



Natural Language Processing and Large Language Models

25/11/2025

Shamsuddeen Muhammad
Google DeepMind Academic Fellow,
Imperial College London
<https://shmuhammadd.github.io/>

Idris Abdulkumin
Postdoctoral Research Fellow,
DSFSI, University of Pretoria
<https://abumafrim.github.io/>

Who is Shamsuddeen?

- BSc CS at Bayero University, Kano Nigeria

- MSc CS, University of Manchester, UK



- PhD Machine Learning, University of Porto



- Advanced Research Fellow/Google DeepMind Fellow, Imperial College London
- Senior Lecturer, Bayero University

Imperial College London NLP and LLMs at AIMS South Africa 2025



2nd in the world

1st in the UK and Europe

QS World University Rankings 2025

<https://shmuhammadd.github.io/>

25/11/2025

Who is Idris?

- BSc CS at Bayero University, Kano Nigeria
 BAYERO UNIVERSITY KANO
- MSc CS, Nottingham Trent University, UK
 Nottingham Trent University
- PhD CS (Machine Translation), Bayero University, Kano Nigeria
 BAYERO UNIVERSITY KANO
- Postdoctoral Research Fellow, DSFSI University of Pretoria, South Africa



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopololo tsa Dihlaefi

<https://abumafrim.github.io/>

About the Course !

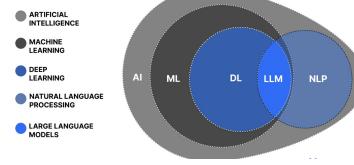
- Natural language processing (NLP) is the field of working with language to automatically perform a variety of tasks, instead of or in collaboration with people.
- Recently, large language models (LLMs) like ChatGPT have gotten the attention of the general public, but they have also greatly changed the landscape of modern NLP research.
- This course will show you **both old & new techniques** that are used today and will give you a basic understanding of why & how we do NLP.

Prerequisite

- Python
- Machine Learning
- Deep Learning
- Foundational understanding of PyTorch, or familiarity with other deep learning frameworks like TensorFlow, will be beneficial.

About the Course !

NLP vs LLMs



- **NLP** applies a combination of rule-based systems and machine learning to process text and speech efficiently.
- **LLMs**, rely on deep learning for language (knowledge) comprehension

About the Course !

NLP vs LLMs

Aspect	Natural Language Processing (NLP)	Large Language Models (LLMs)
Data Requirement	Structured, labelled data	Large-scale, unstructured datasets
Computational Power	Low to moderate; can run on local machines	High-performance GPUs and cloud-based processing
Primary Use Cases	Sentiment analysis, translation, speech recognition, text classification	Conversational AI, content creation, coding assistance, document summarization
Flexibility	Task-specific and specialized	Adaptable across domains and capable of handling diverse queries
Cost	Lower infrastructure demands; more cost-effective	High due to extensive computational and storage requirements
Scalability	Easily scalable for structured applications	Requires significant cloud-based resources to scale effectively

Learning Objective

By the end of the course, you will be able to...

- Foundations of NLP & LLMs:** Learn key concepts such as tokenization, embeddings, language modelling, and the core architecture and training mechanics behind transformer-based LLMs.
- Practical NLP Applications:** Apply models to tasks like text classification, machine translation, summarization, and zero-/few-shot prompting using real-world datasets.
- Hands-On Model Development:** Build NLP pipelines, evaluate multilingual performance (with focus on low-resource languages), and develop a simple language model from scratch.

Course Content

Week 1: Foundations of NLP & Classical Approaches

Core principles of NLP, including text preprocessing, tokenization, n-gram language models, word embeddings, and sequence-to-sequence modelling.

Week 2: Transformers, LLM, & Fine-Tuning

Transformer architecture, attention mechanisms, pre-training and transfer learning, supervised fine-tuning, prompting strategies, and evaluation of LLMs.

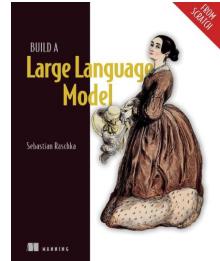
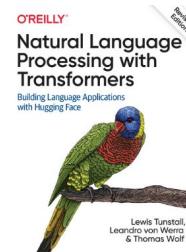
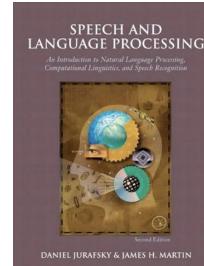
Week 3: Advanced Topics, & Ethical NLP

Advanced Topics, & Ethical NLP

We will be discussing state-of-the-art papers about large language models.

Reference Books

- Course page: <https://github.com/shmuhammad/aims-nlp-course>
- Assignment descriptions



Reference Books NLP

1. Speech and Language Processing, [Dan Jurafsky](https://web.stanford.edu/~jurafsky/slp3/) and James H. Martin
<https://web.stanford.edu/~jurafsky/slp3/>
2. Foundations of Statistical Natural Language Processing, [Chris Manning](#) and [Hinrich Schütze](#)
3. Build a Large Language Model (From Scratch): <https://github.com/rasbt/LLMs-from-scratch>
4. Hands-On Large Language Models: <https://github.com/HandsOnLLM/Hands-On-Large-Language-Models>

Other Sources

Journals

Computational Linguistics, Natural Language Engineering, TACL, JMLR, TMLR, etc

Conferences

ACL, EMNLP, NAACL, COLING, AAAI, IJCNLP, ICML, NeurIPS, ICLR, WWW, KDD, SIGIR, etc

ACL Anthology

ACL Anthology is a collection of papers from the Association for Computational Linguistics (ACL) conferences and workshops. It includes papers from the following events:

Venue	2022 – 2020	2019 – 2010	2009 – 2000	1999 – 1990	1989 and older
ACL	20				
ACL	22	21	19	18	17
CL	22	21	20	19	18
COLNL	21	20	19	18	17
EACL	21	20	19	18	17
EMNLP	21	20	19	18	17
Findings	22	21	20	19	18
HLNLP	22	21	20	19	18
NAACL	21	20	19	18	17
SentEval	22	21	20	19	18
*SEM	22	21	20	19	18
TACL	22	21	20	19	18
WMT	21	20	19	18	17
SIG	21	20	19	18	17

ACL Anthology currently hosts 77778 papers on the study of computation, language, linguistics and natural language processing.

Subscribe to the mailing list to receive announcements and updates to the Anthology.

Full Anthology as BibTeX (462.5 MB) with abstracts (17.33 MB) (last updated: 10/01/2023)

Our Holdings

ACL Anthology | FAQ | Corrections | Submissions | Search... |

ArXiv

arXiv.org > cs > cs.CL

Computation and Language

Authors and titles for recent submissions

- Wed, 19 Aug 2020
- Tue, 18 Aug 2020
- Mon, 17 Aug 2020
- Fri, 14 Aug 2020
- Thu, 13 Aug 2020

[total of 84 entries: 1–25 | 26–50 | 51–75 | 76–84]
[previous page | next page | first | last | page first | page last]

Wed, 19 Aug 2020

[arXiv:2008.07915 [pdf, other]]

Glancing Transformer for Non-Autoregressive Neural Machine Translation

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, Lei Li

Comments: 10 pages, 3 figures, 1 table

Subjects: Computation and Language (cs.CL)

[arXiv:2008.07840 [pdf, other]]

COVID-SEE: Scientific Evidence Explorer for COVID-19 Related Research

Karin Vesipour, Simon Suster, Yuliya Otnakova, Shervin Mendezi, Zhenan Zhai, Biaoany Fang, Jey Han Lau, Timothy Bal

Comments: 1 page, 1 figure, 1 table

Subjects: Computation and Language (cs.CL); Information Retrieval (cs.IR)

[arXiv:2008.07772 [pdf, other]]

Very Deep Graph Transformers for Neural Machine Translation

Xiaoxiao Liu, Zhen Chen, Liyanan Liu, Jianfeng Gao

Comments: 6 pages, 3 figures and 3 tables

Subjects: Computation and Language (cs.CL)

[arXiv:2008.07723 [pdf, other]]

NASE: Learning Knowledge Graph Embedding for Link Prediction via Neural Architecture Search

Xiaoyu Kou, Bingfeng Luo, Huang Hu, Yan Zhang

Comments: accepted by CMLR 2020, short paper

Subjects: Computation and Language (cs.CL)

<https://arxiv.org/list/cs.CL/recent>

Imperial College London NLP and LLMs at AIM3 South Africa 2025

14

25/11/2025

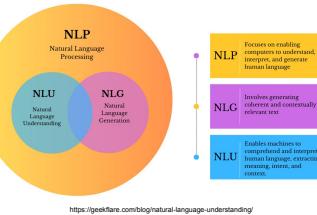
Acknowledgments

- Advanced NLP, Graham Neubig <http://www.phontron.com/class/anlp2022/>
- Advanced NLP, Mohit Iyyer <https://people.cs.umass.edu/~miyyer/cs685/>
- NLP with Deep Learning, Chris Manning, <http://web.stanford.edu/class/cs224n/>
- Understanding Large Language Models, Danqi Chen <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/>
- Natural Language Processing, Greg Durrett <https://www.cs.utexas.edu/~qdurrett/courses/online-course/materials.html>
- Large Language Models: <https://stanford-cs324.github.io/winter2022/>
- Natural Language Processing at UMBC, <https://laramartin.net/NLP-class/>
- Computational Ethics in NLP, https://demo.clab.cs.cmu.edu/ethical_nlp/
- Self-supervised models, CS 601.471/671: Self-supervised Models (jhu.edu)
- WING.NUS Large Language Models, <https://wing-nus.github.io/cs6101>

What is Natural Language Processing (NLP)?

What is Natural Language Processing (NLP) ?

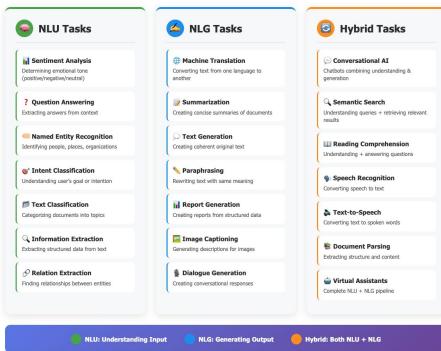
- **Natural Language Processing (NLP)** is a field of artificial intelligence focused on enabling machines to **understand, interpret, and generate** human language.
- NLP combines methods from linguistics, machine learning, and computer science to build systems that work with **text and speech** at scale.
- NLP includes two major subfields:
 - **NLU – Natural Language Understanding:** extracting meaning, intent, entities, and structure.
 - **NLG – Natural Language Generation:** producing coherent, context-appropriate text or speech.



NLU vs NLG

Two side-by-side screenshots comparing NLU and NLG. The left screenshot, titled 'NLU Natural Language Understanding', shows a 'Sentiment Analysis' interface with an input text 'The movie was absolutely amazing! I loved every minute of it', an analysis result (Sentiment: Positive, Confidence: 95%, Emotion: Joy, Polarity: 1.0), and a 'Question Answering' interface with a question 'When was Einstein born?' and an answer 'Albert Einstein was born in 1879 in Germany. He developed the theory of relativity.' The right screenshot, titled 'NLG Natural Language Generation', shows a 'Machine Translation' interface with an English input 'Hello, how are you today?' and a Spanish output 'Hola, como estás hoy?', and a 'Summarization' interface with a text input 'Climate change is one of the most pressing issues of our time. The effects of climate change are causing loss of habitat, loss of life, and extreme weather events. If we do not take immediate action, the consequences could be catastrophic for future generations.' and a summary output 'Climate change threatens the planet with melting ice caps, rising sea levels, and extreme weather events, requiring immediate action to prevent catastrophic consequences.'

NLU vs NLG



Natural Language Processing and Computational Linguistics



Both fields work with human language using computers, but they have different goals and perspectives!

Natural Language Processing (NLP)

- Goal:** Build practical applications that solve real-world problems involving human language.
- Focus:** "How can we make computers DO things with language?"
- Real-World Applications:**
- Speech Recognition: Converting voice to text (Siri, Alexa)
 - Machine Translation: Google Translate, DeepL
 - Information Extraction: Pulling key facts from documents automatically
 - Chatbots: Customer service bots, virtual assistants
 - Spam Detection: Filtering unwanted emails

Computational Linguistics (CL)

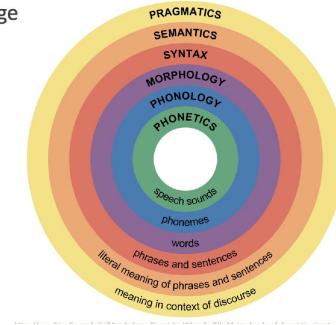
- Goal:** Understand how human language works using computational methods and models.
- Focus:** "How does language actually WORK in the human mind and brain?"
- Research Questions:**
- How do we understand language? Example: How do children learn grammar rules without being explicitly taught?
 - How do we produce language? Example: How does the brain decide which words to use in which order?
 - How do we learn language? Example: What makes some languages easier to learn than others?
 - What are universal language structures? Example: Are there grammar rules common to all human languages?

Natural Language Processing and Computational Linguistics

- Most of the conferences and journals that host natural language processing research bear the name “computational linguistics” (e.g., ACL, NACL).
- NLP and CL may be thought of as essentially synonymous.
- While there is substantial overlap, there is an important difference in focus
 - CL is essentially linguistics supported by computational methods (similar to computational biology, and computational astronomy)
 - NLP focuses on solving well-defined tasks involving human language (e.g., translation, query answering, holding conversations).

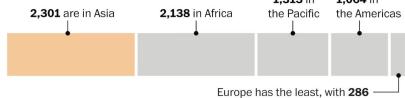
Linguistics

The study of language



Why is NLP interesting?

There are at least 7,102 living languages in the world.

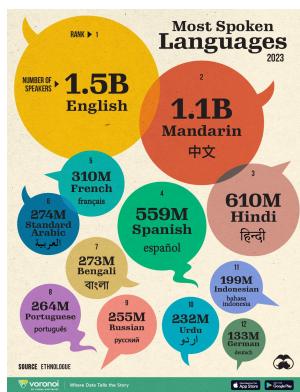


Europe has the least, with 286

NLP powers a broad range of applications:

Machine translation, information extraction, question answering, summarization, sentiment analysis, speech technologies, code generation, document retrieval, fact checking, etc.

The field is also rapidly evolving with the rise of large language models, offering new research challenges



Before Building NLP Systems...

- To build NLP systems, we need **large, high-quality text data**.
- But the world's **7,000+ languages** are not equally represented in digital form.
- This leads to major gaps: many languages have **little or no available corpus**, making them "**low-resource languages** (e.g., **African Languages**.)"
- Before building models, we must understand **what a corpus is, how it is collected, and why it matters** for multilingual NLP.

Where does the data come from?

Corpus and Low-Resource Languages

- **Corpus** (plural: *corpora*): a structured collection of text used for training or evaluating NLP models.
- Languages with limited corpora are known as **low-resource languages**.

How Corpora Are Collected

- **Expert-curated data**: manually tagged and organized by linguists or annotators.
- **Open-web data**: collected from freely accessible sources (e.g., Wikipedia, blogs, forums).
- **Permission-based data**: obtained from closed platforms (e.g., messaging apps, social media) with explicit consent—less common but often higher quality.

Data creation in Africa



Please, consider giving your feedback on using Lanfrica so that we can know how best to serve you. To get started, click here.



Lanfrica catalogues and links African language resources in order to mitigate the difficulty encountered in discovering African works.

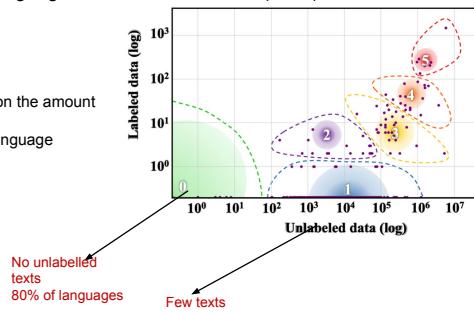
[Browse Records](#) [Join our Slack](#)

Under-resourced languages: Labelled+Unlabelled data

Six-class categorization of languages based on Joshi et al (2020)

- Unlabelled corpora
- Labelled corpora

categorization of languages based on the amount of NLP resources available for each language



Six-class categorization

Highly Resourced (HRL)- Winners

- Languages with extensive NLP resources, including large-scale corpora, pre-trained models, and strong computational tools.
- Examples: English, Chinese, Spanish, French

Moderately Resourced (MRL) - Moderate

- Languages with reasonable NLP resources, but still lacking in some areas such as large-scale pre-trained models.
- Examples: Dutch, Russian, Korean

Somewhat Resourced (SRL) - Hopefuls

- Languages with limited but growing NLP resources, including some datasets and a few pre-trained models.
- Examples: Swahili, Finnish, Turkish

Six-class categorization

Low Resourced (LRL) - Scraping By

- Languages with very limited annotated data, small corpora, and minimal computational resources.
- Examples: Hausa, Tamil, Uzbek

Extremely Low Resourced (XLRL) - Left Behind

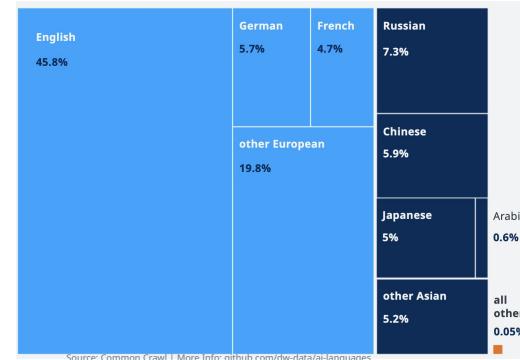
- Languages with almost no NLP resources, where some digitized text may exist, but annotated corpora and NLP tools are scarce.
- Examples: Wolof, Aymara, Tigrinya

Unsupervised (UL) - The Rest

- Languages with virtually no digital footprint, requiring unsupervised or few-shot learning techniques for any NLP progress.
- Examples: Many indigenous and endangered languages like Chadic languages, Amazonian languages

Lack of Publicly Available Dataset

languages in the Common Crawl internet archive



30%

World languages are African (Ethnologue)

0.05%

Why is NLP hard?

Why is NLP hard?

Ambiguity

NLP is challenging due to the inherent complexities of human language.

Lexical Ambiguity

A single word can have multiple meanings, and the intended meaning depends on the context.

Example:

The **bank** is closed today. (Does "bank" refer to a financial institution or the side of a river?)

Syntactic Ambiguity

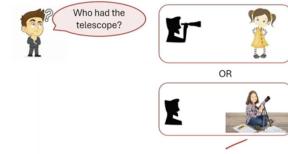
A sentence can have more than one valid structure, leading to different interpretations.

Example:

I saw a girl with the telescope.

Possible meanings:

- I used a telescope to see the girl.
- The girl is the one holding the telescope.



Ambiguity in Language

- I ate food with **Spoon**

"with" = tool used to eat



- I ate rice with **curd**

"with" = ingredient served together



- I ate rice with **Muhammad**

"with" = person I ate together with



Semantic Ambiguity

A sentence can have more than one meaning because its interpretation depends on context.

Example:

"The chicken is ready to eat."

Possible interpretations:

- The chicken is **going to eat** something
- The chicken is **cooked and ready to be eaten**.

Pragmatic Ambiguity

Meaning depends on context, social norms, and the speaker's intention, not just the words themselves.

Example:

“Can you pass the salt?”

Two interpretations:

- **Literal:** “Are you able to pass the salt?” (asking about ability)
- **Pragmatic:** A polite way to say “**Please pass the salt.**”

Ambiguity in Punctuation

A woman without her man is nothing.



A woman, without her man, is nothing.
A woman: without her, man is nothing.

Punctuation is powerful.

How NLP Overcomes Language Challenges

Human language is full of challenges: lexical, syntactic, semantic, and pragmatic ambiguities.

These difficulties vary across languages and are even more complex in **low-resource** contexts.

But despite these challenges:

- NLP provides tools to interpret meaning.
- resolve ambiguity using context and large datasets,
- and build systems that can understand and generate language with high accuracy.

In this course, we will explore how modern NLP, especially transformers and LLMs, addresses these challenges and how these methods can be applied to many languages, including those with limited resources.

Break

NLP Layers

- Understanding the semantics is a non-trivial task.
- Needs to perform a series of incremental tasks to achieve this.
- NLP happens in layers.

Pragmatics & Discourse	<i>Study of semantics in context.</i>
Semantics	<i>Meaning of the sentence.</i>
Parsing	<i>Syntactic structure of the sentence.</i>
Chunking	<i>Grouping of meaningful phrases.</i>
Part of speech tagging	<i>Grammatical classes.</i>
Morphology	<i>Study of word structure.</i>



Increasing
Complexity Of
Processing

Morphology

Morphology is the study of how words are formed and structured.

It looks at the smallest meaningful units (called **morphemes**) like prefixes, suffixes, and root words.

Different languages handle morphology differently -

some use very little word modification (morphologically poor), while others heavily modify words (morphologically rich).

Morphology helps computers understand how words are built and there are two types/

Inflectional Morphology and Derivational Morphology

Inflectional Morphology

Changes the form of a word (tense, number, gender) without changing its core meaning.

Examples: walk → walked (past tense), cat → cats (plural)

Why this matters for NLP:

- **Tokenization & vocabulary size:** Inflected forms multiply the number of word types models must handle.
- **Lemmatization:** Systems need to group different forms into the same base word (*walk*).
- **POS tagging & parsing:** Inflections signal tense, plurality, agreement, etc.
- **Low-resource languages:** Richly inflected languages (e.g., Amharic, Arabic, Hausa) create data sparsity.

Derivational Morphology

Derivation creates **new words** with related meanings:

- *happy* → *unhappy*
- *teach* → *teacher*
- *kind* → *kindness*

Why this matters for NLP:

- **Word embeddings:** Models must learn that *teach* and *teacher* are related but not identical.
- **Sentiment analysis:** Prefixes like *un-* or *dis-* change polarity.
- **Text classification:** Derivational patterns signal topic or domain (*biology*, *biological*, *biologist*).

Morphology

- English, Chinese, etc. are commonly referred as *morphologically-poor* language.
- Hindi, Turkish, Hungarian, etc. are termed as *morphologically-rich* language.

Morphologically Poor	Morphologically Rich
 Little word inflection	 Heavy word inflection
Examples: English, Chinese English  Chinese 	Examples: Swahili, Hausa, Zulu Swahili  Hausa  Zulu 
Characteristics: <ul style="list-style-type: none">✓ Words stay mostly the same✓ Minimal prefixes/suffixes✓ Meaning comes from word order✓ Fewer total word forms	Characteristics: <ul style="list-style-type: none">✓ Verbs change extensively✓ Many prefixes/suffixes✓ Information encoded in word form✓ Thousands of possible word forms
English Example: "I go" → "You go" → "We go" (Only "he/she/they" changes)	African Languages Example: Each subject (I/you/we/he/she) requires different verb forms!

Comparison: "I/We/You/He/She will go"

English	Swahili	Hausa	Zulu	What Changes?
I will go	 taenda	 tafi	 zoya	Verb changes based on subject prefix
We will go	 taenda	 tafi	 zoya	Number agreement (singular vs plural)
You will go	 taenda	 tafi	 zoya	Person agreement (1st/2nd/3rd person)
He will go	 taenda	 tafi	 zoya	Gender/person marking (in Hausa)
She will go	 taenda	 tafi	 zoya	Gender/person marking (in Hausa)

Why Does This Matter for NLP?

Challenge 1: Data Sparsity

Morphologically rich languages create thousands of word forms. A single English verb might have 3-4 forms, but Swahili could have hundreds! This means you need much more training data.

Challenge 2: Complex Encoding

NLP systems must learn that prefixes/suffixes encode important information like tense, number, and subject agreement - not just memorize whole words.

Challenge 3: Low-Resource Languages

Many morphologically rich languages (like African languages) have limited digital resources, making it harder to build effective NLP systems for them.

Key Takeaway: NLP models trained on English often fail on morphologically rich languages because they're not designed to handle complex word structures. We need specialized approaches for these languages!

Word and Token

- A **word** is the smallest meaningful unit of language that can stand alone.
- A **token** is a unit of text obtained after tokenization, which is the process of breaking a text into individual components (words, subwords, or even characters).
- A token may or may not correspond directly to a word.

Word and Token

Sentence: I love NLP research.

Words: I, love, NLP, research | **Tokens** (depending on tokenizer):

- **Word-level:** I, love, NLP, research
- **Subword-level (BPE):** I, love, NL, P, research
- **Key idea:** Humans think in words. NLP models think in tokens, and these may not match human words exactly

*Tokenization is the process of breaking down text into smaller units, called **tokens**, which can be words, subwords, or characters.*

*It is a fundamental preprocessing step in **Natural Language Processing (NLP)** used for tasks such as machine translation, text classification, and language modeling.*

Word and Token

Sentence: I love NLP research.

Words: I, love, NLP, research | **Tokens** (depending on tokenizer):

- **Word-level:** I, love, NLP, research
- **Subword-level (BPE):** I, love, NL, P, research
- **Key idea:** Humans think in words. NLP models think in tokens, and these may not match human words exactly

*Tokenization is the process of breaking down text into smaller units, called **tokens**, which can be words, subwords, or characters.*

*It is a fundamental preprocessing step in **Natural Language Processing (NLP)** used for tasks such as machine translation, text classification, and language modeling.*

Types of Tokenization

There are four main types of tokenization used in Natural Language Processing, each serving different purposes and use cases:

Word Tokenization

Subword Tokenization

Character Tokenization

Sentence Tokenization

Word Tokenization

The text is split into individual words based on **whitespace** and **punctuation**.

Example:

Input:

I love NLP!

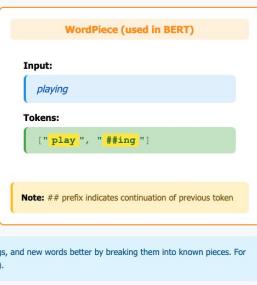
Tokens:

["I", "love", "NLP", "!"]

How it works: Splits text at spaces and separates punctuation marks. This is the simplest and most common tokenization method.

Subword Tokenization

Words are broken into smaller meaningful units, especially for handling **out-of-vocabulary (OOV)** words. Used in modern NLP models like **BERT**, **GPT**.



Character Tokenization

Each **character** is treated as a separate token. Useful for handling **unknown words** in languages with complex morphology (e.g., Chinese, Japanese, Korean).

Example:

Input:
`Hello`

Tokens:

`["H", "e", "l", "l", "o"]`

Character Tokenization

💡 Best Used For:

- ✓ Languages without clear word boundaries (Chinese: 你好世界)
- ✓ Handling misspellings and typos
- ✓ Working with very rare or unknown words
- ✓ Character-level language models

Trade-off: Creates very long sequences (more computation) but has smallest vocabulary size and zero out-of-vocabulary words!

Sentence Tokenization

The text is split into **sentences** instead of words. Useful in **summarization** or **translation** tasks where sentence boundaries matter.

💡 Example:

Input:

I love NLP. It is exciting!

Tokens:

`["I love NLP.", "It is exciting!"]`

Sentence Tokenization

Best Used For:

- ✓ Text summarization (process sentence by sentence)
- ✓ Machine translation (translate complete sentences)
- ✓ Sentiment analysis per sentence
- ✓ Document segmentation and analysis

Challenge: Not all periods mean end of sentence! Need smart algorithms to handle abbreviations (Dr, Mr, U.S.A.), decimals (3.14), and ellipsis (...).

Part-of-Speech Tagging (POS)

- **Part of Speech (PoS)** refers to the grammatical category of words in a sentence based on their function and meaning.
- **PoS** tagging is essential in NLP for understanding sentence structure and meaning.

Grammatical class of the word.

He	ate	an	apple	.
PRP	VBD	DT	NN	.

PoS disambiguation:

- A word can belong to different grammatical classes.

He	went	to	the	park	in	a	car	.
PRP	VBD	TO	DT	NN	IN	DT	NN	.
They	went	to	the	park	the	car	in	the
PRP	VBD	TO	NN	DT	NN	IN	DT	NN

Tags

PRP: Personal Pronoun
VBD: Verb, Past
DT: Determiner
NN: Noun, Singular, Mass
TO: to
IN: Preposition

- 45 tags in Penn Treebank tagset
- 146 tags in C7

Chunking

Chunking is the process of grouping words into meaningful phrases based on their Part of Speech (PoS) tags.

It helps in identifying syntactic structures like noun phrases (NP), verb phrases (VP), and prepositional phrases (PP).

Example of Chunking

Consider the sentence:

"The quick brown fox jumps over the lazy dog."

Chunking

Step 1: PoS Tagging

The (DT) quick (JJ) brown (JJ) fox (NN) jumps (VBZ) over (IN) the (DT) lazy (JJ) dog (NN)

Step 2: Chunking (Noun Phrases & Verb Phrases)

[NP The quick brown fox] [VP jumps] [PP over] [NP the lazy dog]

Here, the **Noun Phrases (NP)** and **Verb Phrases (VP)** are extracted.

Semantics

- **Semantics** is the study of meaning in language—how words, phrases, and sentences convey meaning.
- Semantic analysis helps NLP systems:
 - understand what a sentence means, not just what words it contains,
 - interpret words based on context (“bank” = money vs. riverbank),
 - capture relationships between words (who did what, to whom),
 - generate meaningful and coherent text.

Task we want to solve in NLP?

NLP Tasks

Understanding Tasks

Text Classification

Sentiment analysis, topic classification

Named Entity Recognition

Finding people, places, organizations

Part-of-Speech Tagging

Identifying grammatical roles

Dependency Parsing

Understanding sentence structure

Question Answering

Answering natural-language queries

Maching Reading Comprehension

Extracting meaning from text

Generation Tasks

Machine Translation

Converting text between languages

Summarization

Producing concise summaries

Text Generation

Writing coherent sentences or documents

Dialogue Systems / Chatbots

Human-like conversation

Paraphrasing

Rewriting text with same meaning

NLP Tasks

Speech & Multimodal Tasks

Speech Recognition

Speech → text conversion

Text-to-Speech

Text → speech conversion

Vision-Language Tasks

Image captioning, Visual Question Answering (VQA)

Low-Level / Core Tasks

Tokenization and Segmentation

Breaking text into words, sentences, or characters

Lemmatization and Stemming

Reducing words to their base form

Morphological Analysis

Analyzing word structure and inflections

Coreference Resolution

Determining who/what pronouns refer to

Basic NLP Tasks

Tokenization: Splitting text into words (word tokenization) or sentences (sentence tokenization).
Example:

Input: "NLP is amazing!"

Output: ['NLP', 'is', 'amazing', '!']

Stopword Removal: Eliminating common words like "the," "is," "in," etc. Helps in reducing noise in text analysis.

Example:

Input: "This is a great NLP tutorial."

Output: ['great', 'NLP', 'tutorial']

Basic NLP Tasks

Stemming reduces words to their base or **stem form**, often by chopping off suffixes. However, it may not always result in a meaningful word.

Stemmed Words:

- **cats** → **cat**
- **were** → **were** (unchanged)
- **running** → **run**
- **gardens** → **garden**
- **happily** → **happi** (incorrect stem)
- **injured** → **injur**

- Some words (like *happily* → *happi*) lose meaning due to aggressive chopping.
- It does not consider the proper root word, just removes common suffixes.

Lemmatization

Lemmatization converts words into their **dictionary base form (lemma)**, using linguistic knowledge.

Lemmatized Words:

- **cats** → cat
 - **were** → be (correct lemma)
 - **running** → run
 - **gardens** → garden
 - **happily** → happy
 - **injured** → injure
-
- More accurate and meaningful base forms.
 - Proper grammatical transformations (e.g., *were* → *be*, *happily* → *happy*).
 - Context-aware, ensuring the correct dictionary form.

Stemming vs Lemmatization?

Use Stemming when speed is important and minor errors are acceptable (e.g., search engines).

Use Lemmatization when **accuracy matters** (e.g., NLP applications like chatbots, text analysis).

IMPERIAL

Q and A