1.
'Meiryo', standard font in Windows Japanese edition.

Kangxi Radical Sun(U+2F47): '⽇'

CJK Unified Ideograph Sun(U+65E5): '日' <- Copy and Paste this char from PDF, it should be U+65E5, but you get U+2F47.

(PDF.js on Chrome or Firefox does not reproduce this probrem. Please try with Acrobat Reader.)

2.
The ToUnicode generation algorithm also has problems with European languages.

'Calibri', standard font in Windows has ligatures,
'ff' 'fi' 'fl' 'ffi' 'ffl' 'fb' 'ffb' 'fh' 'ffh' 'fj' 'ffj' 'fk' 'ft' 'fft' 'tf' 'ti' 'tt' 'ttf' 'tti'

Copy and Paste these chars from PDF, you get correct text for first five, but fail for the last fourteen. Because there is no single unicode char for them.

3.
Changing the algorithm can affect various scripts. For example, Arabic.

بِسْـــمِ اللّهِ الرَّحْمـَنِ الرَّحِيم

There are several options on how to handle these complex chars. (ActualText or traditional ToUnicode using reverseMap(), see sample patch.) In above specific case, ActualText seems to work well, but should be checked for various scripts.