Indian languages requirements for string manipulations on Web

1 Introduction

W3C Standard Document Character Model for the World Wide Web-String Matching gives specifications authors, content developers and software developers a general reference on string identity matching and searching on the World Wide Web. The goal of this document is to make web to process and transmit the text in a consistent, proper and clear way. The successful character model permits documents of web works on different writing systems, scripts, and languages on different platforms so that seamless information can be exchanged, read, and searched by the consumers on the web around the world [1].

A string-searching document of W3C covers string-searching operations on the Web in order to allow greater interoperability. String searching refers to matching of natural language through the "find" command in a Web browser[2].

It is possible to generate the same text with different character encodings. The Unicode allows this mechanism for the identical text. Normalization is the mechanism by Unicode that is usually perform while string search and comparisons. It converts the text to use all pre-composed and decomposed characters [3].

2 Variations in user inputs

2.1 Different preferences by the users

The Unicode Standard gives different alternatives to define text but requires that both text should be treated identical. In order to improve efficiency, it is recommended that an application will normalize text before performing string manipulation operations such as search, comparisons on the web. The different variations can occur while define Unicode text such that same character used different Unicode code points sequences [4]. This will cause unexpected results while searching and matching of string by the users as both string uses different code points. Additionally in Indian languages, the same text represents two orthographic representations with different encoding. The spelling variations lead to introduce the inappropriate searching results. The different users can use different spellings of the same text, as both spellings are appropriate. Some examples are shown below:

हिंदी/ɦĩdi/ &हिन्दी/ɦĩdi/ ,मंडी/mə̃ɖi/ &मण्डी /mə̃ɖi/,चम्पक /tʃə̃pək/ &चंपक /tʃə̃pək/

These types of spelling variation may occur in other Indian languages also.

2.2 Keyboard representation

It is requires by the Unicode to store and interchanged the characters in the same logical order or we can say that order that user typed through the keyboards. It is not always true that in the different keyboard layouts, keystrokes and input characters are same and one to one. It is depends on the type of the keyboard layout. Some keyboards can produce numerous characters from a single key press and some keyboards use different keystrokes to produce one abstract character. It is the limitations of Indian languages that too many characters

need to be fit in one single keyboard. This leads to input more complex Indian languages input methods and which makeover keystrokes sequence in character sequences [4].
The Unicode Standard needs that characters can be stored and interchanged in logical order, i.e. roughly corresponding to the order in which text is typed in via the keyboard or spoken. The main limitations of Indian languages are that a limited number of keys can fit on a keyboard. Some keyboards will generate multiple characters from a single key press. In Indian languages, too many characters to fit on a keyboard and must rely on more complex input methods, which transform keystroke sequences into character sequences. It might be occurs that different character sequences of the same text used by different users from the different keyboard and create issues in string identity matching.

3. Use-cases

Normalization not always takes place in string manipulation and comparison. For example, when we define class/Id name in HTML and the same identical name will be used in class name selectors in CSS with different sequences. As result, selectors would not match the class name and the user does not get the desired output. The following are some of the examples that show the variations in the Hindi characters:

3.1 Text variation in syntactic content under HTML/CSS & other applications

The role of syntactic content in a document format and protocol is to represent the text that defines the structure of the document format and protocol. The different values used to define id, class name in markup languages and cascading style sheets are a part of syntactic content. In order to produce output as desired, we should ensure that the selectors and id or class name should be same as shown in the below example:

```
<!DOCTYPE html>
<html>
<head>
<style>
#हिंदी-ज़ांच{
 text-align: center;
 color: red;
}
</style>
</head>
<body>
<p id="हिंदी-ज़ांच">Text in red color</p>
<p>This paragraph is not affected by the style. </p>
</body>
</html>
```

In the above example, the id name defines in the HTML and CSS works on the same character sequences. Gaps will be there if id name uses different character sequences .This is particularly occurs and leads an issue if markup language and the CSS are being handled or maintained by different persons.
Below examples shows the different character sequences as per Unicode Code Charts and different choices by the users on writing the characters as both forms can be written[5].

हिंदी **/ɦĩdi/**= **0939+ 093F+0902+0926+0940**

हिन्दी **/ɦĩdi/=0939+ 093F+0928+094D+0926+0940**

In addition, other below words used identical text with different character sequences.

ज़ांच **/zatʃ/**= **095B+093E+0902+091A**

ज़ांच **/zatʃ/=091C+093C+093E+0902+091A**

There are two types of variations are seen especially in Indian language i.e. spelling variations and different character sequences.

The character sequences should be same in order to get the right results.

Therefore, it is important that characters – to-characters should match so that proper string manipulation should be made on the web.

3.2 Implementation of Internationalized domain name & Email addresses

The user does not have the knowledge of normalized form; user might be use different character sequences for domain name in Indian languages. So, it is required to implement different types of variations while searching and comparison of the strings so that the web document formats and protocols performed the right string-matching operation and user perceive the results.

3.3 Indian language search operations on the web

User can search natural language content by using find command on the web. Different Users might use different character sequences of the same text by performing find command. There should be some common mapping and implementation in order to satisfy the users need. There user might expect that typing one character will find the equivalent character in the same script such as in Devanagari script ल that represents DEVANAGARI LETTER LA and ळ that represents DEVANAGARI LETTER LLA etc.

The few examples are shown below [6]:

आवा़ज़ **/avaz/** :**0906 + 0935 + 093E + 095B**

आवाज़ **/avaz/** :**0906 + 0935 + 093E + 091C + 093C**

फ़ाँसी **/fãsi/** :**095E + 093E +0901 + 0938 + 0908**

फ़ाँसी **/fãsi/** :**092B + 093C + 093E + 0901 + 0938 + 0908**

चम्पक /**tʃə̃pə**k/ : **091A + 092E + 094D + 092A + 0915**

चंपक **/tʃə̃pək**/ : **091A + 0902 + 092A + 0915**

Additionally in Indian languages, some of the text represents two orthographic representations with different encodings. The spelling variations lead to introduce the inappropriate searching results. Some examples are shown below:

हिंदी /ɦĭdi/&हिन्दी /ɦĭdi/ ,मंडी /mə̃ɖi/&मण्डी /mə̃ɖi/ ,चम्पक /ʧə̃pək/ &चंपक /ʧə̃pək/, ठंडा /ʈʰə̃ɗa/&ठण्डा /ʈʰə̃ɗa/, अम्वर /ə̃ʊər/&अंवर /ə̃ʊər/

4. Requirements for Indian languages

The above-defined W3C draft Standards specify the requirements while implementing string matching of syntactic content and search of natural language content by using matching rules.
The following Indian language requirements need to introduce in the standards in order to perform proper string operations on the web:

i. Different characters variations of Indian languages need to address in the current Standards that have orthographically identical but uses different character encoding as discussed in earlier sections.
ii. Equivalent form of character in the same script as discussed in above sub section 3.3
iii The above discussed string variations should be implemented for proper string comparison on the web
iv. Other requirements such as Singleton mapping

ॐ [U+0950 DEVANAGRI OM] &

🕉 [U+1F549 OM SYMBOL].

In addition, it is recommended that Indian languages characters need to be post processed through normalization defined by Unicode for comparison and searching on the web. Unicode specifies the different normalized forms. such as NFD, NFC , NFKC etc. and discussed in Unicode technical report on Normalization forms [6]. It is recommended that NFC is best suitable normalize form for string manipulation on the web.

REFERENCES

[1] Character Model for the World Wide Web: String Matching, 2021: https://www.w3.org/TR/charmod-norm/
[2]W3C String Searching, 2020: https://w3c.github.io/string-search/
[3] Unicode Consortium: https://home.unicode.org/
[4] Normalization in HTML and CSS: https://www.w3.org/International/questions/qa-html-css-normalization/
[5]Unicode Devanagari Code Chart,2020: https://unicode.org/charts/PDF/U0900.pdf
[6] Unicode Normalization Forms, 2020: http://www.unicode.org/reports/tr15/