ML Approaches for NLP

Dr. Pruthwik Mishra, Department of AI, SVNIT Surat
Session in STTP on Recent Trends and Practices in AI
Organized by DoAI, SVNIT Surat
10.07.2025

Early Systems

- Rule Based Systems
  - Expert Systems
    - Collection of Conditions (a long list of if conditions)
  - Word Lists/Dictionaries
- Examples
  - Sentiment Analysis
    - List of Positive and Negative Words
  - Machine Translation
    - Bilingual Dictionaries and Grammar Transfer
    - Analyze->Transfer->Generate (Sampark Systems for Indian Language Pairs)
  - Question Answering
    - Keyword/Pattern Based (one such system was ELIZA in 1960s)
    - "What is the capital of France?"
      - a keyword-based system might search for documents containing both "capital" and "France".
      - Retrieve documents that mention "Paris" as the capital city.
- Rule based systems are hard to design
  - Expensive as they require experts
  - Rules are never exhaustive

ML Approaches for NLP

- Natural Language Processing
  - Processing Natural Language (it is not programming language)
    - Flavors of Natural Language
      - Text
      - Audio
  - In this session, we are going to talk about ML approaches in Text Processing
- ML Approaches to be discussed
  - Supervised
    - Classification
    - Sequence Classification
  - Semi-Supervised
    - Co-Learning
      - Combination of Labeled and Unlabeled Data
  - Unsupervised
    - Clustering

How is NLP different?

- ML Approaches need data/features to train
  - The data is in numeric format

- NLP (Natural Language Processing) deals with text
  - Text are not numbers
  - What is the minimum unit for processing?
  - How do we convert them into a numeric format?

Numeric Conversion of Text

- What are the possible options?
  - Can we just give a separate index to a word in a piece of text?
    - Sentence: I am in Surat .
    - Index Sequence: 11 32 107 1377 32 [Assuming we have 10000 unique words in a corpus]
  - One-Hot-Encoding
    - (Assuming we have 10000 unique words in a corpus) Each word will be of 10000 dimensions
    - The index of the word will be 1 others will be 0
  - Term Frequency
    - Count/Frequency of each word in the corpus
    - Can also be normalized
  - TF-IDF
    - Term Frequency and Inverse Document Frequency
    - One of the most widely used representations
    - TF_IDF = TF * IDF
    - Reference: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

More on TF-IDF

Variants of term frequency (tf) weight

weighting scheme	tf weight
binary	$0, 1$
raw count	$f_{t,d}$
term frequency	$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Variants of inverse document frequency (idf) weight

weighting scheme	idf weight ($n_t = \left\vert\{d \in D : t \in d\}\right\vert$)
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log\left(\frac{N}{1+n_t}\right) + 1$
inverse document frequency max	$\log\left(\frac{\max_{\{t' \in d\}} n_{t'}}{1+n_t}\right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

But, Tokenization matters

- Tokenization
  - Splitting/Dividing text into *meaningful?* tokens
  - Based on Tokens
    - Representations will vary
  - What is a token?
    - Words?
    - Punctuations?
    - Emails?
    - URL?
    - Phone Number?
  - The priority is to reduce the sparsity
    - Sparse means no representation at all

Lemmatization/Stemming/Morph Analysis

- Languages are beautiful
- A root can have multiple word forms
  - walk, walks, walked, walking for *walk*
  - go, goes, went, going, gone for *go*
  - child, children for *child*
- Can we represent all these into a single root form?
  - This can be done using Lemmatization (finding lemma) or Morph Analysis
  - Usually linguistic root
- Stemming
  - Reduces a word to its base form (may not be linguistic)
  - Usually done using basic rules of affixation
- Other Preprocessing
  - Case Normalization
- Indian Languages are much more complex
  - Single root can have thousands of word forms in Dravidian languages

Basic Units of Processing

- Word n-grams
  - Text: I am in Surat .
  - Unigrams (n=1): [I, am, in, Surat, .]
  - Bigrams (n=2): [I am, am in, in Surat, Surat .]
  - Trigrams (n=3): [I am in, am in Surat, in Surat .]
  - So on
- Character n-grams
  - Text: I am in Surat .
  - Unigrams: [I,SPACE, a, m, i, n, S, u, r, t, .]
  - Bigrams: [I*SPACE*, *SPACE*a, am, m*SPACE, SPACE*i, in, n*SPACE, SPACE*S, Su, ur, ra, at, t*SPACE, SPACE*.]
  - So on
- Multiple n-grams at Word level and Character level can be combined
- Both can also be combined

Classification Tasks in NLP

- Text Classification
  - Author Identification
  - Sentiment Classification/Detection
  - Aspect Detection
  - Hate Speech Detection
  - Dialect Identification
  - Offensive Language Identification
  - Fake News Detection
- Usually done using Supervised ML Techniques
- In case of scarce data, Semi-Supervised Techniques are also explored

Hate Speech Detection

- Detecting Offensive language in various social media platforms especially Twitter is challenging
- These often target individuals, communities, organizations, and nations
- Several ML Approaches can be used for this task
  - SVM
  - Random Forest
  - Adaboost
  - Ensemble Models
- Preprocessing
  - Customized tokenizer for Twitter
  - Normalize the Twitter handles and hashtags as “USRTOK”, and urls as “URLTOK”.
- Features
  - TF-IDF
    - Word Unigrams
    - Character 2-5 grams
  - Length of Tweet

Sentiment Analysis

- Detect the sentiment polarity of a piece of text
- Can be binary or multi class
  - Binary - *positive, negative*
  - Multi class - *positive, negative, neutral*
- Example:
  - Text: The Indian team succumbed to a 10-wicket loss against Australia in the Border-Gavaskar series
  - Sentiment: *negative*
- Features
  - TF-IDF
    - Word Unigrams + Bigrams
    - Character 2-6 grams
    - Sentiment Lexicon
      - Count the total positive and negative words in the sample and concatenate with the TF-IDF vectors
- Models
  - SVM
  - Logistic Regression
  - Random Forest

Dialect Identification

- Arabic (language), refers to a wide spectrum of native languages used in Middle East and North Africa
- Automatic identification of these dialects becomes an essential task for major natural language applications such as MT, Speech Recognition, Tourist Guide etc.
- It is also modeled as a multi class classification where the classes are represented by the *dialects*
- Features
  - TF-IDF
    - Word n-gram
    - Character n-grams
  - Language Modeling (LM) Score
    - Given a sequence of words w_1, w_2, w_3, .., w_{n-1}, need to predict the next word w_n
    - Usually represented as log of the conditional probability $P(w_n|w_1, w_2, w_3, .., w_{n-1})$
- Models
  - SVM
  - Multinomial Naive Bayes'
  - Logistic Regression

Text Classification using Naive Bayes'

- Given a piece of text, we need to predict the class
- Let us take the example of document classification
- A document *d* can be considered as a collection of words $w_1, w_2, \ldots, w_n$
- The classification can be modeled as depicted below:

$$\text{argmax}_{c \in C} \mathbb{P}(c|d) = \text{argmax}_{c \in C} \frac{\mathbb{P}(d|c)\mathbb{P}(c)}{\mathbb{P}(d)}$$

$$\hat{c} = \text{argmax}_{c \in C} \mathbb{P}(c|d) = \text{argmax}_{c \in C} \mathbb{P}(d|c)\mathbb{P}(c)$$

$$\hat{c} = \text{argmax}_{c \in C} \mathbb{P}(c)\mathbb{P}(w_1|c)\mathbb{P}(w_2|c)\ldots\mathbb{P}(w_{n-1}|c)\mathbb{P}(w_n|c)$$

Sequence Labeling

- Given a sequence, the task here is to predict the labels
- Many NLP Tasks falls in this category
  - Part-Of-Speech (POS) Tagging
    - Sentence: I am going home .
    - POS Sequence: PRP VAUX VM NN PUNC
  - Chunking
    - Sentence: I am going home .
    - Chunks: NP (I_PRP) VP(am_VAUX going_VM) NP(home_NN ._PUNC)
  - Named Entity Recognition (NER)
    - Sentence: SVNIT Surat is situated in the Surat city.
    - Named Entities: ORGANIZATION(SVNIT Surat), LOCATION(Surat)

What is required for POS Tagging?

- Tagset

Language	Tagset	#Tags
English	British National Corpus (BNC) Basic	61
	Penn Treebank POS	36
	Universal POS	17
Indian Languages	BIS Tagset	42 (maximum)
	Universal POS	17

BIS POS Tagset

- Languages Covered
  - Bangla
  - Gujarati
  - Hindi
  - Kashmiri
  - Konkani
  - Maithili
  - Marathi
  - Punjabi
  - Urdu
  - Telugu
  - Kannada
  - Malayalam
  - Tamil
  - Odia (not included, but all types covered by BIS)
- Does not include the tagsets from North-Eastern region
  - However, can easily be extended
- Hierarchical with more than one level of information
  - Top Level
  - Subtypes
- Morphologically richer languages use the superset of tags (finer levels of verb tags) while others use a subset of tags

POS Tags in BIS

N_NN	PR_PRL	V_VM_VF	RP_INTF
N_NNP	DM_DMD	V_VM_VNF	RP_INJ
N_NST	DM_DMI	V_VM_VNG	RD_PUNC
N_NNV	DM_DMR	V_VAUX	RD_SYM
PR_PRP	DM_DMQ	CC_CCD	RD_UNK
PR_PRI	JJ	CC_CCS	RD_ECH
PR_PRF	RB	CC_CCS_UT	QT_QTF
PR_PRC	PSP	RP_RPD	QT_QTC
PR_PRQ	V_VM	RP_NEG	QT_QTO

Approaches

- Rule Based
- Machine Learning Based
  - Naive Bayes
  - Trie Based
  - Hidden Markov Model (HMM)
  - Support Vector Machine (SVM)
  - Conditional Random Fields (CRF)
  - Maximum Entropy Markov Models (MEMM)
- Deep Learning Approaches
  - BiLSTM Based
  - BERT/Transformer Based
  - LLM Based

Conditional Random Fields (CRF)

- Each sample or token is represented as a vector of features
- Each sequence or sentence is a collection of features
- Features used for Morph Analysis and POS Tagging
  - Token Features
    - Token
    - Prefixes upto length 4
    - Suffixes upto length 7
    - Binary Length Feature (MORE if len(token) > 4 else LESS)
  - Context Features
    - Window length 5 (w_{i-2},w_{i-1}, w_i, w_{i+1}, w_{i+2})

Example

- Mary Jane can see Will.
- Spot will see Mary.
- Will Jane spot Mary?
- Mary will pat Spot.

https://www.mygreatlearning.com/blog/pos-tagging/

Implementation of CRF

- Implemented using CRF++ toolkit (https://taku910.github.io/crfpp/)

- Code shared for training the CRF models for Hindi
  - Available at https://github.com/Pruthwik/Hindi-POS-Tagger-and-Chunker-Training-Using-CRF

Unsupervised Learning

- Clustering
  - Hate Speech Detection