CS323:Introduction to NLP

Getting Started with NLP

Ashish Anand

Professor, Dept. of CSE, IIT Guwahati

Associated Faculty, Mehta Family School of Data Sc and AI, IIT Guwahati

Outline

- What is NLP?
  - Definition
  - Ambiguities and Different levels of NLP
- Getting Started
  - Different types of corpora
  - Text Normalization
    - Basic pre-processing
    - Word and Sentence Segmentation
      - Rule and Heuristics – Language specifi
      - Subword Tokenization

Learning Objective

- Understand the different levels of NLP and how they contribute to ambiguities and complexities
- Introduction to different types of corpora
- Essential pre-processing and normalization tasks while working with raw text

Defining NLP

What do we mean by NLP?

- Natural Language – Written or Spoken language used by humans. Example: Assamese, Bengali, Hindi, Sanskrit, English, German, …

- NLP – Computational methods to learn, understand & generate natural language content

- Multiple distinct fields study human language: Linguists, Speech Recognition, Computational Linguists etc.

Three Themes of NLP

- Learning and Knowledge
- Search and Learning
- Relational, Compositional and Distributional Perspectives

Learning and Knowledge

Debate on learning from scratch vs linguistic knowledge

- Whom to prioritize “Learning from scratch” or “understanding the linguistic structure and inferring from logic-based representation
- Age-old debate
- Giving rise to two paradigms: Rationalist and Empiricism

Rationalist Paradigm

- Transform text into linguistic structures
  - Subword units called ***morphemes***, word-level ***parts-of-speech***, tree-structured ***grammar representations***, **logic-based** representations of meaning
  - Use them appropriately for the desired applications

- Primary Objective
  - describe the language models of human mind (I-Language)

- Argument
  - Existence of innate language faculty [Noam Chomsky]
  - Language learning capabilities of children: faster and with fewer examples

Rationalist: In Practice

- Focuses on
  - Rule based system and defining grammar
- Initial AI systems mimicked innate language faculty by trying to hardcode a lot of starting knowledge and reasoning mechanism
- Models: State Machines, Formal rule systems (Regular Grammar/CFG), Logic

Empiricist: Sense and experience in tandem with generic cognitive ability

- Primary objective: describe the language as it actually occurs (E-Language)

- Differers with rationalist in degree of belief about nature of precoded knowledge
  - Does assume generic ability of association, pattern recognition and generalization
  - Generic ability works in tandem with rich sensory inputs

Empiricist: In Practice

- Focuses on
  - Large collection of text and data-driven approaches
- Explores and uses common patterns in language use
- Appropriate Probabilistic, Statistical, Pattern-recognition and ML Models
  - Objective is to tune model parameters to learn the complicated and extensive language structure
  - We will see plenty of them during the course

Synthesis of the two paradigms

- Exploit linguistic structure as features in learning models
- Building model architectures inspired by linguistic theories

Two Relevant Discussions: Optional Reading

- Church, K. 2011. A pendulum swung too far, *Linguistic Issues in Language Technology* 6(5): 1-27

- Manning, C. D. 2015. Last words: Computational linguistics and deep learning. *Computational Linguistics* 41(4): 701-707

Search and Learning

Generic Formulation

Many NLP problems can be mathematically formulated as

$$\hat{\mathbf{y}} = argmax_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \Psi(\mathbf{x}, \mathbf{y}; \theta)$$

where,

- $\mathbf{x}$: input
- $\mathbf{y}$: output
- $\boldsymbol{\Psi}$: scoring function (**model**) mapping elements of the set $\mathcal{X} \times \mathcal{Y}$ to real numbers
- θ: set of model parameters
- $\hat{\mathbf{y}}$: predicted output

Examples of $\mathbf{x}$: social media post, sentence in one language
Examples of $\mathbf{y}$: sentiment, sentence in another language, named entities

Search

- Computes the *argmax* of the function Ψ
- Often machinery of **Combinatorial optimization** as often outputs are discrete variables
- Simple search algorithms to dynamic programming and beam search

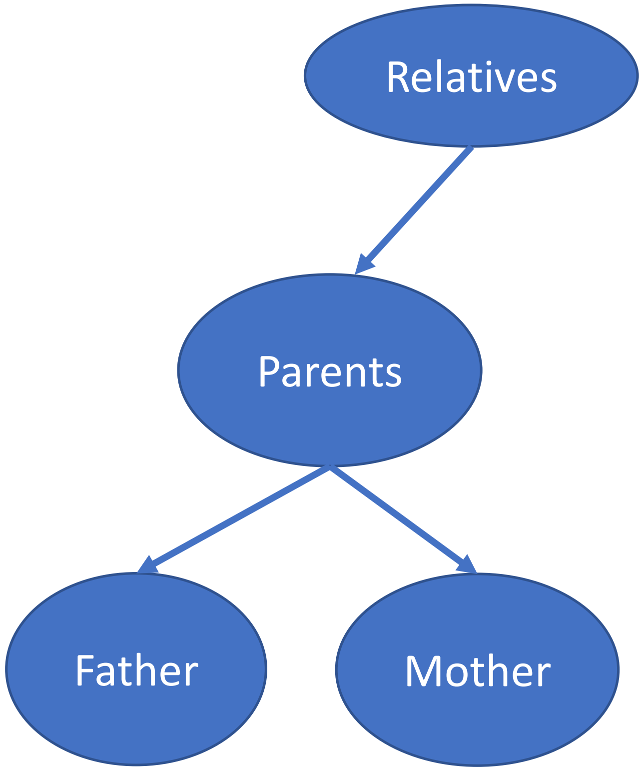
Learning

- Finding the model parameters $\boldsymbol{\theta}$
- Mostly, again an **optimization** problem.
- Relying on **numerical optimization**, as parameters are often continuous

Three complimentary perspectives of meaning

Relational, Compositional and Distributional

Relational Perspectives



Relational Perspectives

- Basis for **semantic ontologies** such as **WordNet**

- However, not easy to formalize the problem mathematically or computationally,

- Building manually is also challenging

Compositional Perspective

- The meaning of word is constructed from the constituent parts
- Can be applied to larger units: phrases, sentences, and beyond

Distributional Perspectives

- However, some words, *idiomatic phrases* have meaning different from the sum of words

- Distributional perspectives allow to learn about meaning from unlabelled data

- This perspective is being exploited in vector semantics

Why NLP is Hard?

"WHAT IS YOUR LITTLE BROTHER CRYING ABOUT?"
"OH, 'IM—'E'S A REG'LAR COMP'TATIONAL LINGUIST, 'E IS."

http://specgram.com/CLIII.4/08.phlogiston.cartoon.zhe.html

Language is ambiguous

Example:

I made her duck

Time flies like an arrow.

- What is your inference of the two sentences?
- Whether all of them are meaningful/grammatically correct ?

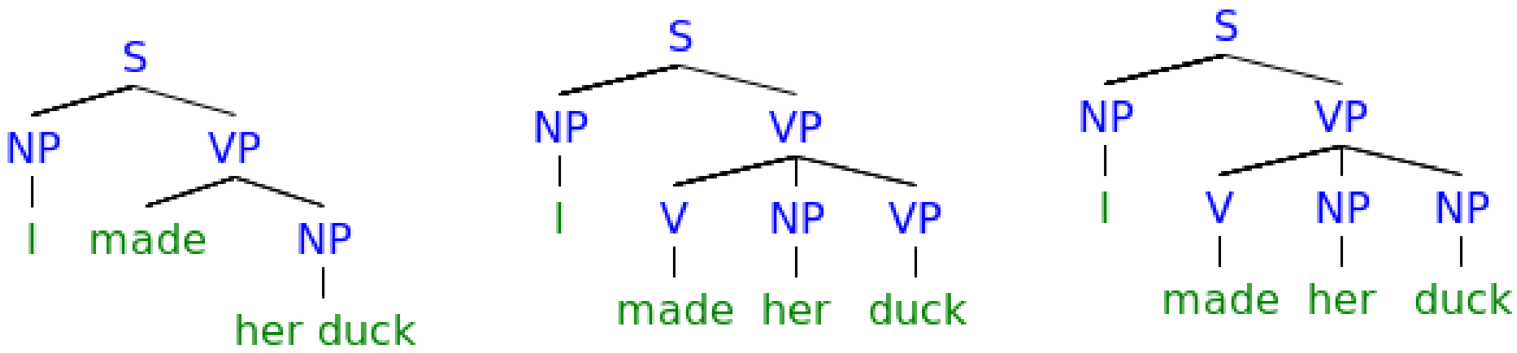
Language is ambiguous

Example: *I made her duck*

- Interpretations :
  - *I cooked duck for her*
  - *I cooked duck belonging to her*
  - *I caused her to quickly lower her body*

Ambiguity

The variation in interpretation is due to



More examples of ambiguity

- Anne Hathaway vs. Warren Buffett's Berkshire Hathaway stock
  - When *Bride Wars* opened the stock rose 2.61%.

    [source: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1162/handouts/cs224n-lecture1.pdf]

- *Every Indian has a mother* vs. *Every Indian has a prime minister*

- *We gave the monkeys the bananas because they were hungry* vs. *We gave the monkeys the bananas because they were over-ripe*

Types of Ambiguity

- Phonetic
  - My finger got number
- Morphological
  - Impossible vs important
  - Ram is quite impossible/ Ram is quite important
- Part of speech
  - Geeta won the first round
- Syntactic
  - Call Ram a taxi

Types of Ambiguity

- Pp attachment
  - The children ate the cake with a spoon.
- Cc attachment
  - Ram likes ripe apples and pears
- Sense
  - Ram took the bar exam
- Referential
  - Ram yelled at Shyam. He was angry at him
- Metonymy
  - Sydney called and left a message for Ram

Some other sources of difficulties

- Non-standard, slang, novel and short words
  - A360, +1-646-555-2223
  - Selfie, chillax
- Inconsistencies
  - junior college, college junior
- Parsing problems
  - Cup holder
- Metaphors, Humors, Sarcasm

Summary: why NLP is hard?

- Highly ambiguous at all levels
- Context is important to convey meaning
- Involves reasoning about the world

Different Levels of NLP

- Word
  - Phonetics and Phonology: study of linguistic sounds
  - Morphology: study of meaningful components of words [example]
- Syntax: structural relationship between words
- Semantic: study of meaning
  - Lexical semantics: study of meanings of words
  - Compositional semantics: How to combine words
- Pragmatics and Discourse: dealing with more than a sentence: paragraph, documents

Getting Started with NLP

Source: Corpus

- Corpus (plural : *corpora*)
  - Special collection of texts collected according to a predefined set of criteria
  - May be available as pre-pr0cessed and linguistically-marked-up or in raw format
- Different types of corpora
  - Monolingual
  - Parallel: bilingual or multilingual [Vary at the alignment level]
  - Comparable: bilingual or multilingual
  - Learner Corpus
  - Diachronic Corpus

Examples of Corpus

Corpus	Tokens	Types
Switchboard phone conversations	2.4 million	20000
Shakespeare	884,000	31000
Brown	1 million	38000
Google N-grams	1 trillion	13 million

Two ways to talk about words:

1. Tokens: each occurrence of all words is counted
2. Types: number of distinct words

More Examples of Corpora

- Access to multiple corpus from tools like *NLTK*
- Building from databases such as PubMed, free text from web, Wikipedia, Social media platforms etc.
- Task specific
- Shared task challenges: ACE, CoNLL, SemEval, BioAsq, SQuAD, CORD-19

- **Caution:** One shoe does not fit all.
- **Caution:** Ethical and Bias Issues

Text Preprocessing

- Removing non-text (e.g. tags, ads)
- Text Normalization
  - Segmentation: Word and Sentence Segmentation
  - Normalizing Word Formats
    - Spelling Variations: Labeled/labelled
    - Capitalization: Led/LED
    - Lemmatization
    - Stemming
    - Morphological analysis: dealing with smallest meaning-bearing units

Text normalization

Tokenization: Word Segmentation

Definition

- Process to divide the input text into units, also called, *tokens*, where each is either a *word* or a *number* or a *punctuation mark*.

What counts as a word?

I am interested in Natural Language Processing, but I’m not sure of the required prerequisites.

What counts as a word?

- Should I count punctuation as a word?
- Should I treat I’m as one word or break them into three words: I, ’, m? [**Clitic**]
- Should I consider “Natural Language Processing” as one word or 3 words?

What counts as a word?

- Kucera and Francis (1967) defined "*graphic word*" as follows :
  - "a string of contiguous alphanumeric characters with space on either side; may include hyphens and apostrophes, but no other punctuation marks"

Challenges in defining word as a contiguous alphanumeric characters

- Too restrictive
  - Should we consider “$12.20” or “Micro$oft” or “:)” as a word?

- We can expect several variants especially in forums like Twitter etc. which may not obey exact definition but should be considered as a word.

- Simple Heuristic: *Whitespace*
  - *“a space* or *tab* or the *new line*” between words.
  - Still to deal with several issues.

Some challenges with simple heuristics

- Periods
  - Wash. vs wash
  - Abbreviations at the end vs. in the middle – e.g. etc.
  - More on this while discussing sentence segmentation

- Single apostrophes
  - Contractions such as I’ll, I’m etc.: should be taken as two words or one word?
  - *Penn Treebank* split such contractions.
  - Phrases such as *dog’s vs. yesterday’s* in “The house I rented yesterday’s garden is really big”.
  - Orthographic-word-final single quotation (often comes at the end of sentence/quoted fragment) and cases like (plural possessive) “boys’ toys”.

Defining words: Problems

- Hyphenation
  - Again the same question – “do sequences of letters with a hyphen in between count as one word or two?
  - Occurrences like `e-mail`, `co-operate` vs. `non-lawyer`, `so-called`, `text-based`
  - Inconsistency in using words like “cooperate” as well as “co-operate”
  - Line-breaking hyphen vs. actual hyphen happens at the end of line [***haplology***]
  - Hyphens to indicate correct grouping of words: take-it-or-leave it in “`a final take-it-or-leave it offer`”

- Word with a whitespace between its parts
  - New Delhi, San Francisco
  - … the New Delhi-New Jalpaiguri special train …

Defining words: Problems: Spoken Corpora

- This lecture umm is main- mainly divided into two components

- Two types of **disfluencies**
  - **Fragments: main-**
  - **Fillers/Filled pauses: uh.. Umm..**

Some other issues

- Quite a large vocabulary
  - Restricting a vocabulary size enhances OOV problem
- No implicit notion of similar words
  - Each word is given distinct id

Tokenization in Practice

- Deterministic algorithms based on regular expressions
- Compiled into efficient finite state automata

Word segmentation in other languages

- 请将这句话翻译成中文 [Please translate this sentence into Chinese]
    - Languages like Chinese, Japanese have no spaces between words
    - Japanese is further complicated with multiple alphabets intermingled

- Compound nouns written as a single word
    - Lebensversicherungsgesellschaftsangestellter [Life insurance company employee]

Word Tokenization in Chinese

- Chinese words are composed of characters
  - Characters are generally 1 syllable and 1 morpheme.
  - Average word is 2.4 characters long.
- Standard baseline segmentation algorithm:
  - Maximum Matching (also called Greedy)

Source: SLP-Slides-Chap2

Maximum Matching
Word Segmentation Algorithm

- Given a wordlist of Chinese, and a string.

1) Start a pointer at the beginning of the string
2) Find the longest word in dictionary that matches the string starting at pointer
3) Move the pointer over the word in string
4) Go to 2

Source: SLP-Slides-Chap2

Max-match segmentation illustration

- Thecatinthehat
- Thetabledownthere

the cat in the hat

the table down there

theta bled own there

- Doesn’t generally work in English!
- But works astonishingly well in Chinese
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- Modern probabilistic segmentation algorithms even better

Source: SLP-Slides-Chap2

Subword Tokenization: Motivation

- Frequent words should be identified as a token
- Rare words should be broken into meaningful subword tokens:
  - Unknowingly : “un”, “know”, “ing”, “ly”
  - Helps in taking care of OOV, rare and related words
- Reasonable vocabulary size
- To make it language independent

Subword Tokenization: Popular Methods

- Byte Pair Encoding (BPE)[1]
- Wordpiece[2]
  - Similar to BPE, except the merging criteria is different
- Unigram[3] and Sentencepiece[4]
  - Rely on unigram language model
  - Language independent

1. Sennrich et al. 2015. Neural machine translation of rare words with subword units. *ACL 2016*
2. Schuster and Nakajima. 2012. Japanese and Korean voice search. *ICASSP 2012*
3. Kudo. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. *ACL2018*
4. Kudo et al. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. *EMNLP 2018 (demo paper)*

Byte Pair Encoding

- Used for **data compression** in **Information theory**
- **Idea:** Iteratively merge most frequently byte pairs into a byte not present in the data.

BPE for Word Tokenization

- **Assumption:** corpus has been already tokenized
- Step 1: Count the frequency of each word appearing in the given corpus.
- Step 2: For each word, append them with a special token ``<E>", signifying end of a word.
- Step 3: Break each word into their constituent characters. So a word "exam" will be converted into a sequence of characters ["e","x","a","m","<E>"].

BPE for Word Tokenization

- Step 4: In each iteration, count the frequency of each consecutive byte pair and merge the most frequent byte pairs into one.

- Step 5: Stop after a fixed number of iterations (i.e. merge operations) or after obtaining a maximum number of tokens.

BPE Tokenization: Illustration

- Dictionary
  - {'low<E>': 5, 'lower<E>': 2, 'newest<E>': 6, 'widest<E>': 3}
- Vocabulary on characters
  - {'d','e','i','l','n','o','r','s','t','w','<E>’}
- 1st Iter: {'d','e','i','l','n','o','r','s','t','w','<E>','es’} [e and s occurred together 9 times]
- 2nd Iter: {'d','e','i','l','n','o','r','s','t','w','<E>','es’, ‘est’}
- And So on.

BPE Tokenization: Encoding: Text Data Tokenization

- **Question:** How to tokenize a given sequence of words into learned tokens?

- **Answer**
  - Idea: Run the merged byte pairs in the order they were learned.
  - Segment each test word into characters
  - Apply first merge rule [Our example, merge ‘e’ and ‘s’]
  - Then second and so on…
  - Example: newer -> “new” “er_”

Text normalization

Sentence Segmentation

Defining Sentence Boundary

- Something ending with a ‘.’, ‘?’, or ‘!’
  - Language specific
- Problem with ‘.’
  - Still 90% of periods are sentence boundary indicators [Riley 1989].
- Sub-sentence structure with the use of other punctuation
  - “The scene is written with a combination of unbridled passion and sure-handed control: In the exchanges …….. inexorability of separation”

Defining Sentence Boundary: A heuristic

- Put putative sentence boundaries after occurrences of ., ?, ! (and may be ;, :, -)
- Move the boundary after following quotation marks, if any.
- Disqualify a period boundary if –
  - It is preceded by a known abbreviation that does not generally occur at the end of sentence such as Dr., Mr. or vs., but is commonly followed by a capitalized proper name
  - It is preceded by a know abbrev. and not followed by an uppercase word. This will deal with cases like etc. or Jr.
- Disqualify a boundary with a ? or ! If
  - It is followed by a lowercase letter (or name)

Issues with Heuristic or set of pre-defined rules

- Is it possible to define such rules without the help of experts?
- Will it work for all languages?

Machine Learning Methods: Sentence boundary as classification problem

- Riley (1989) used classification trees
  - Features: case & length of the words preceding and following a period; prior prob of words occurring before and after a sentence boundary etc.
- Palmer and Hearst (1997) used neural network model
  - Instead of prior probability, PoS distribution of the preceding and following words.
  - Language-independent model with accuracy of 98-99%
- Reynar and Ratnaparkhi (1997) and Mikheev (1998) used Max. Ent approach
  - Language independent model with accuracy of 99.25%

Tools to getting started with NLP

spaCy

flair

TextBlob

Stanford University

S

NLP

Natural Language Processing

jupyter

NLTK

Natural Language Toolkit

Reference Guide

OPEN NLP™

AllenNLP

NLP ARCHITECT

NLP.js

Source: https://medium.com/microsoftazure/7-amazing-open-source-nlp-tools-to-try-with-notebooks-in-2019-c9eec058d9f1

References

- Jurafsky and Martin, Speech and Language Processing, 3rd Ed. Draft [Available at https://web.stanford.edu/~jurafsky/slp3/]
- Eisenstein, Introduction to NLP, MIT Press

Thanks!
Question and Comments!

anand.ashish@iitg.ac.in https://www.iitg.ac.in/anand.ashish