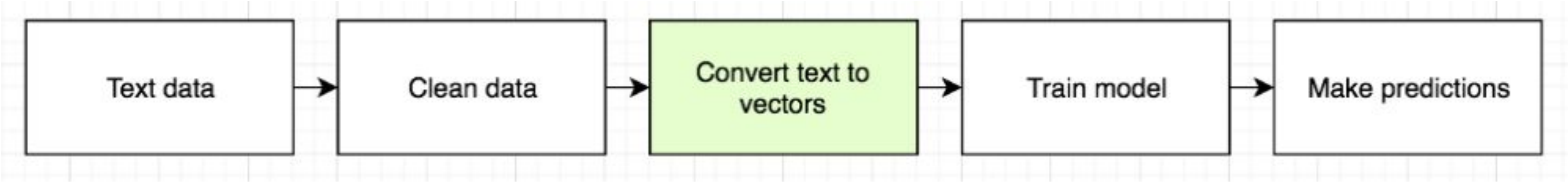


Feature Extraction

Basic feature extraction techniques in NLP is used to analyse the similarities between pieces of text.

Need of feature extraction techniques: Machine Learning algorithms learn from a pre-defined set of features from the training data to produce output for the test data. But the main problem in working with language processing is that machine learning algorithms cannot work on the raw text directly. So, we need some feature extraction techniques to convert text into a **matrix(or vector) of features**. Some of the most popular methods of feature extraction are :

- Bag-of-Words
- TF-IDF

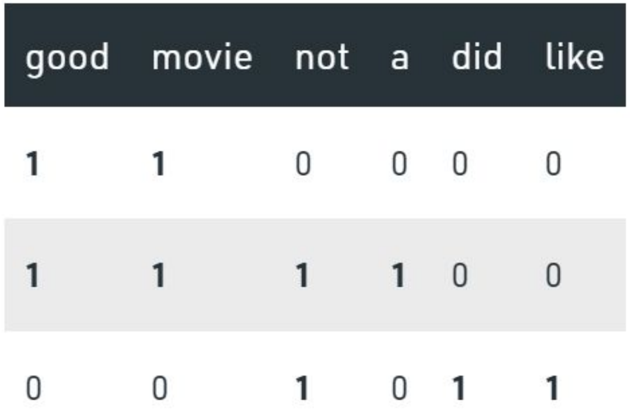


Bag of Words (BoW)

Machine learning models require numerical data as input. We call these numerical representations “vectors”. So if you’re working with text you’ll need to convert the text into a vector before feeding it to a model. We call it a bag of words to emphasize the fact that the order of words is not taken into account

let’s suppose, we have a hotel review text. Let’s consider 3 of these reviews, which are as follows, and the relative matrix of features :

1. *good movie*
2. *not a good movie*
3. *did not like*



good	movie	not	a	did	like
1	1	0	0	0	0
1	1	1	1	0	0
0	0	1	0	1	1

For each review we can now create a vector. Eg. input “good movie”; Output [1 1 0 0 0 0]

TF-IDF

Aim

The TF-IDF technique gives you the relevant term. This relevant term is the one by which the whole context can be understood instead of reading the whole text.

Intuition

- The **word occurring multiple times** implies its importance (TF).
- But at the same time if it appears across multiple docs too frequently then it may not be relevant (IDF). These words we can refer to as **stopwords** such as the, this, etc.

$$\text{TF (m)} = \frac{\text{Number of times term m in doc}}{\text{Total terms in doc}}$$

$$\text{IDF (m)} = \log \frac{\text{Total number of docs}}{\text{Number of docs with term m}}$$

TF-IDF

corpus =”She is wonderful” ; “She is lovely”

Compute Term Frequency (TF): TF_doc2 (“is”) = ⅓ TF_doc2(“lovely”) = 1/3

Compute Inverse Document Frequency (IDF):

IDF(“is”) = log (2/2) = 0 IDF(“lovely”) = log (2/1) = 0.30

As can be seen clearly that the weightage of “is” is less than the “lovely”. Thus lovely seems more relevant.

Dot product of TF and IDF word:

TF-IDF (“is”) = TF . IDF = (1/3) * 0 = 0 TF-IDF (“lovely”) = (1/3) * 0.3 = 0.09

The results showed that the word “is” is irrelevant while “lovely” holds some importance. Reading just the word “lovely” distinct the sentence.

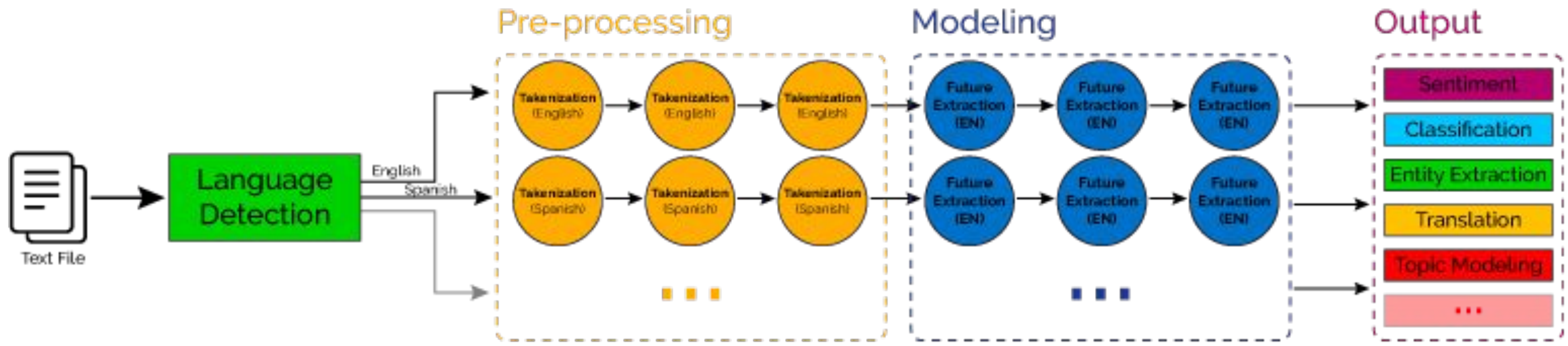
Evaluation Metrics

- Intrinsic Evaluation — Focuses on intermediary objectives (i.e. the performance of an NLP component on a defined subtask)
- Extrinsic Evaluation — Focuses on the performance of the final objective (i.e. the performance of the component on the complete application)

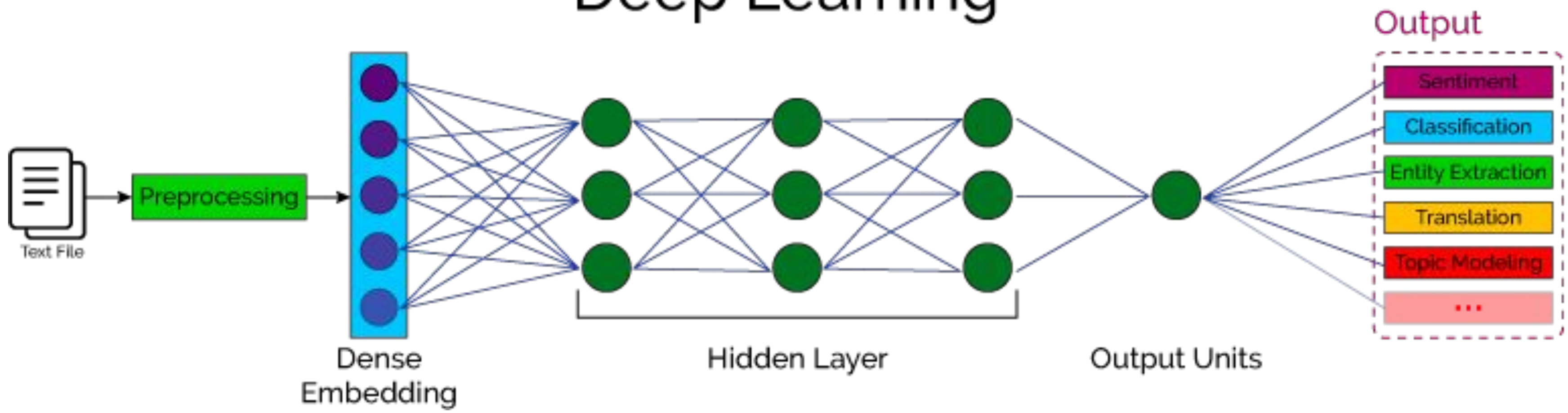
Stakeholders typically care about extrinsic evaluation since they'd want to know how good the model is at solving the business problem at hand. However, it's still important to have intrinsic evaluation metrics in order for the AI team to measure how they are doing.

Deep Learning and NLP

Classical NLP



Deep Learning



Group Discussion

What can be applications of NLP?

NLP Applications

- **Spam detection**: You may not think of spam detection as an NLP solution, but the best spam detection technologies use NLP's text classification capabilities to scan emails for language that often indicates spam or phishing. These indicators can include overuse of financial terms, characteristic bad grammar, threatening language, inappropriate urgency, misspelled company names, and more. Spam detection is one of a handful of NLP problems that experts consider 'mostly solved' (although you may argue that this doesn’t match your email experience).
- **Machine translation**: Google Translate is an example of widely available NLP technology at work. Truly useful machine translation involves more than replacing words in one language with words of another. Effective translation has to capture accurately the meaning and tone of the input language and translate it to text with the same meaning and desired impact in the output language. Machine translation tools are making good progress in terms of accuracy. A great way to test any machine translation tool is to translate text to one language and then back to the original. An oft-cited classic example: Not long ago, translating “*The spirit is willing but the flesh is weak”* from English to Russian and back yielded “*The vodka is good but the meat is rotten*.” Today, the result is “*The spirit desires, but the flesh is weak*,” which isn’t perfect, but inspires much more confidence in the English-to-Russian translation.
- **Virtual agents and chatbots:** Virtual agents such as Apple's Siri and Amazon's Alexa use speech recognition to recognize patterns in voice commands and natural language generation to respond with appropriate action or helpful comments. Chatbots perform the same magic in response to typed text entries. The best of these also learn to recognize contextual clues about human requests and use them to provide even better responses or options over time. The next enhancement for these applications is question answering, the ability to respond to our questions—anticipated or not—with relevant and helpful answers in their own words.
- **Social media sentiment analysis:** NLP has become an essential business tool for uncovering hidden data insights from social media channels. Sentiment analysis can analyze language used in social media posts, responses, reviews, and more to extract attitudes and emotions in response to products, promotions, and events–information companies can use in product designs, advertising campaigns, and more.
- **Text summarization:** Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read full text. The best text summarization applications use semantic reasoning and natural language generation (NLG) to add useful context and conclusions to summaries.

QUESTIONS?