Contents

Chapter 1

HTML

HTML, an abbreviation for Hypertext Markup Language, is the standard language for creating web pages and web applications. It is a cornerstone technology of the World Wide Web and forms the structure and layout of web content.

1.1 HTML Elements

An HTML document is composed of a series of elements, which are denoted by tags. Elements have an opening tag and a closing tag with content in between. Some elements, however, are self-closing and do not contain any content. For example, the paragraph tag ‘<p>‘ is used to denote a paragraph:

```
<p>This is a paragraph.</p>
```

1.2 HTML Document Structure

A typical HTML document has a specific structure, including the following elements:

- **DOCTYPE declaration**: This informs the browser about the version of HTML. For HTML5, it is ‘<!DOCTYPE html>‘.
- **html**: This tag encloses the entire HTML document.
- **head**: This contains meta-information about the document, such as its title, meta tags, and links to scripts and stylesheets.
- **body**: This contains the content of the web page that is rendered in the browser.

Here is a basic example of an HTML document:

```
<!DOCTYPE html>
<html>
<head>
    <title>My First HTML Page</title>
</head>
<body>
    <h1>Welcome to My First HTML Page!</h1>
    <p>This is a paragraph.</p>
```

```
</body>
</html>
```

Chapter 2

Introduction to Text

In computer systems, text is represented in files as a sequence of characters, each of which corresponds to a specific number, known as a character code. These character codes are then stored in the file as binary data.

2.1 Newlines and Carriage Returns

Two of the character codes that have special meanings are the newline (often represented as '\n') and the carriage return (often represented as '\r').

The newline character signifies the end of a line of text and the beginning of a new one. The carriage return character moves the cursor to the beginning of the line. The use of these characters can vary between operating systems. Unix-based systems (like Linux and MacOS) use the newline character to indicate the end of a line, while Windows systems use a combination of a carriage return and a newline ('\r\n').

2.2 ASCII

The American Standard Code for Information Interchange (ASCII) is one of the earliest character encodings. It uses 7 bits to represent each character, allowing it to define up to $2^7 = 128$ different characters. These include the English alphabet (in both lower and upper cases), digits, punctuation symbols, control characters (like newline and carriage return), and some other symbols.

2.3 UTF-8

UTF-8 (8-bit Unicode Transformation Format) is a variable-width character encoding that can represent every character in the Unicode standard, yet remains backward-compatible with ASCII. For the ASCII range (0-127), UTF-8 is identical to ASCII. However, it can use additional bytes (up to 4 bytes in total) to represent characters that are not included in ASCII, such as characters from other languages, emojis, and many other symbols. This has made UTF-8 a widely used encoding in many modern systems.

Chapter 3

Stop Words

Chapter 4

Stemming and Lemmatization

Stemming and lemmatization are two fundamental techniques in natural language processing that are used to prepare text data. They help in reducing inflectional forms of a word to a common base form.

4.0.1 Stemming

Stemming is the process of reducing a word to its word stem (its basic form). For example, a stemming algorithm would reduce the words ″jumping″, ″jumped″, and ″jumps″ to the stem ″jump″.

It's important to note that stemming may not always lead to actual words. For example, the stem of the word ″running″ could be ″runn″ depending on the stemming algorithm used.

Stemming is generally simpler and faster than lemmatization, but it is also less precise.

4.0.2 Lemmatization

Lemmatization, on the other hand, reduces words to their base or root form, which is linguistically correct. For example, ″running″ and ″runs″ are both changed to ″run″.

Lemmatization uses a more complex approach to achieve this. It considers the morphological analysis of the words and requires detailed dictionaries, which the algorithm can look through to link the form back to its lemma.

To summarize, both stemming and lemmatization help in text normalization and preprocessing, but while stemming can be faster and simpler, lemmatization is more accurate, as it uses more informed analysis to create groups of words with similar meanings based on the context.

Appendix A

Answers to Exercises

INDEX