PDF Font mapping

Context

1. Any data stored by computers is at the end of the day is a bunch of bytes.
2. Encodings define how to interpret the bytes as a stream of characters. For example, Unicode defines the bytes corresponding to “U+0041” corresponds to the character “A”.
3. Now comes fonts. Fonts define glyphs where each glyph maps to one or more Unicode characters.

    ▼ Example glyphs

Different glyphs for the Unicode character ‘a’ from different fonts

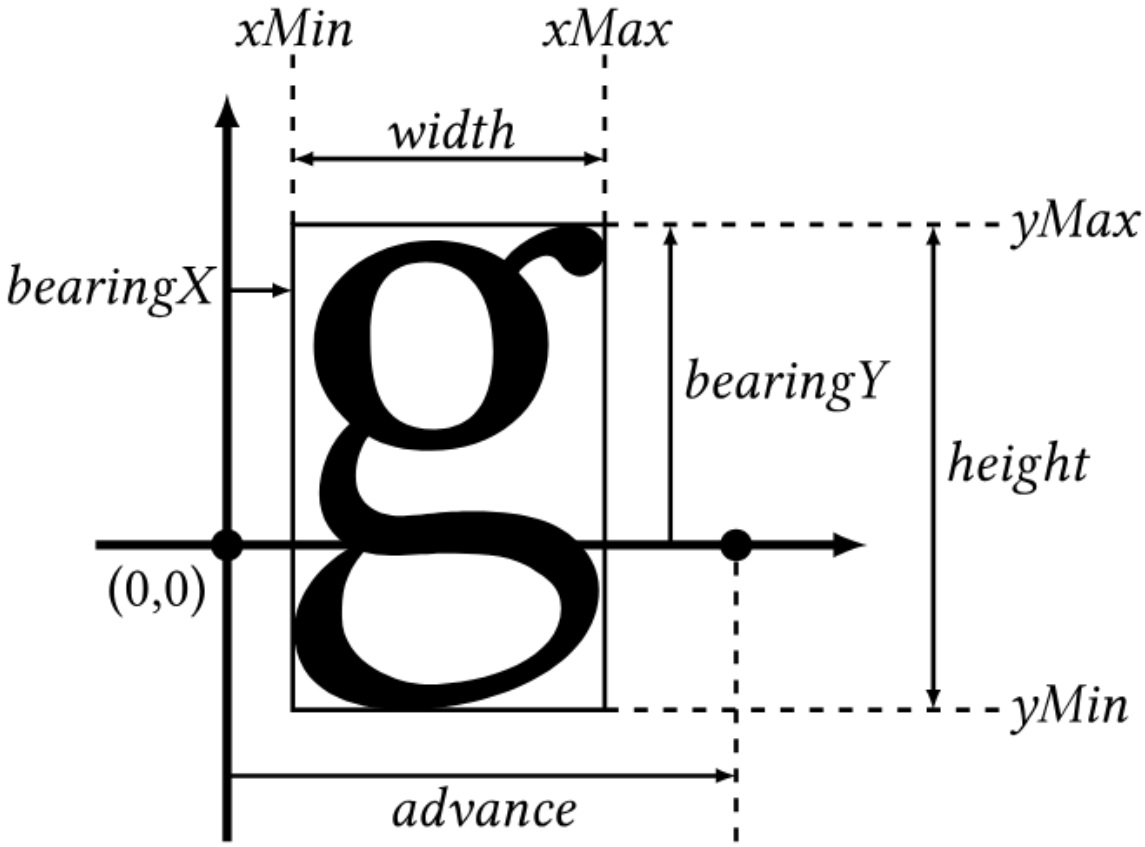
4. What we see when we open a pdf is actually the glyphs corresponding to the Unicode text stored in the pdf.
5. When we copy text from a pdf or use any pdf text extraction library like Pdfium, we obtain the Unicode text stored in the pdf.

Problem

Now the problem is that there exist many fonts (especially Indic fonts) where the human understanding of the glyph doesn't correspond to the corresponding Unicode character.

For example, in my font, I can map the Unicode character “U+0041” which corresponds to the character “A” to the following glyph:

▼ glyph



So, if we extract text from a pdf using this font, we will extract the character ‘A’ but any human viewing the pdf will read it as the character ‘g’.

Proposed solution

Essentially, we need a classifier to map any glyph to its corresponding character as defined by Unicode and we can extract text just the way a human would read it. Problem solved.