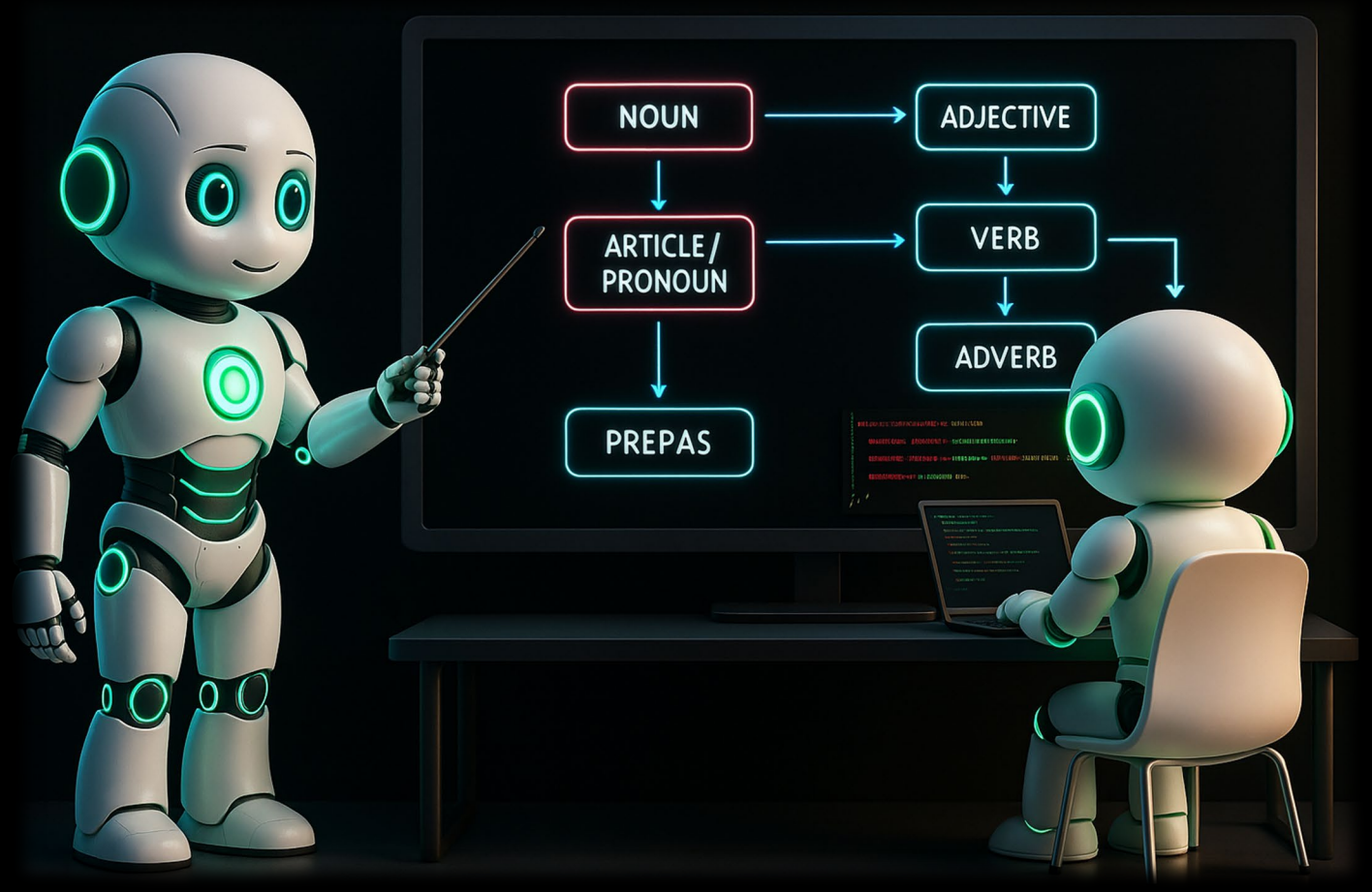
Part-of-Speech Tagging.

Teaching Computers Grammar

So, They Don't Sound Like Robots!!

ITAI 2373 - Mod 05

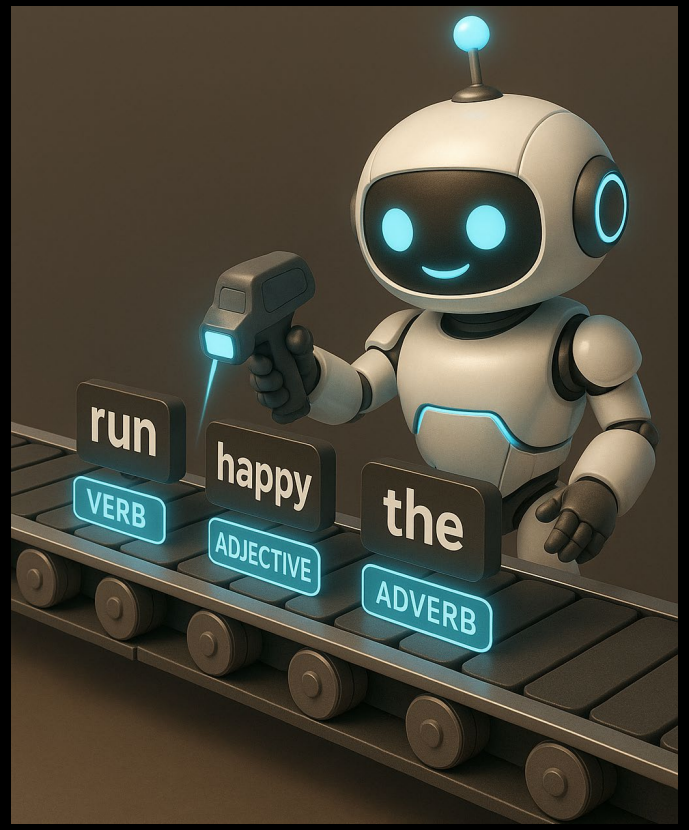


Learning Objectives

By the end of this module, you will be able to:

- Understand why POS tagging is the "grammar police" of NLP
- Navigate different POS tag sets like choosing between Netflix and Hulu
- Explain the evolution from rule-based to statistical approaches (spoiler: data wins)
- Identify the challenges that make POS tagging trickier than your high school English class
- Understand how human annotation creates the "training data" that makes everything possible
- Appreciate how POS tagging powers the apps you use every day

What Even IS Part-of-Speech Tagging?



- **POS Tagging**: Teaching computers to identify if a word is a noun, verb, adjective, etc.
- **Why Care?** It's like giving every word in a sentence an ID badge
- **Real Talk**: Without this, AI would be as confused by language as you are by IKEA instructions

Why POS Tagging is Actually Your Daily Superhero

Search Engines: "Apple stock" vs "apple pie" - different vibes entirely

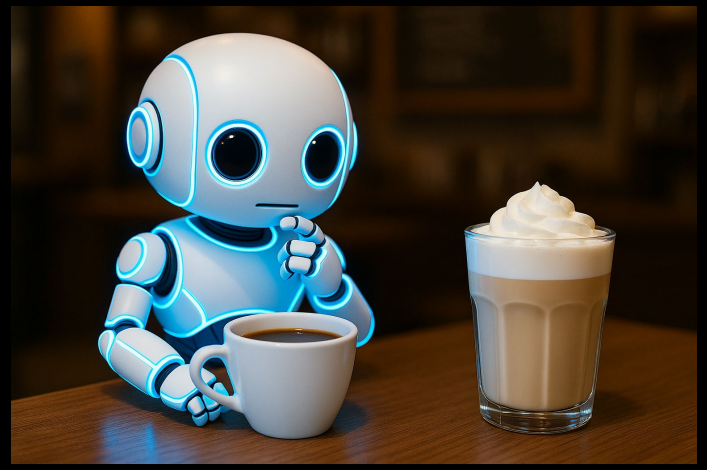
Voice Assistants: When Siri actually understands what you meant

Translation Apps: Why Google Translate doesn't think you're talking about fruit when you mention Apple Inc.

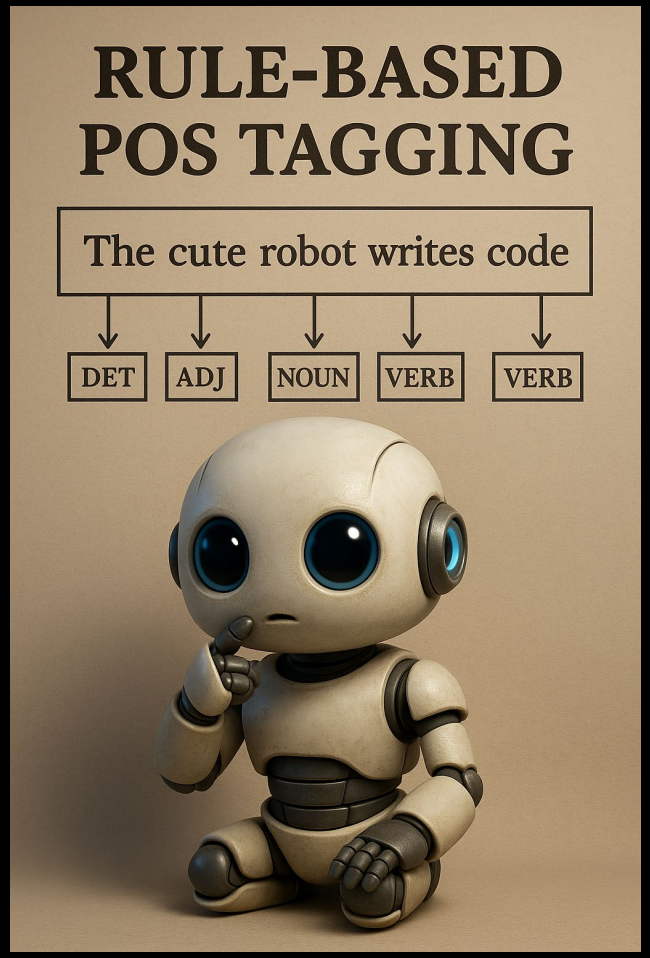
Autocorrect: Sometimes knows you meant "duck" not... well, you know

The Tag Set Menu - Choose Your Fighter

- **Penn Treebank:** The detailed menu (~45 tags) - "NN, NNS, NNP, NNPS“
- **Universal Dependencies:** The simplified menu (~17 tags) - "Just NOUN, please“
- **Why Different Sets?:** Like choosing between a coffee black or 'large dark roast with oat milk



The Old School Approach - Rule-Based Tagging



- **The Concept:** Write explicit rules like "If word ends in -ly, it's probably an adverb“
- **Example:** "If it's -ing and follows 'is,' tag it as VBG (gerund)“
- **Pros:** Makes sense to humans, works great for covered cases
- **Cons:** Language laughs at your rules and breaks them constantly

The Plot Twist - Statistical Methods Save the Day

The Revelation: Let computers learn patterns from massive amounts of data

How It Works: "In our training data, 'bank' was a noun 847 times and a verb 23 times“

Famous Models: Hidden Markov Models (HMMs), Maximum Entropy

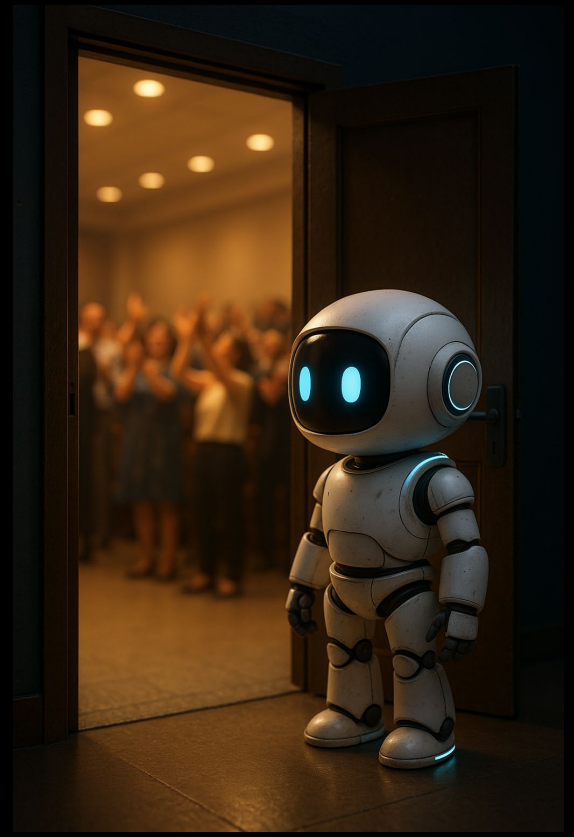
The Upgrade: Handles ambiguity like a pro, learns from mistakes

When POS Tagging Gets Complicated (Plot Twists Everywhere)

- **Identity Crisis:** "lead" (metal) vs "lead" (guide) vs "lead" (past tense of "lead")
- **New Kids on the Block:** Words the tagger has never seen before
- **Corporate Confusion:** "Apple" (fruit) vs "Apple" (your expensive habit)
- **Phrasal Verbs:** "look up" (research) vs "look" + "up" (direction)

The Unsung Heroes - Human Annotators

- **The Job:** Humans manually tag thousands of sentences with correct POS labels
- **Why It Matters:** Creates the "answer key" that computers learn from
- **The Reality:** Someone actually sat there and tagged "The quick brown fox...“
- **Fun Fact:** Your Netflix recommendations are only as good as the human ratings they learned from



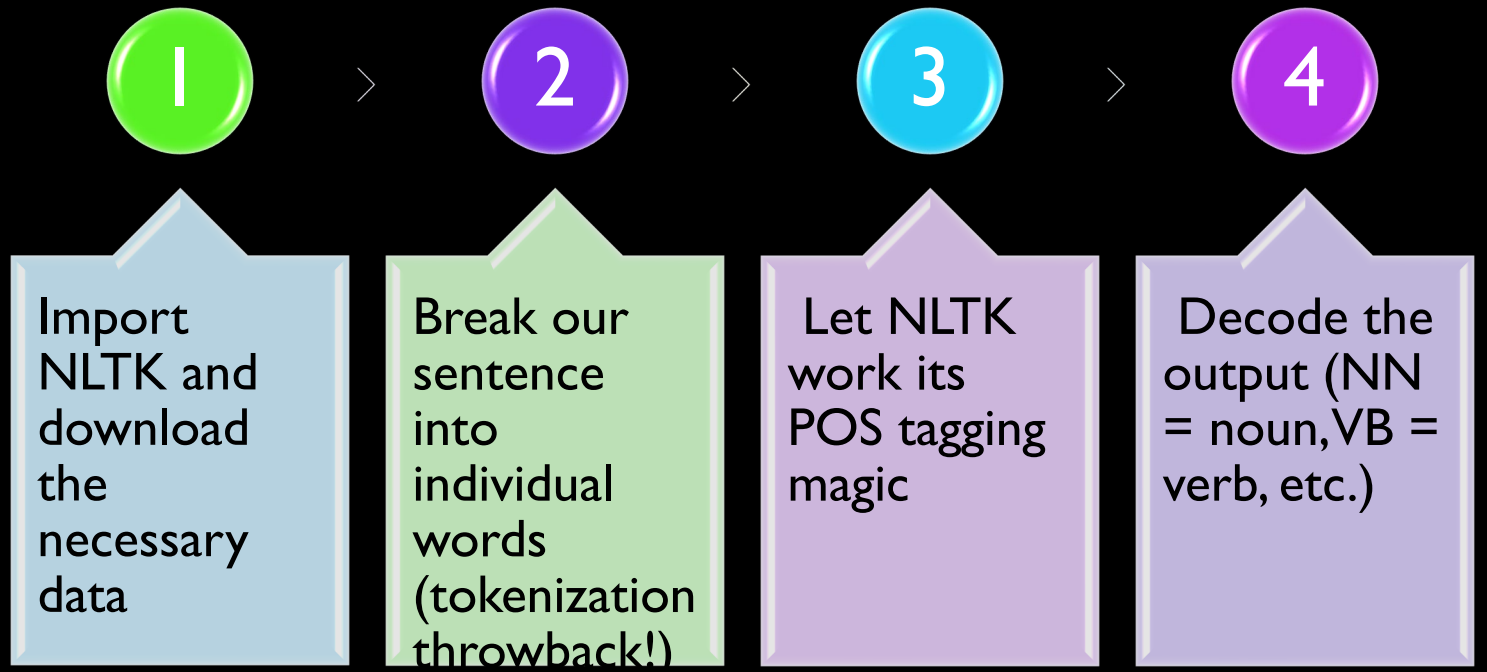
The Evolution Continues - From Statistics to AI Superpowers

- **The Journey:** Rules → Statistics → Neural Networks → Large Language Models
- **Neural Networks:** Like giving computers a really good intuition about language
- **LLMs** (ChatGPT & Friends): So good at language they do POS tagging without even trying
- **Plot Twist:** Modern AI is so advanced it makes POS tagging look easy

Getting Our Hands Dirty - NLTK vs SpaCy Showdown

- **NLTK:** The comprehensive toolkit - like a Swiss Army knife for NLP
- **SpaCy:** The speed demon - built for getting stuff done fast
- **The Choice:** Academic exploration vs production deployment
- **Reality Check:** Both are awesome, just for different reasons

Hands-On Time - Making NLTK Do the Grammar Police Thing



Let's make this computer show off its grammar skills!"

Code Example

```python
import nltk
# Download required data (run these once)
# nltk.download('punkt')
# nltk.download('averaged_perceptron_tagger')

# Our test sentence - a classic!
sentence = "The quick brown fox jumps over the lazy dog."

# Step 1: Tokenize (break into words)
tokens = nltk.word_tokenize(sentence)
print("Tokens:", tokens)

# Step 2: POS tag those words
pos_tags = nltk.pos_tag(tokens)
print("POS Tags:", pos_tags)
```

- **Expected Output Explanation:** "You should see something like: [('The', 'DT'), ('quick', 'JJ'), ('brown', 'JJ'), ('fox', 'NN'), ('jumps', 'VBZ'), ('over', 'IN'), ('the', 'DT'), ('lazy', 'JJ'), ('dog', 'NN'), ('.', '.')]

- Translation: DT = Determiner (like 'the'), JJ = Adjective (like 'quick', 'brown', 'lazy'), NN = Noun (like 'fox', 'dog'), VBZ = Verb 3rd person singular present (like 'jumps'), IN = Preposition (like 'over'), and even punctuation gets tagged! Pretty cool, right?"

SpaCy's Turn - The Speed Demon Shows Off

Step 1: Load SpaCy's pre-trained English model

Step 2: Process the entire sentence in one go

Step 3: Extract POS tags from the processed document

Step 4: Notice the cleaner, more universal tag names

Real-World Drama - Customer Service Call Analysis

- **The Mission**: Analyze thousands of customer service call transcripts
- **The Challenge**: People don't talk like textbooks (shocking, I know)
- **The Goal**: Extract customer intent, problems, and sentiment from messy speech
- **The Hero**: POS tagging helps cut through the linguistic chaos

When Speech Recognition Meets Reality (Spoiler: It's Messy)

- **Disfluencies**: "Um, I, uh, need to, like, reset my password"
- **Accent Adventures**: ASR might transcribe "about" as "aboot"
- **Casual Chaos**: "gonna," "ain't," "y'all" - grammar rules? What grammar rules?
- **Punctuation Problems**: ASR output is often one long run-on sentence

Fighting Back - Making POS Tagging Tougher

- **Domain Training:** Teach taggers using actual spoken language data
- **Robust Models:** Build taggers that don't panic when things get weird
- **Clean-Up Crew:** Post-processing to fix common speech-to-text errors
- **Context is King:** Use surrounding words to solve ambiguities

Wrapping Up - You're Now POS Tagging Pros!

- **POS Tagging Matters:** You now understand how computers identify parts of speech in text—a foundational skill in NLP.
- **Tools at Your Fingertips:** You can confidently use tools like NLTK and spaCy to tag words in real-world text.
- **More Than It Seems:** You recognize that POS tagging isn't just labeling—it involves ambiguity, context, and complexity.
- **You're Ready:** You're prepared to take the next step: building applications that can interpret and use language structure effectively.