These three words have PUA-encoded characters, typed in directly. None of them is searchable in either a webpage or a PDF document. They also won't be read correctly by a screen reader.

written click Eliminate

Next the special characters are all produced by OpenType features. In "written" and "click," however, the substituted ligatures still have associated PUA code points. They are searchable in webpages but not in PDF documents (not consistently, anyway: they are searchable in Adobe Acrobat Reader DC but not in Apple Preview or Firefox; also, the Apple screen reader will not read them correctly). In "Eliminate," the variant characters are also produced by OpenType features, but the substituted characters are all **unencoded** duplicates of characters encoded in the PUA: because the PUA code points are not present in either a webpage or a PDF, they are searchable everywhere (also, a screen reader will pronounce this word correctly).

written click Eliminate

Next are a few letters with diacritics, defined by MUFI. First the PUA-encoded versions, which can't be searched as æ, o or u:

ǣ̆ ö́ u̍

Next the letters typed as base letter + combining mark(s). These are searchable (and most software either can or will ignore the combining marks):

ǣ̆ ö́ u̍

Next a new MUFI combining character: ð with combining dotless i above, where an alternate shape of ð is substituted to accommodate the combining mark. The first is PUA encoded, and the second consists of an unencoded alternate ð plus an unencoded alternate combining mark. You can search on ð for the second but not the first. The third case is eth followed by the PUA-encoded combining mark U+F02F. You can still search on ð because no substitution has been made, but no alternate ð is substituted, and the mark is not positioned correctly over the ð. This is because LibreOffice has isolated the mark in its own "run," and so it has no context to interact with.

ðͥ ðͥ ðͥ

Something similar will happen every time with PUA-encoded combining marks. Use the PUA-encoded version and things will go wrong; use the unencoded version and it will be correctly positioned:

wͥ wͥ Eͥ Eͥ