Sub CleanUpOCR()

Dim aPar As Paragraph

Dim aRange As Range

SetLanguage

FormattingToMarkup

For Each aPar In ActiveDocument.Paragraphs

 Set aRange = aPar.Range

 If Len(aPar.Range.Text) > 2 Then

 aRange.End = aRange.End - 1

 aRange.Select

 Selection.Cut

 Selection.PasteAndFormat (wdFormatPlainText)

 End If

Next aPar

 'Tags door opmaak vervangen

 fkt\_Search "<b>", "</b>", "Fett", False

 fkt\_Search "<i>", "</i>", "Kursiv", False

 fkt\_Search "<u>", "</u>", "Unterstrichen", False

End Sub

Sub FormattingToMarkup()

 '(c) René Probst

 Dim oRange As Range, oRng As Range

 Set oRange = ActiveDocument.Range

 Application.ScreenUpdating = False

 With oRange.Find

 .Font.Bold = True

 Flag = .Execute

 While Flag = True

 Set oRng = .Parent

 If Asc(oRng.Characters.Last.Text) < 30 Then

 'falls Formatierung über das Zeilenende hinausgeht

 oRng.SetRange oRng.Start, oRng.End - 1

 End If

 oRng.InsertBefore "<b>"

 oRng.InsertAfter "</b>"

 oRange.SetRange Start:=.Parent.End + 1, End:=ActiveDocument.Range.End + 1

 Flag = .Execute

 Wend

 End With

 Set oRange = ActiveDocument.Range

 With oRange.Find

 .Font.Italic = True

 Flag = .Execute

 While Flag = True

 Set oRng = .Parent

 If Asc(oRng.Characters.Last.Text) < 30 Then

 'falls Formatierung über das Zeilenende hinausgeht

 oRng.SetRange oRng.Start, oRng.End - 1

 End If

 oRng.InsertBefore "<i>"

 oRng.InsertAfter "</i>"

 oRange.SetRange Start:=.Parent.End + 1, End:=ActiveDocument.Range.End + 1

 Flag = .Execute

 Wend

 End With

 Set oRange = ActiveDocument.Range

 With oRange.Find

 .Font.Underline = True

 Flag = .Execute

 While Flag = True

 Set oRng = .Parent

 If Asc(oRng.Characters.Last.Text) < 30 Then

 'falls Formatierung über das Zeilenende hinausgeht

 oRng.SetRange oRng.Start, oRng.End - 1

 End If

 oRng.InsertBefore "<u>"

 oRng.InsertAfter "</u>"

 oRange.SetRange Start:=.Parent.End + 1, End:=ActiveDocument.Range.End + 1

 Flag = .Execute

 Wend

 End With

 ActiveDocument.Range.Font.Reset

 Application.ScreenUpdating = True

End Sub

Function fkt\_Search(strStart As String, strEnd As String, strAktion As String, Optional bInclude As Boolean = False)

 Dim rng As Range

 Dim rng2A As Range, rng2E As Range

 Dim rngText As Range

 ' Range festlegen

 Set rng = ActiveDocument.Range

 ' Range festlegen

 Set rngText = ActiveDocument.Range(0, 0)

 rngText.Collapse wdCollapseStart

 ' Such-Schleife

 With rng.Find

 .Format = False

 .Text = strStart

 ' Suche nach Start-Tag

 .Execute

 Do While .Found = True

 ' Fundstelle mit Start-Tag anlegen

 rngText.SetRange rng.Start, rng.End

 ' Suchtextbereich reduzieren

 rng.SetRange rng.End, ActiveDocument.Range.End

 ' Suche nach End-Tag

 .Execute FindText:=strEnd, Forward:=True

 ' Abbruch wenn kein End-Tag

 If .Found = False Then Exit Function

 ' Fundstelle bis End-Tag erweitern

 rngText.SetRange rngText.Start, rng.End

 rngText.Select

 If bInclude = True Then

 ' mit Tags

 ' Beispiel: Kursiv, Fett, Unterstrichen...

 Select Case strAktion

 Case "Kursiv"

 rngText.Font.Italic = True

 Case "Fett"

 rngText.Font.Bold = True

 Case "Unterstrichen"

 rngText.Font.Underline = wdUnderlineSingle

 End Select

 ElseIf bInclude = False Then

 ' ohne Tags, Tags löschen

 ' Beispiel: Kursiv, Fett, Unterstrichen...

 Set rng2A = ActiveDocument.Range(0, 0)

 Set rng2E = ActiveDocument.Range(0, 0)

 rng2E.SetRange rngText.End - Len(strEnd), rngText.End

 rng2E.Select

 rngText.SetRange rngText.Start, rng.End

 rng2A.SetRange rngText.Start, rngText.Start + Len(strStart)

 rng2A.Select

 rngText.SetRange rngText.Start + Len(strStart), rngText.End - Len(strEnd)

 Select Case strAktion

 Case "Kursiv"

 rngText.Font.Italic = True

 Case "Fett"

 rngText.Font.Bold = True

 Case "Unterstrichen"

 rngText.Font.Underline = wdUnderlineSingle

 End Select

 rng2E.Delete 'Start-Tag löschen

 rng2A.Delete ' End-Tag löschen

 End If

 ' Suchtextbereich zur Endposition reduzieren

 rng.Collapse wdCollapseEnd

 ' Start-Tag suchen

 .Execute FindText:=strStart, Forward:=True

 Loop

 rng.Collapse wdCollapseEnd

 End With

End Function

Sub SetLanguage()

Dim rngStory As Word.Range

Dim lngJunk As Long

Dim aShp As Shape

Dim aLanguage As String

Dim aLanguageID As String

aLanguage = InputBox("Type the TMX-style code for your target language (de-DE, es-ES, fr-FR, it-IT, nl-NL, en-GB etc.)", "Enter target language", "nl-NL")

Select Case aLanguage

 Case "de-DE"

 aLanguageID = wdGerman

 Case "en-GB"

 aLanguageID = wdEnglishUK

 Case "fr-FR"

 aLanguageID = wdFrench

 Case "nl-NL"

 aLanguageID = wdDutch

 Case "es-ES"

 aLanguageID = wdSpanish

 Case "it-IT"

 aLanguageID = wdItalian

 Case Else

 'Do Nothing

 aLanguageID = wdNoProofing

End Select

lngJunk = ActiveDocument.Sections(1).Headers(1).Range.StoryType

For Each rngStory In ActiveDocument.StoryRanges

 'Iterate through all linked stories

 Do

 On Error Resume Next

 rngStory.LanguageID = aLanguageID

 Select Case rngStory.StoryType

 Case 6, 7, 8, 9, 10, 11

 If rngStory.ShapeRange.Count > 0 Then

 For Each aShp In rngStory.ShapeRange

 If aShp.TextFrame.HasText Then

 aShp.TextFrame.TextRange.LanguageID = aLanguageID

 End If

 Next

 End If

 Case Else

 'Do Nothing

 End Select

 On Error GoTo 0

 'Get next linked story (if any)

 Set rngStory = rngStory.NextStoryRange

 Loop Until rngStory Is Nothing

Next

End Sub