**Notes on Morphology and Part of Speech Tagging**

**1. Linguistic Essentials**

Understanding the fundamental aspects of linguistics is crucial for Natural Language Processing (NLP).

It includes:

* **Lexical Analysis:** Study of words and their meanings.
* **Morphology:** Structure and formation of words.
* **Syntax:** Arrangement of words and phrases to create well-formed sentences.

**2. Morphology and Finite State Transducers (FST)**

* **Morphology** deals with the structure of words.
* **Finite State Transducers (FST)**:
  + A finite-state machine that maps input sequences to output sequences.
  + Used in **morphological analysis**, **stemming**, and **spell checking**.

**3. Part of Speech (POS) Tagging**

* Assigns grammatical labels (noun, verb, adjective, etc.) to words.
* Used in **speech recognition, syntactic parsing, and information retrieval**.
* **Tagsets**: Predefined sets of POS tags, e.g., Penn Treebank Tagset.

**Types of POS Tagging Approaches**

1. **Rule-Based POS Tagging**
   * Uses linguistic rules to assign tags.
   * Example: If a word ends with "-ing", it is likely a verb.
2. **Markov Models**
   * Based on probability distributions of word sequences.
   * **Hidden Markov Models (HMMs)**: Uses observed words and hidden states (POS tags) to determine the most probable sequence.
3. **Transformation-Based Models**
   * Learns a set of transformation rules to improve initial tag assignments.
   * Example: Brill Tagger.
4. **Maximum Entropy Models**
   * Uses probability distributions without making strong independence assumptions.
5. **Conditional Random Fields (CRFs)**
   * A probabilistic graphical model for sequence labeling.
   * More powerful than HMMs since it considers past and future context simultaneously.

**4. Syntax Parsing**

* **Syntax Parsing**: Determining the syntactic structure of a sentence.
* **Grammar Formalisms**: Set of rules defining valid sentence structures.
* **Treebanks**: Annotated datasets containing syntactic structures of sentences.

**Types of Parsing**

1. **Parsing with Context-Free Grammars (CFGs)**
   * A formal grammar where each rule maps a single non-terminal symbol to a sequence of symbols.
   * Example: S → NP VP (A sentence consists of a noun phrase and a verb phrase).
2. **Features and Unification**
   * Enriches CFGs by adding constraints.
   * Example: Ensuring subject-verb agreement.
3. **Statistical Parsing and Probabilistic CFGs (PCFGs)**
   * Assigns probabilities to grammar rules based on corpus data.
   * Helps determine the most likely parse tree for a sentence.
4. **Lexicalized PCFGs**
   * Extends PCFGs by incorporating word-specific probabilities.
   * Improves parsing accuracy for languages with rich morphology.

**Questions and Answers**

**Q1: What is morphology in NLP?**

**A:** Morphology is the study of word formation and structure, including root words, affixes, and word derivation.

**Q2: What is the purpose of Part of Speech (POS) tagging?**

**A:** POS tagging assigns grammatical categories (e.g., noun, verb) to words in a sentence, aiding in syntactic and semantic analysis.

**Q3: How do Rule-Based POS taggers work?**

**A:** They use manually defined linguistic rules to determine POS tags based on word structure and context.

**Q4: What is the difference between Hidden Markov Models (HMM) and Conditional Random Fields (CRF)?**

**A:**

* **HMMs** assume each state (POS tag) depends only on the previous state.
* **CRFs** consider both past and future context, making them more accurate for sequence labeling.

**Q5: What is syntax parsing?**

**A:** Syntax parsing analyzes a sentence's structure to determine relationships between words based on grammatical rules.

**Q6: What are Context-Free Grammars (CFGs)?**

**A:** CFGs are a type of formal grammar where each rule replaces a single non-terminal symbol with a sequence of symbols.

**Q7: What are treebanks, and why are they important?**

**A:** Treebanks are annotated corpora containing sentence structures. They help train and evaluate parsing models.

**Q8: What is a Probabilistic Context-Free Grammar (PCFG)?**

**A:** PCFG assigns probabilities to CFG rules based on observed frequencies in a corpus, allowing for statistically informed parsing.

**Q9: How do lexicalized PCFGs improve parsing?**

**A:** Lexicalized PCFGs incorporate word-specific information, improving accuracy for languages with complex morphology.

**Q10: What is the main advantage of using Conditional Random Fields (CRFs) over HMMs?**

**A:** CRFs consider both past and future context, avoiding independence assumptions that limit HMM performance.

**Solved Examples for POS Tagging, Parsing, and Morphological Analysis**

**Example 1: Morphological Analysis Using Finite State Transducers (FST)**

**Input Sentence:**

*"He is running quickly."*

**Morphological Breakdown:**

| **Word** | **Root** | **Affix** | **Morphological Feature** |
| --- | --- | --- | --- |
| He | He | - | Pronoun |
| is | be | -s | Verb (3rd person singular present) |
| running | run | -ing | Verb (present participle) |
| quickly | quick | -ly | Adverb |

**Explanation:**

* The verb "running" is derived from "run" + "-ing" (present participle).
* "Quickly" is derived from "quick" + "-ly" (adverbial suffix).

**Example 2: Part of Speech (POS) Tagging**

**Input Sentence:**

*"The cat sat on the mat."*

**Rule-Based POS Tagging:**

| **Word** | **POS Tag** | **Rule Used** |
| --- | --- | --- |
| The | DT (Determiner) | Article rule |
| cat | NN (Noun) | Common noun rule |
| sat | VBD (Verb, past tense) | Past tense verb rule |
| on | IN (Preposition) | Preposition rule |
| the | DT (Determiner) | Article rule |
| mat | NN (Noun) | Common noun rule |

**HMM-Based POS Tagging:**

Using a **Hidden Markov Model (HMM)**, probabilities are assigned based on corpus data:

* P(NN | DT) = High (since a noun follows a determiner often)
* P(VBD | NN) = High (since a verb follows a noun frequently)
* The most probable POS sequence is:  
  **DT → NN → VBD → IN → DT → NN**

**Example 3: Syntax Parsing Using Context-Free Grammar (CFG)**

**Input Sentence:**

*"The boy eats an apple."*

**Grammar Rules:**

1. **Sentence (S) → Noun Phrase (NP) + Verb Phrase (VP)**
2. **Noun Phrase (NP) → Determiner (DT) + Noun (NN)**
3. **Verb Phrase (VP) → Verb (V) + Noun Phrase (NP)**
4. **Determiner (DT) → {the, an}**
5. **Noun (NN) → {boy, apple}**
6. **Verb (V) → {eats}**

**Parse Tree:**

S

/ \

NP VP

/ \ / \

DT NN V NP

| | | / \

The boy eats DT NN

| |

an apple

**Explanation:**

* "The boy" is a **noun phrase** (DT + NN).
* "eats an apple" is a **verb phrase** (V + NP).
* The sentence follows **CFG parsing** rules.

**Example 4: Probabilistic Context-Free Grammar (PCFG)**

**Sentence:**

*"She reads a book."*

**PCFG Rule Probabilities (based on corpus data):**

1. **S → NP VP (0.9)**
2. **NP → DT NN (0.8)**
3. **VP → V NP (0.85)**
4. **V → reads (0.7)**
5. **DT → a (0.6)**
6. **NN → book (0.75)**

**Most Probable Parse Tree Calculation:**

P(S → NP VP) \* P(NP → DT NN) \* P(VP → V NP) \* P(V → reads) \* P(DT → a) \* P(NN → book)

= **0.9 × 0.8 × 0.85 × 0.7 × 0.6 × 0.75**

= **0.214** (Most probable structure)

**Example 5: Lexicalized PCFG Parsing**

**Input Sentence:**

*"Dogs bark loudly."*

**Lexicalized PCFG Rules:**

1. **S → NP(dogs) VP(bark) [0.95]**
2. **VP(bark) → V(bark) Adv(loudly) [0.7]**
3. **NP(dogs) → NN(dogs) [0.85]**

**Most Likely Parse Tree:**

S

/ \

NP VP

| / \

NN V Adv

| | |

Dogs bark loudly

**Explanation:**

* This tree **incorporates lexical items** (words themselves) instead of just generic POS tags, making it more context-aware.

**Example 6: POS Tagging Using Conditional Random Fields (CRFs)**

**Input Sentence:**

*"She bought a new dress."*

**Feature-Based Tagging Using CRF:**

| **Word** | **Previous Word** | **Next Word** | **POS** |
| --- | --- | --- | --- |
| She | \_ (start) | bought | PRP |
| bought | She | a | VBD |
| a | Bought | new | DT |
| new | A | dress | JJ |
| dress | New | \_ (end) | NN |

**CRF Explanation:**

* The CRF model **considers both past and future context**.
* Unlike HMMs, it avoids independence assumptions.

**Example 7: Morphological Analysis with Inflectional and Derivational Morphology**

**Input Sentence:**

*"The children are playing happily."*

**Morphological Breakdown:**

| **Word** | **Root** | **Affix** | **Morphological Feature** | **Type of Morphology** |
| --- | --- | --- | --- | --- |
| The | the | - | Determiner | - |
| children | child | -ren | Plural noun | Inflectional |
| are | be | - | Auxiliary verb | - |
| playing | play | -ing | Present participle verb | Inflectional |
| happily | happy | -ly | Adverb | Derivational |

**Explanation:**

* "Children" is an **inflectional** change (plural form of "child").
* "Happily" is a **derivational** change (adjective "happy" → adverb "happily").

**Example 8: Finite State Transducer (FST) for Morphological Analysis**

**Input Word: "cats"**

**FST Representation:**

* **State 1:** "cat" (Root)
* **State 2:** "s" (Plural marker)
* **Output:** ["cat", "+PLURAL"]

**Explanation:**

* The **finite state transducer** recognizes "cats" as "cat" + "s" and identifies it as a **plural noun**.

**Example 9: Rule-Based Part of Speech (POS) Tagging**

**Input Sentence:**

*"John quickly runs home."*

**Manually Applied Rules:**

1. If a word starts with a **capital letter** and is a **known name**, it's a **proper noun** (NNP).
2. If a word **ends in -ly**, it’s usually an **adverb** (RB).
3. If a word is **known to be a verb in the dictionary**, it’s a **verb** (VB).
4. If a word is found in a **list of known places**, it’s a **noun** (NN).

**Tagged Output:**

| **Word** | **Rule Applied** | **POS Tag** |
| --- | --- | --- |
| John | Capitalized proper noun | NNP |
| quickly | Ends in "-ly" | RB |
| runs | Verb | VBZ |
| home | Known place | NN |

**Example 10: Hidden Markov Model (HMM) POS Tagging**

**Input Sentence:**

*"He can fish."*

**Ambiguity:**

* "can" could be a **modal verb** (MD) or a **noun** (NN).
* "fish" could be a **noun** (NN) or a **verb** (VB).

**Probabilities from a Trained HMM:**

1. P(MD | PRP) = 0.8 (Modal verb follows a pronoun)
2. P(VB | MD) = 0.9 (Verb follows modal)
3. P(NN | VB) = 0.7 (Noun follows verb)

**Most Probable Sequence (Viterbi Algorithm Applied):**

PRP (He) → MD (can) → VB (fish)

**Final Tagged Output:**

| **Word** | **POS Tag** |
| --- | --- |
| He | PRP |
| can | MD |
| fish | VB |

**Example 11: Transformation-Based Tagging (Brill Tagger)**

**Input Sentence:**

*"The dog barked loudly."*

**Initial POS Tags (Baseline Tagger):**

| **Word** | **Initial POS Tag** |
| --- | --- |
| The | NN (Incorrect) |
| dog | NN |
| barked | NN (Incorrect) |
| loudly | RB |

**Transformation Rules Applied:**

1. If a **word follows "The"**, it’s likely a **noun (NN)**.
2. If a **word ends in -ed** and follows a **noun**, it's likely a **verb (VBD)**.

**Corrected POS Tags After Transformation:**

| **Word** | **Corrected POS Tag** |
| --- | --- |
| The | DT |
| dog | NN |
| barked | VBD |
| loudly | RB |

**Example 12: Conditional Random Fields (CRFs) for POS Tagging**

**Input Sentence:**

*"She is watching a movie."*

**Features Used in CRFs:**

| **Word** | **Previous Word** | **Next Word** | **POS Tag** |
| --- | --- | --- | --- |
| She | - | is | PRP |
| is | She | watching | VBZ |
| watching | Is | a | VBG |
| a | Watching | movie | DT |
| movie | A | - | NN |

**CRF Explanation:**

* Unlike HMM, CRF **considers the entire context** when predicting POS tags.

**Example 13: Parsing with Context-Free Grammar (CFG)**

**Input Sentence:**

*"The bird sings."*

**CFG Rules:**

1. **S → NP VP**
2. **NP → DT NN**
3. **VP → V**

**Parse Tree:**

S

/ \

NP VP

/ \ |

DT NN V

| | |

The bird sings

**Example 14: Statistical Parsing with Probabilistic Context-Free Grammars (PCFG)**

**Input Sentence:**

*"She saw a man with a telescope."*

**Ambiguity:**

1. **Did she use a telescope to see the man?**
   * **(She saw) (a man with a telescope).**
2. **Did she see a man who was holding a telescope?**
   * **(She saw a man) (with a telescope).**

**PCFG Rule Probabilities:**

1. VP → V NP [0.8]
2. PP → P NP [0.5]
3. NP → NP PP [0.6]

**Disambiguation Using Probabilities:**

* The **higher probability parse tree** is chosen, depending on corpus training.

**Example 15: Lexicalized PCFG**

**Input Sentence:**

*"John eats an apple."*

**Lexicalized Rules:**

1. S → NP(John) VP(eats)
2. VP(eats) → V(eats) NP(apple)
3. NP(apple) → DT(an) NN(apple)

**Lexicalized Parse Tree:**

S

/ \

NP VP

| / \

NNP V NP

| | / \

John eats DT NN

| |

an apple