Regular Expressions and Pattern Matching

Dr. Sambit Praharaj

Assistant Professor (II)

Pattern Matching

- Process of finding patterns in text
- Exact match, wildcard, regex
- Example: Find all emails in a document

What is Regular Expression (Regex)?

- Formal language for defining text patterns
- Used for search, validation, extraction
- Example: \d{10} → 10-digit number

Literal Matching

- Pattern: cat
- Matches: cat, concatenate
- Does NOT match: cut

Character Classes [ ]

- Pattern: [aeiou]
- Matches: any vowel

- Pattern: gr[ae]y
- Matches: gray, grey

Wildcard .

- Pattern: c.t
- Matches: cat, cut, cot
- Meaning: . → any single character

Quantifiers

- * → zero or more
- + → one or more
- ? → optional

- Example:
- go+ → go, goo
- go* → g, go

Exact Repetition {m,n}

- Pattern: \d{3}
- Matches: 123

- Pattern: \d{2,4}
- Matches: 12, 123, 1234

Anchors ^ and $

- ^hello → starts with hello
- world$ → ends with world

- Example:
- ^\d{10}$ → exactly 10 digits

Grouping ( ) and OR |

- Pattern: (ha)+
- Matches: ha, haha, hahaha

- Pattern: cat|dog
- Matches: cat or dog

Predefined Character Sets

- \d → digit (0–9)
- \w → word character
- \s → whitespace

- Example: \w+ → words

Email Regex Example

- [a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[A-Za-z]{2,}

- Matches valid emails

Phone Number Example

- \d{10} → any 10-digit number
- (+91)?[6-9]\d{9} → Indian mobile

Regex in NLP Preprocessing

- Remove punctuation
- Normalize spaces
- Validate tokens
- Extract entities

Pattern Matching vs Regex

- Pattern Matching: general concept
- Regex: powerful tool
- Regex $\subset$ Pattern Matching

Summary

- Regex defines text patterns
- Widely used in NLP preprocessing
- Essential skill for text analytics