ACADEMIC YEAR 2026/27

Text Mining & Search [DS] / Information Retrieval [CS]

Part-of-Speech (POS) Tagging and Sequence Labeling

TEXT MINING & SEARCH

Prof. Marco Viviani

University of Milano-Bicocca

Department of Informatics, Systems and Communication

INFORMATION RETRIEVAL

Dr. Georgios Peikos

University of Milano-Bicocca

Department of Informatics, Systems and Communication

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Ikr3 Lab — information and knowledge representation, retrieval, and reasoning LAB

So far

So Far – Processing, representation, and modeling

- Text processing involves text preprocessing, tokenization, and word normalization, among other.
- Text representation converts lexical content into computational forms (e.g. vectors).
- Language modeling assigns probabilities to token sequences.
- Today we add grammatical labels to tokens.
  - Today we predict categories such as NOUN and VERB instead of predicting the next word.
    - Today we will see Part of Speech Tagging.

Prerequisites – What we need to remember

- Word order and surrounding words provide contextual evidence.
- Bag of Words does not explicitly encode grammatical roles.
- Conditional probabilities describe an event given its context.
  - Language models assign probabilities to token sequences, or estimate next token's probability, conditioned on preceding content.
- N-grams are sequences of words, e.g. bi-grams, trigrams (hot dog, New York).

Language – And its complex structure

- Texts have an inherently complex structure.
  - A simple BoW model is not enough for text understanding.
- Phrases can be more informative than single words.
  - Documents containing "*black sea*" differ from documents containing the separate words "*black*" and "*sea*".
- Phrases can be less ambiguous.
  - "*big apple*" provides more context than "*apple*".

Grammar Basics – Nouns, verbs, and modifiers

- A noun names an entity or concept, such as "*student*", "*book*", or "*knowledge*".
- A verb expresses an action, event, or state, such as "*read*", "*arrive*", or "*know*".
- An adjective describes a noun. In "a small book", "*small*" describes "*book*".
- An adverb modifies a verb, adjective, or another adverb.
  - In "*read quickly*", it describes how we read.
- We assign a category to the word occurrence in its sentence.

Part-of-Speech (POS) Tagging

Introduction

Question – Context and ambiguity

Consider the word “*can*” in these sentences.

They can swim.
They recycle a can.

Which part-of-speech tag does “*can*” receive in each sentence?
What helped you decide the role of “*can*” in each sentence?

POS Tagging – A First Example

EXAMPLE 1

I will book a flight.

VERB “book” functions as a **verb**.

EXAMPLE 2

I read a book.

NOUN “book” functions as a **noun**.

Key Takeaway: POS tagging assigns each token a grammatical category **given the context in which it appears**.

Grammatical Information – Why lexical features are insufficient

- The same word form can have several grammatical uses.
  - A word count alone cannot identify the intended use.
- POS tags make grammatical categories available to other systems.
  - For example, the word "*book*" when used as a noun it names an object. When used as a verb it expresses an action.
  - A tag is useful evidence, but it does not describe the whole meaning.

POS Foundations – Categories, context, and tagsets

- A POS category describes grammatical behavior in a sentence.
  - The available tags (e.g. VERB, NOUN) define the labels and annotation one can use.
- We use Universal Dependencies labels in the classroom examples.
  - Universal Dependencies labels are standardized tags used in NLP and linguistics to describe the grammatical and syntactic relationships between words in a sentence.
- Tags can have big and smaller categories.
  - Example: NOUN is a broad tag; a smaller sub-tag may also encode singular or plural.

POS Tags – Common lexical categories

Tag	Category	Examples
NOUN	Common noun	book, study, researcher
PROPN	Proper noun	Giulia, Milan, Bicocca
VERB	Main verb	book, study, read
ADJ	Adjective	small, useful, young
ADV	Adverb	quickly, very, often
INTJ	Interjection	hello, wow, oh

The grammatical use determines the category.

POS Tags – Understanding the categories

- NOUN names an entity or concept.
- PROPN names a particular person or place.
- VERB expresses an action, event, or state.
- ADJ describes a noun.
- ADV can describe how an action occurs.
- INTJ expresses a reaction.

POS Tags – Function words and other categories

Tag	Category	Example
DET	Determiner	the book
PRON	Pronoun	she reads
AUX	Auxiliary	can swim
ADP	Adposition	in Milan
CCONJ	Coordinating conjunction	books and papers
SCONJ	Subordinating conjunction	because it rains
PART	Particle	to swim
NUM	Numeral	two books
PUNCT	Punctuation	. , !
SYM	Symbol	%
X	Other	unclassified token

Together, these two tables contain all 17 universal POS tags.

POS Tags – Understanding the categories

- DET introduces a noun expression.
- PRON can stand for a noun expression.
- AUX helps express the predicate.
- ADP expresses a relation, such as “in Milan”.
- CCONJ links comparable elements.
- SCONJ introduces a subordinate clause.
- PART, NUM, PUNCT, SYM, and X cover particles, numbers, and other token types.

POS Foundations – Morphology and lemmas

books POS = NOUN Number = Plur Lemma = book

walked POS = VERB Tense = Past Lemma = walk

- Different annotation fields describe different aspects of the same token.
  - POS identifies the grammatical category.
  - Morphological features describe properties such as number and tense.
  - A lemma records a base form.

Exercise – Categories and ambiguity

The young researchers study language.

Their study is useful.

- Assign a POS tag to every token, including punctuation.
  - Explain the tag of "*study*" in each sentence.
  - Identify one contextual clue for each decision.

Tag	Category	Examples
NOUN	Common noun	book, study, researcher
PROPN	Proper noun	Giulia, Milan, Bicocca
VERB	Main verb	book, study, read
ADJ	Adjective	small, useful, young
ADV	Adverb	quickly, very, often
INTJ	Interjection	hello, wow, oh

Tag	Category	Example
DET	Determiner	the book
PRON	Pronoun	she reads
AUX	Auxiliary	can swim
ADP	Adposition	in Milan
CCONJ	Coordinating conjunction	books and papers
SCONJ	Subordinating conjunction	because it rains
PART	Particle	to swim
NUM	Numeral	two books
PUNCT	Punctuation	. , !
SYM	Symbol	%
X	Other	unclassified token

Exercise Answers – The two uses of “study”

Sentence	Token sequence	POS sequence
1	The / young / researchers / study / language / .	DET / ADJ / NOUN / VERB / NOUN / PUNCT
2	Their / study / is / useful / .	DET / NOUN / AUX / ADJ / PUNCT

- In sentence 1, researchers perform an action, so “*study*” receives VERB.
- In sentence 2, “*their study*” names something, so “*study*” receives NOUN.
- “*Is*” receives AUX (instead of VERB) in this copular construction under the UD convention. This is due to the fact that we have selected the Universal Dependencies (UD) annotation framework in our annotations.

Questions?

Comments?

Sequence Labeling

Sequence Labeling – One aligned label per token

Token	I	will	book	a	flight	.
POS	PRON	AUX	VERB	DET	NOUN	PUNCT

The input and output sequences have the same length.

- Each column pairs one token with one tag. The token order stays unchanged.
  - PRON means pronoun, AUX means auxiliary, and DET means determiner.
- "*Will*" supports the action. "*Book*" expresses the action. "*Flight*" names the object.

Sequence Labeling – Context and label dependencies

- A **tagger** receives an ordered token sequence $x_1, \ldots, x_n$.
  - It predicts an aligned sequence $y_1, \ldots, y_n$.
  - x means input tokens, y means output tags, and n means the number of tokens in a given context (e.g. a sentence).
- Input **context** can help predict each token's label.
  - Some models also score dependencies between output labels.

Questions?

Comments?

POS

Approaches we can use

Tagging Methods – The main approaches

Approach	Main evidence
Most frequent tag **baseline (Tag Frequency)**	Training counts for the word
Rules	Dictionary candidates and contextual conditions
Probabilistic sequence model	Scores for words and label sequences
Contextual neural tagger	Learned token representations from sentence context
Prompted language model	Task instructions, examples, and language knowledge

Tagging Methods – Understanding the approaches

- A frequency method **remembers** common labels.
- A rule method **tests** contextual conditions.
- A sequence model **predicts** a POS label for each word using the sentence context.
  - Some models also consider relationships between neighbouring labels to select the most likely label sequence.
- A neural tagger learns a mapping function from sentence representations to categories.
- A prompted model receives instructions and returns labels as text.

Learning from Examples – What a labeled dataset contains

- A dataset is a collection of examples used to develop or evaluate a system.
  - An input example may contain “I read a book.” split into tokens.
- Its reference labels are PRON, VERB, DET, NOUN, and PUNCT.
  - An annotator (i.e. a human) follows written guidelines to assign the reference labels.
- A supervised model learns to reproduce this input-to-label mapping on new text.

Example – The shared tagging problem

I will book a flight.

I read a book.

- Dictionary tag candidates for "*book*" include NOUN and VERB.
- The goal is to select the correct tag for a given occurrence, given its context.
- We will compare the context used by different approaches.

Approach 1 Tag Frequency – Most frequent tag step by step

Example
I will book a flight.

I read a book.

1. Count the training tags associated with each word.
   a. Suppose "*book*" appears 80 times as NOUN and 20 times as VERB, in our collection.
2. Store NOUN as the most frequent tag for "*book*".
3. Predict NOUN for "*book*" in all new sentences.

The baseline succeeds for "*a book*" and fails for "*will book*". Our simple tagger that relies on "how frequently is *book* used as a noun and verb" fails. But it's ok, because is just a baseline.

A baseline is a simple method used as a reference when evaluating more complex methods.

Approach 2 Rules – Contextual decisions step by step

Example
I will book a flight.

I read a book.

1. Look up the candidate tags for "*book*". Keep NOUN and VERB.
   a. In "*will book*", inspect the preceding auxiliary.
2. Apply a rule that favors a base-form VERB after a modal auxiliary. (If token_before == AUX then..)
3. In "*a book*", use the determiner as evidence favoring NOUN.
4. Select the remaining candidate or apply these fallback rules to get the annotations.

This approach is simple. But, how many rules should we create?

Rules – Why simple patterns can fail

a book DET NOUN

a small book DET ADJ NOUN

- A noun does not always immediately follow a determiner.
- A rule based only on the preceding word can miss useful evidence.
- Rules need exceptions, wider context, and tests on unseen sentences.

Probability Basics – Reading conditional probabilities

- Higher probabilities values indicate more likely events under the model.
  - P(N | V) reads as “the probability of noun after verb” in our transition model.
  - If 80 of 100 observed transitions from V lead to N, the simple probability estimate is 80 ÷ 100 = 0.8.
- A model combines several probabilities to compare a complete candidate sequence.
- The next example uses only two possible tag categories, N and V, so we can simplify our calculations.

Approach 3 HMM Tagging – Transitions and emissions

- A Hidden Markov Model treats tags as hidden states.
  - The input words are observations, i.e. what we can “see”. What exists.
  - The HMM’s assumption is that there are some hidden states that we do not observe, that are the reason why we observe certain words (i.e. these are the parts of speech tags).
  - Formally, hidden states are the underlying linguistic categories, such as Noun, Verb, Adjective, or Preposition, that you cannot directly observe in raw text.
  - HMM assumes that the hidden state (the tag) generates the observed word according to an emission probability, and that the sequence of hidden states moves from one to the next via a Markov chain.
- A transition probability scores a tag given the previous tag.
- An emission probability scores a word given its tag.
- The model compares complete candidate tag sequences.

HMM Tagging – A small probability model

Transition	To N (noun)	To V (verb)
START	0.6	0.4
N	0.3	0.7
V	0.8	0.2
Emission	**P(word | N)**	**P(word | V)**
book	0.10	0.20
flights	0.30	0.01
OTHER	0.60	0.79

Synthetic probabilities for the input "*book flights*".

- START denotes the beginning of the sentence.
  - A transition probability scores a tag given the previous tag.
  - The probability of transitioning from a tag V to a tag N is 0.8.
- An emission probability scores a word given its tag.
  - P(word | tag) means the probability of observing that word given the tag.
  - Given that the current POS tag is a noun, the probability of observing "*book*" as the current word is 0.10.
  - OTHER groups all remaining words, that we do not consider in this example.

HMM Tagging – Sequence scoring step by step

- What is the probability of the candidate tag of V N for "*book flights*":
  - Start with V. P(V | START) = 0.4
  - Emit "*book*". P(book | V) = 0.20
  - Move to N. P(N | V) = 0.8
  - Emit "*flights*". P(flights | N) = 0.30
- Joint probability of "*book flights*" = 0.4 × 0.20 × 0.8 × 0.30 = 0.0192
- What is the probability of the candidate tag of N N for the same phrase?

Transition	**To N (noun)**	**To V (verb)**
START	0.6	0.4
N	0.3	0.7
V	0.8	0.2
Emission	**P(word | N)**	**P(word | V)**
book	0.10	0.20
flights	0.30	0.01
OTHER	0.60	0.79

HMM Tagging – Comparing the candidates

Candidate tags	Joint score
N N	0.6 × 0.10 × 0.3 × 0.30 = 0.0054
N V	0.6 × 0.10 × 0.7 × 0.01 = 0.00042
V N	0.4 × 0.20 × 0.8 × 0.30 = 0.0192
V V	0.4 × 0.20 × 0.2 × 0.01 = 0.00016

The model selects V N for “book flights”.

HMM Tagging – Understanding the example

- Two words and two possible tags produce four candidates: N N, N V, V N, and V V.
- Each score multiplies the start, word, transition, and second-word probabilities.
- V N wins because 0.0192 is larger than the other three scores.

Sequence Decoding – Why choosing greedily can fail

- Decoding means selecting output labels from the model's scores.
- Our HMM gives N the larger start probability, but the winning complete path starts with V.
- Later evidence can change which earlier choice belongs to the best complete sequence.
- A sequence decoder retains alternatives until it can compare the complete paths.

Sequence Decoding – The idea behind Viterbi

- Checking every complete label sequence quickly becomes expensive.
  - With 2 possible tags and 10 tokens, there are 2^{10} = 1,024 candidates.
- Viterbi keeps the best partial path ending in each possible tag.
  - At the next word, it compares ways to extend each retained path.
  - It stores the previous choice for each retained path. This record is a backpointer.
  - Move word by word through the sentence. For each tag, calculate the product of the previous path probability, the tag transition probability, and the word emission probability, keep only the maximum score for each state.
- At the end, it follows these records backward to recover the best complete sequence.

Approach 4 Neural Model Basics – Representations and classification

- A neural encoder learns to produce useful (token) representations from the input sentence.
- A classifier converts each representation into a score for every allowed POS tag.
- Labeled examples teach the classifier which tag should receive the highest score.
- Pre training provides a starting point. POS training adapts the model to this task.

Neural Tagging – Contextual prediction step by step

1. Tokenize the sentence for a pretrained encoder such as BERT.
2. Encode each token using the surrounding sentence (context/attention mechanism).
3. Pass each representation to a trained POS classification layer.
4. Convert label scores into predictions.

An encoder converts tokens into numerical representations. A neural classifier maps them to tags.

Neural Tagging – The two representations of "*book*"

I will book a flight.

The encoder represents "*book*" using this sentence context, into an embedding representation.

The POS neural classification layer should assign VERB.

I read a book.

The encoder represents "*book*" using the different context. (embedding representation)

The POS neural classification layer should assign NOUN.

Approach 4 LLM Tagging – A concrete prompt and expected output

- LLM means large language model.
  - We can give it a tagging task through written instructions (prompts).
- Supply the fixed tokens [“I”, “will”, “book”, “a”, “flight”, “.”].
- Instruct the model to use Universal Dependencies tags and preserve token order.
- Request exactly six labels and no added tokens.
- The expected labels are [PRON, AUX, VERB, DET, NOUN, PUNCT].
- Check both output alignment and grammatical correctness **before using the result**.

LLM Tagging – Prompting and output checks

1. Provide the sentence, tagset, and annotation instructions.
2. Optionally include a few tagged examples.
3. Request one output label for each supplied token.
4. Check token alignment and whether every label belongs to the tagset.
5. Evaluate the predictions on reference annotations.

A valid output format does not guarantee correct grammatical labels.

Method Comparison – Evidence and practical tradeoffs

Method	Main strength	Main limitation
Frequency baseline	Simple comparison point	Ignores context
Rules	Readable decisions	Coverage and maintenance
HMM / CRF	Explicit sequence modeling	Assumptions and data sparsity
Contextual neural tagger	Learns sentence features	Domain shift and compute
Prompted LLM	Can use task instructions	**Cost** and output consistency

Evaluate the method on the target text.

POS
Systems, Libraries, & Applications

Real Systems – Documented uses of POS

Systems/Libraries	What it exposes or uses
LanguageTool	POS tags in grammatical error patterns
spaCy	Token annotations, POS-dependent rule lemmatization
Stanford CoreNLP	A POS annotator in its NLP pipeline
Stanza	Universal and treebank-specific POS annotations

These systems document explicit POS tagging.

Applications in 2026 – Research examples

- Historical texts require annotation despite limited labeled data.
  - Research has investigated POS tagging and lemmatization for Ancient Greek, Armenian, Georgian, and Syriac.
  - Other studies investigate tagging methods for Medieval Occitan, Catalan, and French.

Main Takeaways

01 POS tagging describes grammatical categories in context.

02 Sequence labeling predicts an aligned (1-to-1) output label sequence.

03 Rules and learned models use different sources of evidence.

04 Contextual input representations do not guarantee correctness in POS tags, or consistency.

05 Useful predictions require consistent annotation and evaluation.

Contacts

Dr. Georgios Peikos

georgios.peikos@unimib.it

Academic Year 2026/27

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Ikr3 Lab information and knowledge representation, retrieval, and reasoning LAB