15-06-2023

**INTRODUCTION TO NLP- UNIVERSITY OF LONDON**

1. **Introducing NLP: Patterns and structure in language**

* **Tokenization** involves breaking a text into individual words, punctuation marks, and other discrete items.
* It mentions several ways to identify personal names within tokenized text, such as using titles, capitalization followed by a comma and a number, or capitalization followed by a verb that applies to humans.
* The concept of **parts of speech** includes categories such as nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions, and determiners.
* Assigning a part of speech to a word depends on its function in context.
* The notion of **constituent structure** is highlighting patterns and phrasal categories that can occur in similar contexts.
* **Production rules** are a way to represent constituent structure.
* Texts can be analysed as hierarchical structures, and it encompasses two formal techniques for modeling grammatical knowledge: word chain devices (finite state machines) and phrase structure grammars, which include recursion.

1. **Finite State Machines (FSM):**

A finite-state machine can be thought of as a device that consists of lists of symbols and rules for transitioning between the lists. It can be used to either accept or generate strings.

* The passage begins by presenting a simple example of a wordchain device or FSM. It consists of word lists and rules.
* The FSM can be operated by reading a string symbol by symbol and determining whether each symbol is found in the current list.
* If a symbol is found, the machine advances to the next state and reads the next symbol.
* Alternatively, the FSM can be used to generate strings by picking one word from each list in sequence.
* The passage then presents a more complex example that involves sequence, selection, and iteration operations.
* It demonstrates how to represent FSMs using different notations, such as graphical representation (nodes and arcs) and tabular form (state-transition table).

1. **Declarative alternative to finite-state machines:**
2. **Regular expressions (REs):**

These formalisms allow for the separation of language specification from processing strategies. Regular expressions (REs) are used to identify patterns in text and are widely used in computer science applications.

They are based on three fundamental concepts:

* 1. **Sequence:** Specifies the order in which items occur. It can include a wildcard character (written as a period or full stop '.') that matches any character.
  2. **Selection:** Specifies a choice between alternative items or sequences using the '|' operator.
  3. **Iteration:** Specifies repetition of items or sequences using the '\*' operator, indicating zero or more occurrences of the preceding element.

**Additional operators** extend the basic regular expression syntax in programming languages like Java, Perl, and Python.

These operators include '+', indicating one or more occurrences of the previous item; '?', indicating optional occurrence of the previous item; and character ranges like '[A-Z]' and '[0-9]'.

Regular expressions can be used to recognize patterns like personal names.

1. **Regular grammars** are another declarative forma to FSM.

* Regular grammars consist of production rules or rewrite rules.
* The rules specify how non-terminal symbols can be replaced by sequences of terminal or non-terminal symbols.
* Regular grammars have certain restrictions on the placement of non-terminal symbols in the production rules, resulting in right-linear or left-linear grammars.

1. **Tree diagrams** are used to represent the structure of sentences generated by regular grammars. Each node in the tree represents a symbol, and the edges show how the symbols are related through production rules.
2. **Word Structure:**

* In the context of linguistics and natural language processing, word structure refers to how words are formed and how their forms can change to convey different meanings and grammatical functions.
* In English, words can have different endings or internal changes to indicate various grammatical features.
* For example, nouns can have different endings to show singular or plural forms, like "box" (singular) and "boxes" (plural). In some languages, this information may be indicated at the beginning of the word instead.
* For instance, in Swahili, "ziwa" means "lake" while "maziwa" means "lakes."
* Verbs in English also undergo changes in their endings to indicate person, number, tense, and mood.
* However, different languages may make different distinctions in verb forms.
* Some verbs are classified as regular or irregular based on whether their inflected forms can be derived by following simple rules.
* Regular verbs follow predictable patterns, while irregular verbs have unique or irregular forms.
* The field of linguistics known as morphology focuses on studying the structure of words and formulating rules for deriving different word forms based on their grammatical roles.
* Morphological rules help in understanding how words change and how their various forms can be derived.
* In natural language applications like machine translation systems, a lexicon or word database is often used, along with rules for deriving word endings.
* When translating a word from one language to another, the system may follow a process called stemming, which involves removing affixes or endings from words to derive their basic form or stem.
* The stem can then be used to find the appropriate translation or apply further rules for inflection in the target language.

1. **Summary:**
2. Some fundamental operations in text analysis include tokenisation, which involves extracting these meaningful elements from a stream of electronic characters and discarding white space, line feed characters and other material which has no explicit meaning for a human reader, and using regular expressions to identify patterns in a text.
3. Regular expressions are composed of the three basic operations of sequence, selection and iteration, and have many applications in computational linguistics and computer science at large. A finite-state machine is a process whose operations can be specified by means of regular expressions. A regular grammar is a set of production rules or rewrite rules that defines the sentences that make up a language, and any language defined by a regular grammar can be processed by a finite state machine or described using a regular expression.
4. A complete syntactic analysis of natural language sentences is generally held to require the additional operation of centre-embedded recursion, which is beyond the power of finite-state methods. Recursion is formally encoded in context-free grammars.
5. Not only do words combine in various patterns and structures to form sentences; they also have internal structure which can be described to an extent using rules for regular and irregular forms