Lucerne University of Applied Sciences and Arts

NLP Pipeline

NLP
Andreas Marfurt

Information Technology
21.02.2025

FH Zentralschweiz

Overview

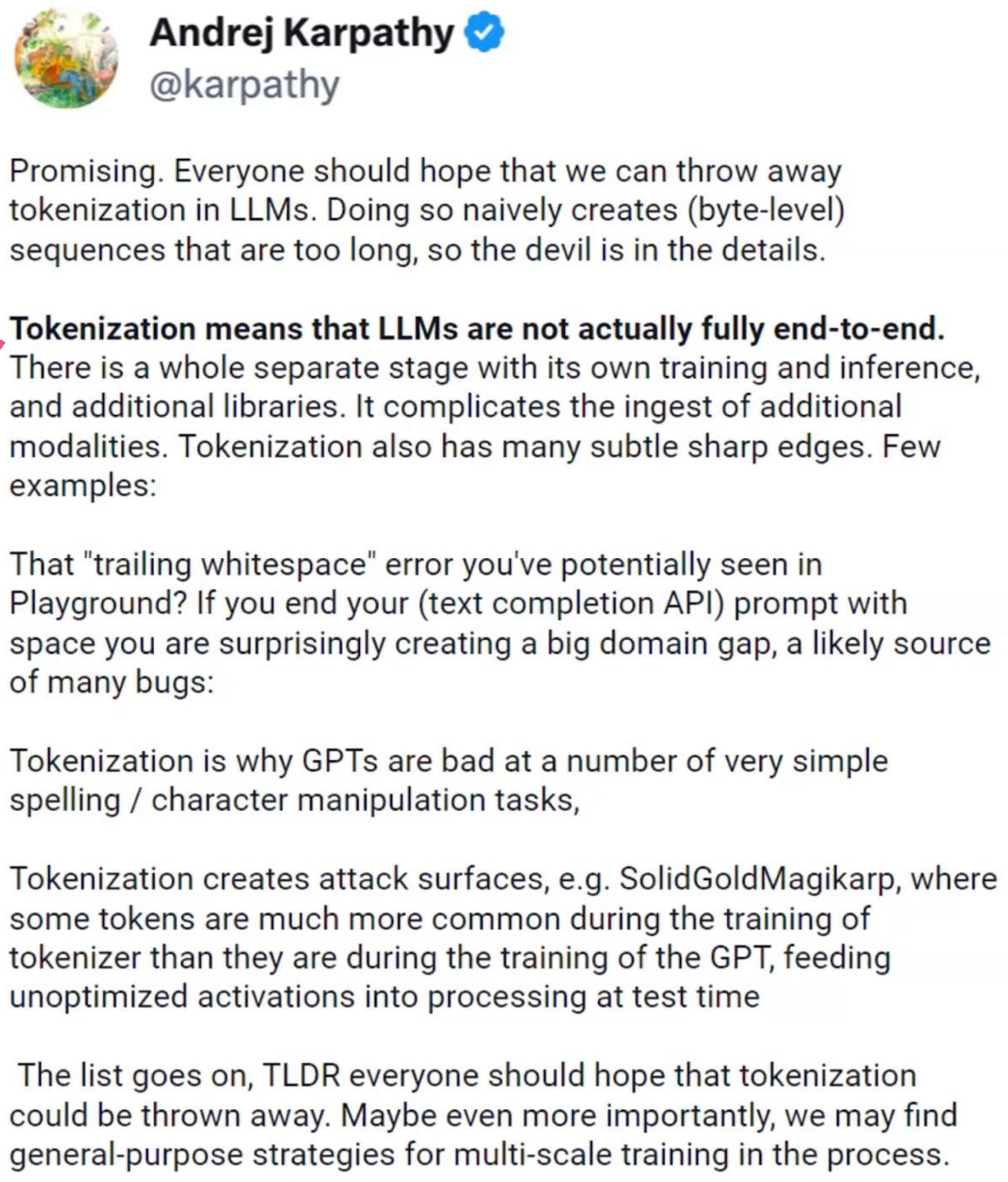
- Tokenization
- Stemming
- Lemmatization
- Part of speech
- Parsing
- (Named) entity resolution/linking

Preprocessing Text

- Want to process text with a model
- Text (e.g. news article) is one long sequence of ...
  - characters?
  - words?
  - bytes?
- What steps are necessary to bring text into a common form?
  - Correct spelling mistakes?
  - Remove formatting/HTML?
  - Convert scripts/encoding?
  - Split into meaningful chunks?
  - Remove words that appear too frequently/infrequently?
- Preprocessing has a big impact on model performance, but is often treated as irrelevant

Preprocessing Text

- Want to process text with a model
- Text (e.g. news article) is one long sequenc
  - characters?
  - words?
  - bytes?
- What steps are necessary to bring text into
  - Correct spelling mistakes?
  - Remove formatting/HTML?
  - Convert scripts/encoding?
  - Split into meaningful chunks?
  - Remove words that appear too frequently/inf
- Preprocessing has a big impact on model pe irrelevant



Andrej Karpathy
@karpathy

Promising. Everyone should hope that we can throw away tokenization in LLMs. Doing so naively creates (byte-level) sequences that are too long, so the devil is in the details.

Tokenization means that LLMs are not actually fully end-to-end. There is a whole separate stage with its own training and inference, and additional libraries. It complicates the ingest of additional modalities. Tokenization also has many subtle sharp edges. Few examples:

That "trailing whitespace" error you've potentially seen in Playground? If you end your (text completion API) prompt with space you are surprisingly creating a big domain gap, a likely source of many bugs:

Tokenization is why GPTs are bad at a number of very simple spelling / character manipulation tasks,

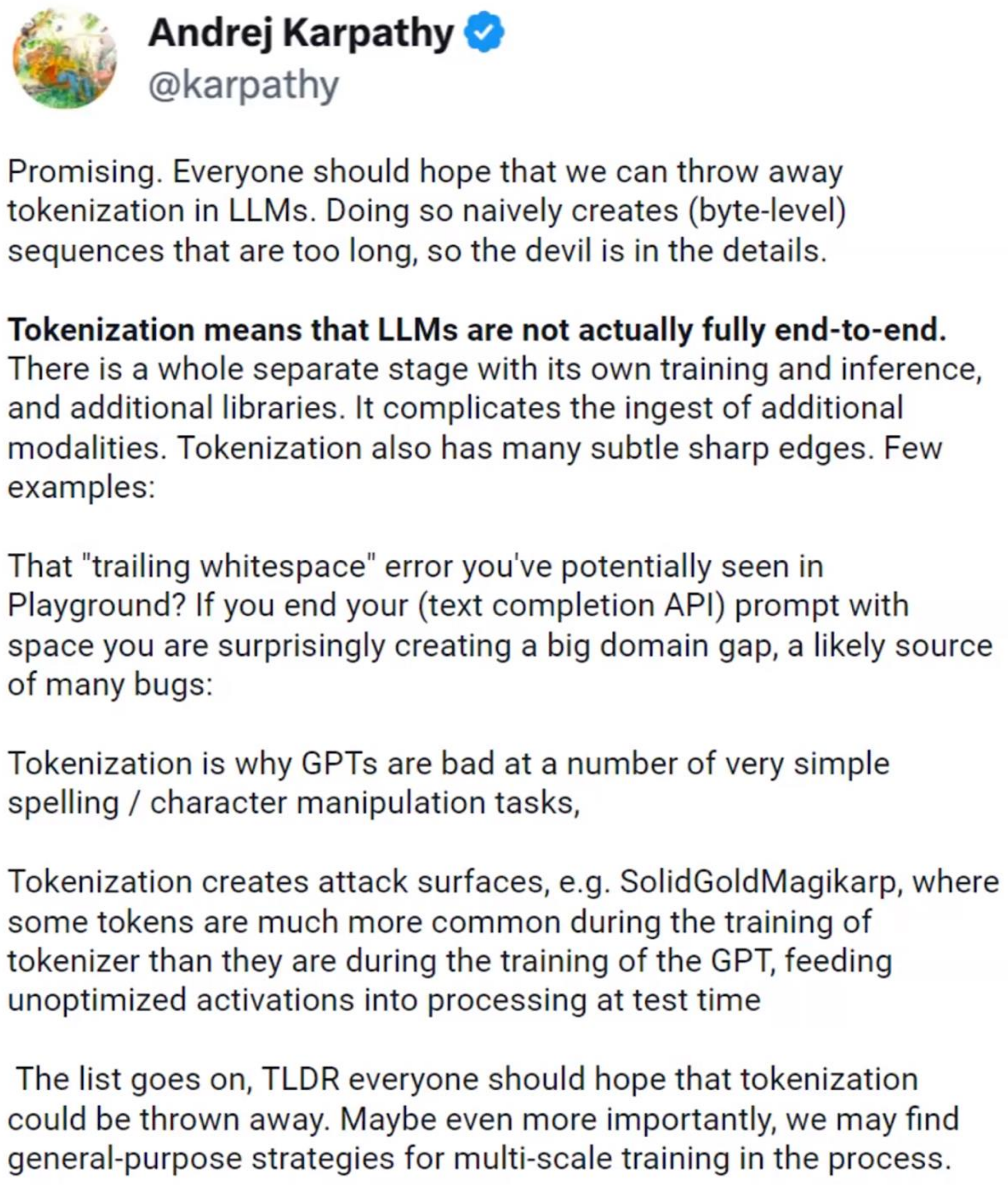
Tokenization creates attack surfaces, e.g. SolidGoldMagikarp, where some tokens are much more common during the training of tokenizer than they are during the training of the GPT, feeding unoptimized activations into processing at test time

The list goes on, TLDR everyone should hope that tokenization could be thrown away. Maybe even more importantly, we may find general-purpose strategies for multi-scale training in the process.

Preprocessing Text

- Want to process text with a model
- Text (e.g. news article) is one long sequenc
  - characters?
  -
  -
- Wh to
  -
  -
  -
  -
  - //in
- Preprocessing has a big impact on model pe irrelevant

Preprocessing means neural networks are not trained end-to-end (i.e. the loss is not backpropagated through the *entire* computation).


Andrej Karpathy
@karpathy

Promising. Everyone should hope that we can throw away tokenization in LLMs. Doing so naively creates (byte-level) sequences that are too long, so the devil is in the details.

Tokenization means that LLMs are not actually fully end-to-end. There is a whole separate stage with its own training and inference, and additional libraries. It complicates the ingest of additional modalities. Tokenization also has many subtle sharp edges. Few examples:

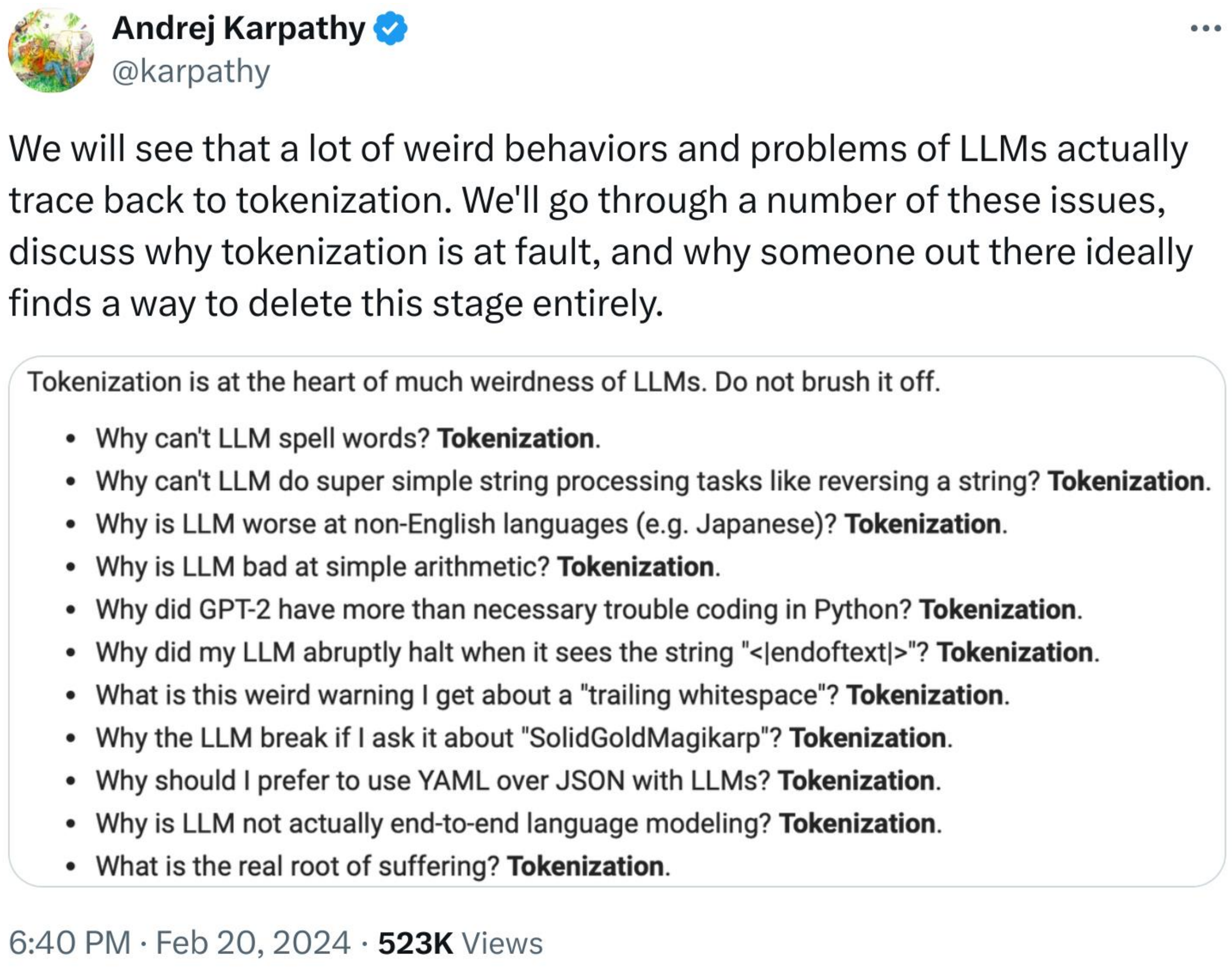
That "trailing whitespace" error you've potentially seen in Playground? If you end your (text completion API) prompt with space you are surprisingly creating a big domain gap, a likely source of many bugs:

Tokenization is why GPTs are bad at a number of very simple spelling / character manipulation tasks,

Tokenization creates attack surfaces, e.g. SolidGoldMagikarp, where some tokens are much more common during the training of tokenizer than they are during the training of the GPT, feeding unoptimized activations into processing at test time

The list goes on, TLDR everyone should hope that tokenization could be thrown away. Maybe even more importantly, we may find general-purpose strategies for multi-scale training in the process.

Preprocessing Text



Andrej Karpathy
@karpathy

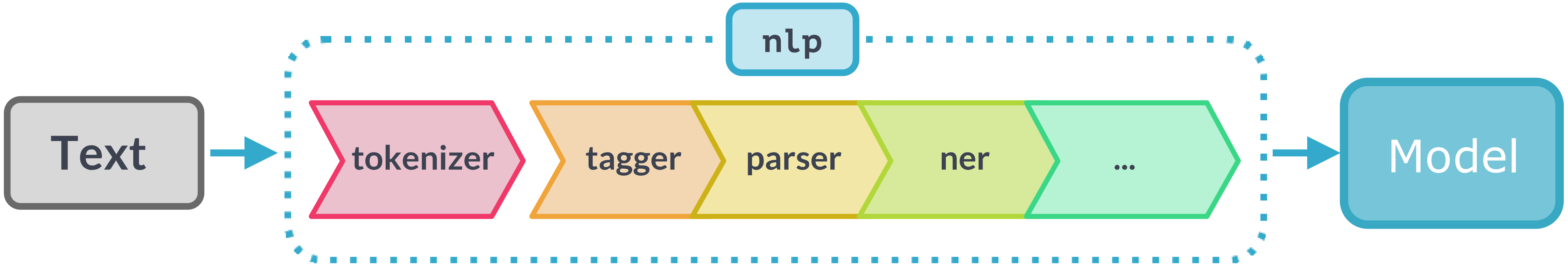
We will see that a lot of weird behaviors and problems of LLMs actually trace back to tokenization. We'll go through a number of these issues, discuss why tokenization is at fault, and why someone out there ideally finds a way to delete this stage entirely.

Tokenization is at the heart of much weirdness of LLMs. Do not brush it off.

- Why can't LLM spell words? **Tokenization**.
- Why can't LLM do super simple string processing tasks like reversing a string? **Tokenization**.
- Why is LLM worse at non-English languages (e.g. Japanese)? **Tokenization**.
- Why is LLM bad at simple arithmetic? **Tokenization**.
- Why did GPT-2 have more than necessary trouble coding in Python? **Tokenization**.
- Why did my LLM abruptly halt when it sees the string "<|endoftext|>"? **Tokenization**.
- What is this weird warning I get about a "trailing whitespace"? **Tokenization**.
- Why the LLM break if I ask it about "SolidGoldMagikarp"? **Tokenization**.
- Why should I prefer to use YAML over JSON with LLMs? **Tokenization**.
- Why is LLM not actually end-to-end language modeling? **Tokenization**.
- What is the real root of suffering? **Tokenization**.

6:40 PM · Feb 20, 2024 · **523K** Views

NLP Pipeline

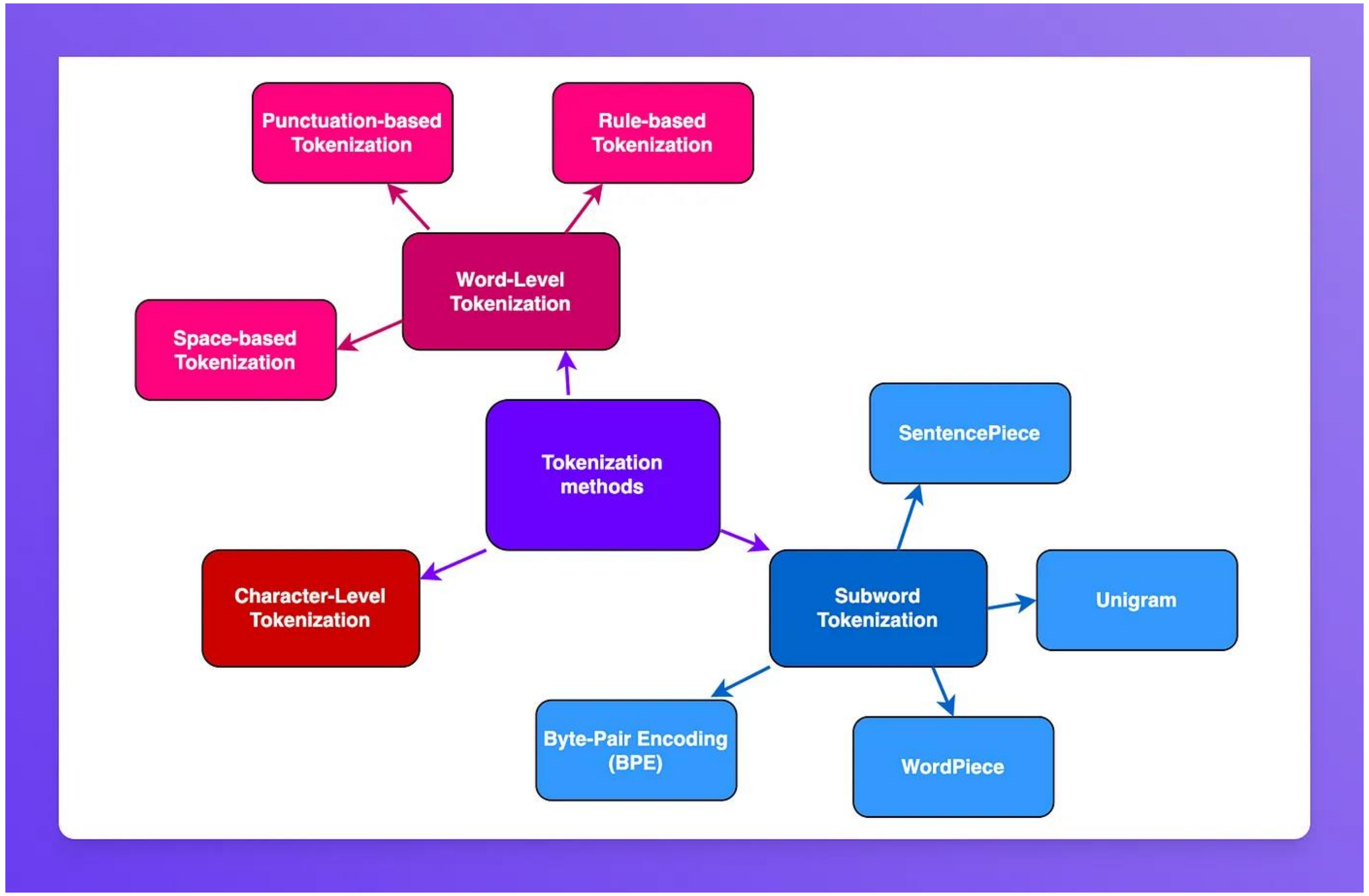


Tokenization

- Splits text sequence (sentence, paragraph, document) into parts
- Trivial?
- Sentence: Split on periods?
  - *On Monday, 18.09.2023, Mr. Smith started his website www.let-ai-study-for-me.com.*
- Words: Split on whitespace?
  - *それは日本人には当てはまりません！*
- Typically we process short text sequences at the level of words …
  - Except "short" no longer true: GPT-4 with 32k tokens context
  - Except "words" not true: Process at subword level (most often BPE tokens)

HSLU

Tokenization Methods



BPE Tokenization

- BPE = Byte pair encoding
  - See also: https://en.wikipedia.org/wiki/Byte_pair_encoding
- Algorithm
  - Look at the entire dataset
  - Every character gets a vocab entry
  - Add a vocab entry for the most common pair of characters
  - Replace the pair with a new symbol
  - Repeat for X merge operations
- Vocab sizes
  - GPT-3.5, GPT-4: 100k
  - LLaMA 3.1: 128k
  - Gemma: 256k

BPE Tokenization Example

- Data: abc abc baba

- Most common pair: ab (appears 3 times).
- Replace with X
- Xc Xc bXa → new most common pair: Xc = Y
- Y Y bXa → Y_ = Z
- ZZbXa
- No further compression

HSLU

NLTK Tokenizer

- NLTK = Natural Language Toolkit, an NLP library
- Word and sentence tokenization
- Rule-based, accumulated over many years, fast

```
import nltk

nltk.sent_tokenize("On Monday, 18.09.2023, Mr. Smith started his website www.let-
ai-study-for-me.com. It was a huge success!")

['On Monday, 18.09.2023, Mr. Smith started his website www.let-ai-study-for-
me.com.', 'It was a huge success!']

nltk.word_tokenize("The quick brown fox jumps over the lazy dog.")

['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog', '.']
```

HSLU

Hugging Face Tokenizers

- Tokenizers translate text into vectors that the model can process
- Different models have different tokenization strategies
  - The tokenizer is now part of the model

Sentence: "I study in Rotkreuz."

```
# BERT tokenizer
['[CLS]', 'i', 'study', 'in', 'rot', '##kr', '##eu', '##z', '.', '[SEP]']
# BART tokenizer
['<s>', 'I', 'Ġstudy', 'Ġin', 'ĠRot', 'k', 're', 'uz', '.', '</s>']
# T5 tokenizer
['▁I', '▁study', '▁in', '▁Rot', 'kreuz', '.', '</s>']
```

HSLU

Stemming

- Removes anything but the word stem
  - forgive → forgiv
  - houses → hous
  - itemization → item
  - flies → fli
  - dies/dying → die

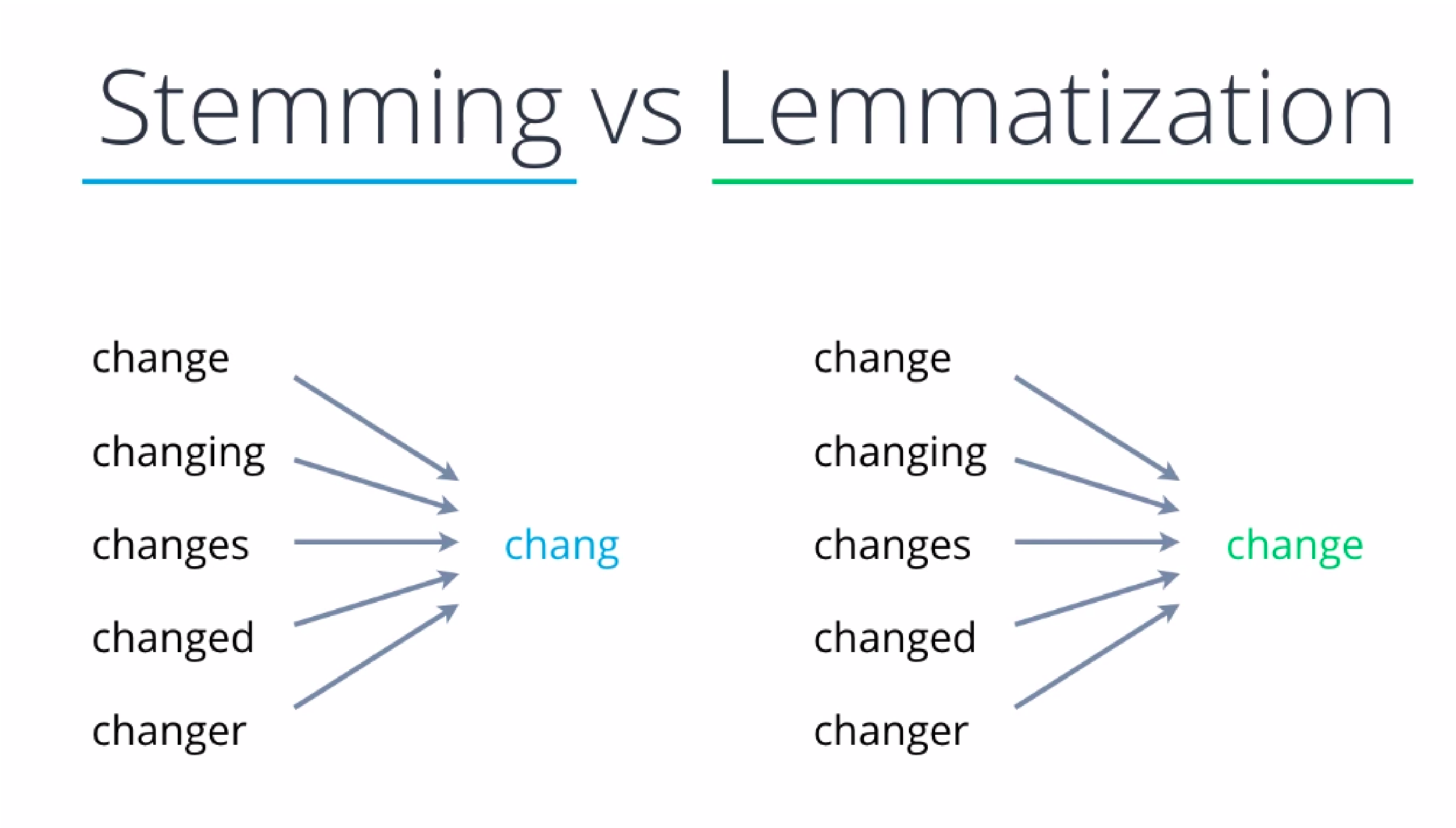
```
import nltk
stemmer = nltk.stem.PorterStemmer()
# or use: stemmer = nltk.stem.SnowballStemmer('english')
stemmer.stem('obviously')
'obvious'
```

Lemmatization

- Transforms all inflections of a word into its base form
  - forgive → forgive
  - houses → house
  - itemization → itemization
  - flies → fly
  - dies → dy (color something), dying → dying

```
import nltk
lemmatizer = nltk.stem.WordNetLemmatizer()
lemmatizer.lemmatize('flies')
'fly'
```

Stemming vs. Lemmatization



Part-of-speech (POS) Tagging

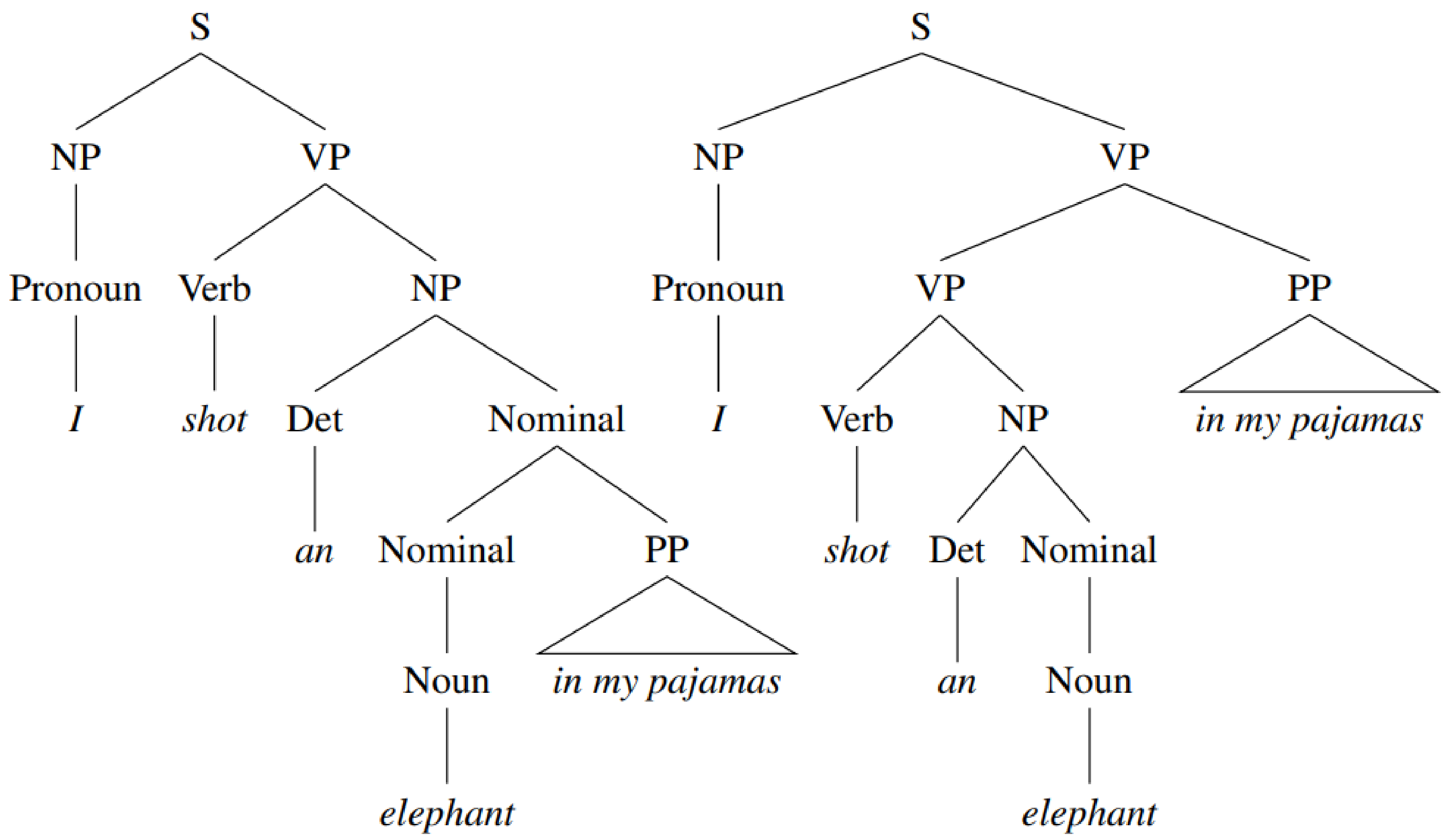
- Determine part of speech of each word
- Different tagsets
  - Penn Treebank tags
  - Universal POS (UPOS) tags

```
import nltk

words = nltk.word_tokenize("A brown dog chases a quick cat.")

nltk.pos_tag(words)

[('A', 'DT'), ('brown', 'JJ'), ('dog', 'NN'), ('chases', 'VBZ'), ('a', 'DT'),
('quick', 'JJ'), ('cat', 'NN'), ('.', '.')]

nltk.help.upenn_tagset('JJ')

JJ: adjective or numeral, ordinal

[…]
```

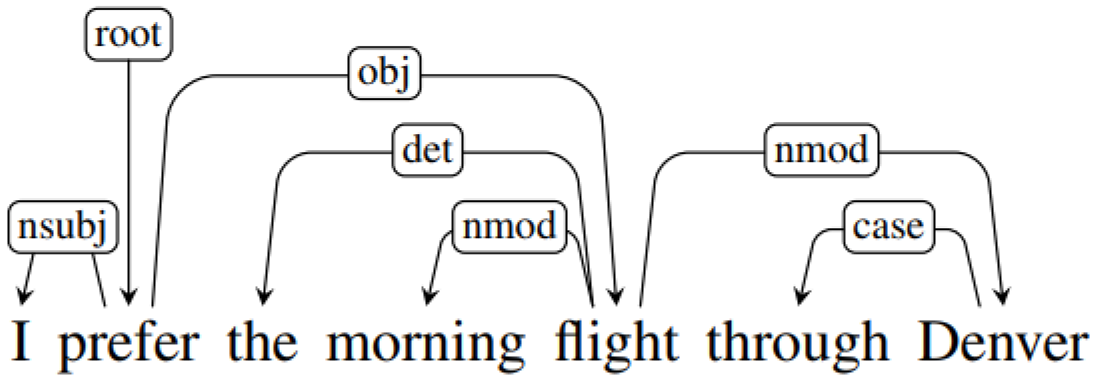
Parsing

- Parsing: Describe the syntactic structure of a sentence
- Constituency parsing
  - Groups of words can behave as single units (*constituents*)
- Dependency parsing
  - Directed grammatical relations between words
  - Labeled with an arc from a *head* to a *dependent*

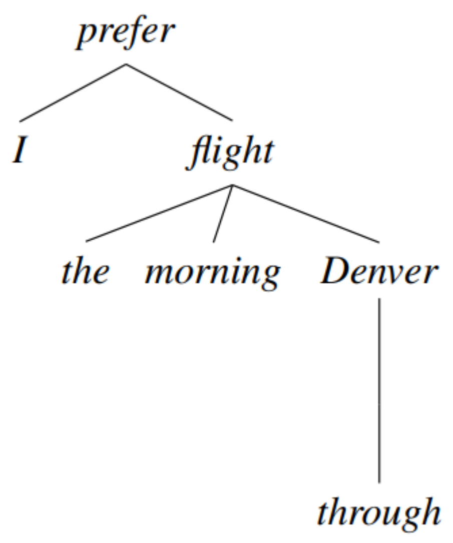
Constituency Parsing

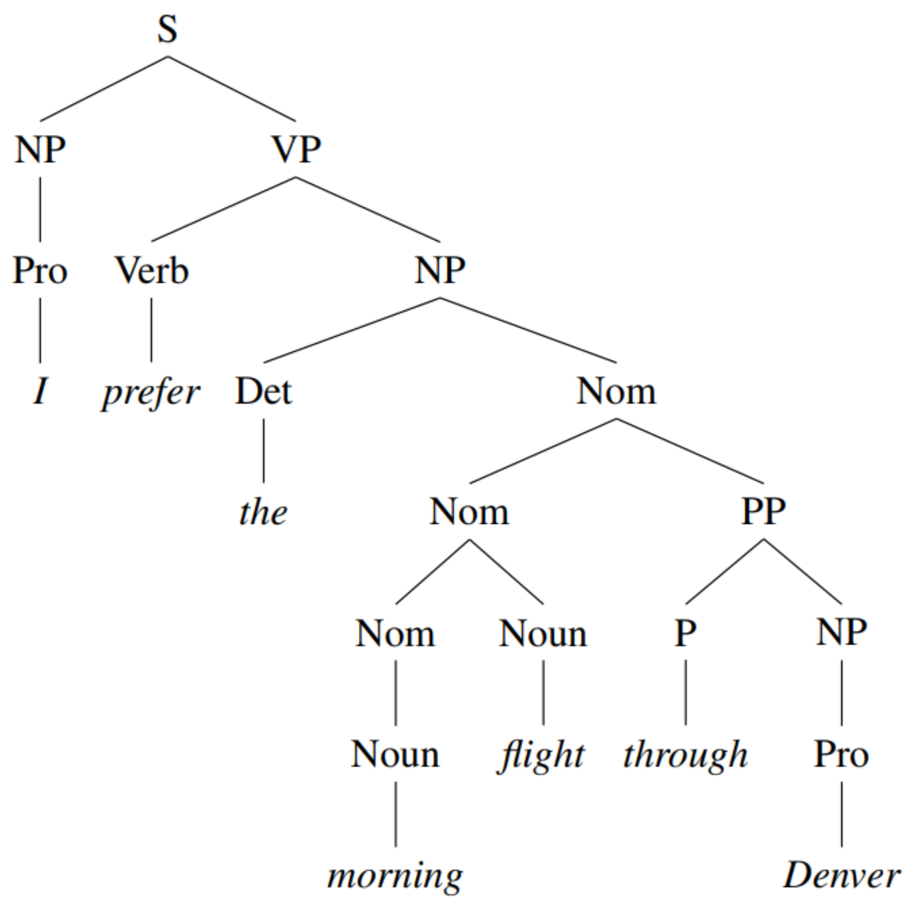


Dependency Parsing



Dependency vs. Constituency Parsing





Dependency Parsing with NLTK

```
sentence =
"""1       I       _       NN      NN      _       2       nn      _       _
2    shot    _       NN      NN      _       0       null    _       _
3    an      _       AT      AT      _       2       dep     _       _
4    elephant        _       NN      NN      _       7       nn      _       _
5    in      _       NN      NN      _       7       nn      _       _
6    my      _       NN      NN      _       7       nn      _       _
7    pajamas _       NNS     NNS     _       3       dobj    _       _"""
graph = nltk.parse.DependencyGraph(sentence, top_relation_label='null')
graph.tree().pprint()
(shot I (an (pajamas elephant in my)))
```

HSLU https://www.nltk.org/howto/dependency.html

Named Entities

- Named Entity Recognition (NER)
  - Discover “New York Times” as an entity
- Entity Linking
  - Find that “US president” and “Joe Biden” reference the same entity (in 2024)

```
import nltk
sentence = "U.S. President Joe Biden
arrived at London Heathrow airport this
morning."
words = nltk.word_tokenize(sentence)
tagged = nltk.pos_tag(words)
print(nltk.ne_chunk(tagged))
```

```
(S
  (GPE U.S./NNP)
  President/NNP
  (PERSON Joe/NNP Biden/NNP)
  arrived/VBD
  at/IN
  (ORGANIZATION London/NNP Heathrow/NNP)
  airport/NN
  this/DT
  morning/NN
  ./.)
```

Exercise: NLP Pipeline