# Light Technical Guide to Handling Subtitles Offline,

# Aligning Subtitle Languages for Comparison,

# & Creating Voice-Over Script from Subtitles

A screenshot of a computer

Description automatically generated

Subtitle-Tool programmed by: Joschka Rick

**0. Download**

The tool can be downloaded at

<https://github.com/joschkarick/the_week_scripts>

The other scripts are available at

<https://github.com/TextGenes/subtitletools>

(Scripts by beginner Dimitri Molerov with ChatGPT support, use with care).

## Handling and displaying subtitles

Intro:

Saving single file: After download of, e.g., youtube-subtitles (YT) as ‘.srt’-file, you can open the saved file in Notepad editor to preview or edit. You can also edit the text and save again as .srt file. (Make sure to keep the format of the timings intact to avoid compiling errors). Time could be manually adjusted, but it’s easier to do within Youtube studio or another subtitle editor program. After saving, upload the updated .srt-file again to test sync with your video and keep consistent.

You can split and merge subtitles in youtube editor. It updates the number and format. (However, need to rerun below for aligned versions). Mind the youtube timing is not very precise (0.0-0.20, 1.0 second) and can get cumbersome for professional editing.

To display subtitles in Vimeo: upload .srt file for language options. Press CC in player and choose language option to display. For text pictures (in conversation starters), increase the subtitle background to 100%.

Vimeo settings to set opacity – for better focus:A screenshot of a computer

Description automatically generatedexample (conversation starter, full overlay, at 75%):

A screenshot of a video

Description automatically generated (see … file for subtitle word art)

## Align several subtitle files for joint editing and review

Purpose

For comparing multiple subtitle versions and team editing, you can copy the content, e.g., to an excel file, each .srt-file into one separate column with nothing else in it. When done with edits, copy the entire column back into the Notepad and save as .srt-file. Reupload to youtube or other video host.

Aligning subtitles (heights) mergered/separated: Downloaded Youtube subtitles are automatically numbered and listed. However, in case of one-liners, subtitle mergers, or subtitle separations, the file length changes. When merging two subtitles into one or separating one into two subtitles, the number of subtitles changes. Doing this in Youtube studio/other editor is strongly recommended (keeps score of the automatic numbering). However, the overall length changes, too. The same sections in different files are not on the same line (height) anymore (this hinders comparison). The following procedures help with creating an overview in excel.

VBA scripts can help change excel contents. Use with care, save a copy! Below are some scripts.

Loading VBA script: To access in excel (2016): go to Developer > Visual Basic. Insert > Module (opens the space). File > import > [script name.bas] & open (loads the Macro). Excel > Developer > Macro (should now have the Macro name) > Run. Or Edit (to change the code). Run may take 5 minutes.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**6**

**2**

**5**

**4**

**3**

**1**

Optional: Coding VBA script: To make own changes, type in script > Debug1. Run (troubleshoot). Run. (ask GPT to explain). Press G to see script running (debug notes). Copy or export finished code.

1. Format subtitles one-liners > Two-liners

In the Subtitle Tool: 1. Browse and open saved input .srt-file. 2. Press “Save SRT file as formatted” with a file name and save. => After each one-liner an additional empty line has been added, making all subtitles the same length.

1. Adjust speaker timings format

Paste the speaker timings (the first time or for changes to material) in column one (later two).

Load and run the two VBA scripts:

2a Speaker-Timing-format-add-leading-zero (0:00:00 -> 00:00:01)

2b Speaker-Timing-format-spaceout-firstcolumn-to-every-fifth-row (spaces the timings out in height).

1. Convert start timings to seconds

Place the speaker timings in column B. In the column A paste the formula that converts timings to seconds (00:01:14 -> 74). And pull down the formula across the whole column height next to timings.

See formula in appendix.

1. Paste subtitles and align subtitles columns

Paste one subtitle file in one column. For the scripts, place it in column G, L etc. (always five apart)

Copy the formula from step 3 for any new or updated subtitle columns, too. (in columns F, K etc.; seconds go left of the subtitles for the scripts).

5x5 rows x columns (1 seconds, 2 subtitles, 3-5 empty/comments) are considered one subtitle block for one version or language, which stays together. Place the first subtitle in number 1 in row 10. (The formatting up to now ensures they have this standard size.)

Once you have two or more versions/languages in the columns, with seconds: load and run script:

4-Align-subtitle-5x5blocks-in-columns-by-height

It will match the subtitles by the seconds, within 2 seconds difference (sensitivity).

* You get a nice side-by-side overview of all languages.

A screenshot of a computer

Description automatically generated

You can pin the first (or any) language version for comparison.

Work on subtitles in the second column of each block as needed, place comments, e.g., for timing or questions in columns 3 to 5. Comments will stay aligned with subtitles when using the scripts.

Challenge: However not if subtitles are split/merged in youtube. Then readjust manually.

Rerun the script and prior steps when adding new languages. The script is made for 10 blocks wide and 1200 subtitle blocks down. You can add more by editing the script: change the last cell (‘AI6000’).

1. Optional: Delete empty cell blocks

If it gets too messy, you can delete empty blocks in a column. Run this script(several times if needed):

5-Align-subtitle-5x5blocks-remove-empty-blocks-in-column

1. Optional: Verify complete subtitles

To verify if all subtitles are still there, this script checks the subtitle numbers in the current column:

6-Subtitle-missing-numbers-check-in-column

## Creating a Voice-Over Doc from Subtitles

1. Input Speaker timings file

Use a given speaker timing file, or if needed, recreate speaker timings from youtube or other video, using voice recognition software. Mind the output format of the speakers. It’s helpful to have names and start times or start seconds (to merge or align):

A screenshot of a computer

Description automatically generated

(Save as csv: The speaker file formatting needs to be saved as German .csv. i.e., the separator is a comma. A csv is an excel file saved for an editor, where each column is separated by a separator character (e.g, “;”). Do this by briefly setting your computer settings (windows) timing to Germany (see screenshot). Open the speaker file in excel and then save as .csv. It will now automatically save with a comma as separator sign. Or set separators signs yourself with open office. This is recognized by the Subtitle Tool.)

* Challenge: Some recognition software seems to lag behind 1-2 seconds. Subsequently many speaker names are slightly early, unless corrected manually.

1. **Subtitle.srt to Voice-Over Script / Transcript**

In the subtitle tool: Browse and open .srt-file. Browse and open speaker timings file.

Press button “Merge Srt file and speaker file…” and save the doc file.

=> You receive a Word document with all the speakers’ names followed by their respective lines (as given in the subtitles).

The timings are currently not included, but could be if needed.

=> Challenge: The Tool does this by placing each speaker at the time which most overlaps with the previous title. Manual adjustment may be needed. First check if the speaker timings are correct.

=> Spoken language could be longer or different than subtitles and vice versa.

## Appendix (Script code for Copy & Paste):

1 After .srt download

run through Joschka's script to add additional spaces after oneliners subtitles.

save result, open with notepad, and copy one subtitle file into one excel column

2.0 converting youtube subtitle timings

VBA Script (see general info)

2a Time Format - Add Leading Zero to speaker timings

VBA script

From format 0:00:00.00 to 00:00:00.00

Script:

Sub FormatSpeakerTimings()

Dim ws As Worksheet

Dim lastRow As Long

Dim currentRow As Long

Dim currentRow As Long

'Set the worksheet

Set ws = ActiveSheet

'Set ws = ThisWorkbook.Sheets("...")

' Change "Sheet1" to your actual sheet name

' Find the last row in current column (replace by number of column with the timings; typically only needed for speaker timings, as yt timings have a fixed format)

lastRow = ws.Cells(ws.Rows.Count, currentCol).End(xlUp).Row

' Loop through each row in column A

For currentRow = 1 To lastRow

' Check if the time string starts with one digit followed by ":"

If Left(ws.Cells(currentRow, currentCol).Value, 2) = "0:" Then

' Add leading zero if necessary

ws.Cells(currentRow, currentCol).Value = "0" & ws.Cells(currentRow, currentCol).Value

End If

Next currentRow

End Sub

2b Space out speaker timings

VBA script

From 1 in 1 to 1 in 5 cells for speaker timings, adds four empty cells between:

Script:

Sub AddEmptyCellsBetweenTimings()

Dim ws As Worksheet

Dim lastRow As Long

Dim currentRow As Long

Dim currentColumn As String

' Set the worksheet

Set ws = ActiveSheet

' Set ws = ThisWorkbook.Sheets("...")

' Change "Sheet1" to your actual sheet name

' Set the column with speaker timings

currentColumn = "A" ' Change "A" to your actual column letter

' Find the last row with data in the specified column

lastRow = ws.Cells(ws.Rows.Count, currentColumn).End(xlUp).Row

' Loop through each row starting from the last row

For currentRow = lastRow To 2 Step -1

' Insert four empty cells below each timing

ws.Range(currentColumn & currentRow + 1 & ":" & currentColumn & currentRow + 4).Insert Shift:=xlDown

Next currentRow

End Sub

3 convert timings in seconds

Excel Formula

Paste in cell, extra column, left of timings, paste and pull down across column

English system formula (untested)

=IFERROR(VALUE(MID(OFFSET(INDIRECT(ADDRESS(ROW(), COLUMN())), 0, 1), 1, 2)) \* 3600 + VALUE(MID(OFFSET(INDIRECT(ADDRESS(ROW(), COLUMN())), 0, 1), 4, 2)) \* 60 + VALUE(MID(OFFSET(INDIRECT(ADDRESS(ROW(), COLUMN())), 0, 1), 7, 2)), 0)

German system formula

=IFERROR(VALUE(MID(OFFSET(INDIRECT(ADDRESS(ROW(); COLUMN())); 0; 1); 1; 2)) \* 3600 + VALUE(MID(OFFSET(INDIRECT(ADDRESS(ROW(); COLUMN())); 0; 1); 4; 2)) \* 60 + VALUE(MID(OFFSET(INDIRECT(ADDRESS(ROW(); COLUMN())); 0; 1); 7; 2)); 0)

4 aligning multiple versions in columns (in blocks) by height

VBA script

Preparation:

Make sure that subtitles are formatted regularly, at 5 row intervals, with steps 1 and 3 completed (2 for new speaker timings).

Each 'cell block comprises 5x5 cells, so there should be 5 columns between one timing in seconds and the next one.

(columns suggestion: 1 timing in seconds, 2 timing speaker (+names)/subtitle column, 3-5 empty/comments).

(The first cell with timing in seconds ("0") would be in A10.)

(Place the first (additional) timing in B11, e.g., "00:00:01" in B11, gives "1" in A11)

This script runs for cells A10 to AI6000.

Up to ten subtitle columns (or 5 languages with two versions) can be posted. For more, change to higher letters for: AI6000. If it outruns row 6000, change to more rows.

Function:

Comparisons are performed on the respective cell (2nd row, 1st column) in each block (subtitle start timing in seconds), if it contains a number higher than 0 (all others ignored).

In each block, the scripts picks comparison cell (timing in seconds) and compares it to others in the row (if there is more than one timing in this row).

If any timings are more than 2 seconds higher than others in the row, the higher timings are moved down a block, with comments and all, by inserting a 5x5 empty cell block.

At the end, you may pull along the numbers formula from step 3 (but not needed, all empty are 0).

For repeated use:

You can copy in subtitles in another language or version in the next 5x5 block.

Else, when copying a subtitle column back into notepad and saving as .srt, when you upload to youtube, the empty rows will be cleared.

(So, if you redownload an .srt, repeat steps 1 (oneliners in program), 3(pull up formula in seconds) and run 4 (height alignment) to realign).

Script:

Sub SortAndShiftBlocks()

Dim ws As Worksheet

Dim rng As Range

Dim rowBlock As Range

Dim blockRange As Range

Dim compareValue As Double

Dim sensitivity As Double

Dim referencevalue As Double

Dim candidateCollection As Collection

Dim comparisonCollection As Collection

Dim targetCell As Range

Dim emptyBlock As Range

Dim i As Long, j As Long, k As Long, l As Long, m As Long

' Set the worksheet to the active sheet

Set ws = ActiveSheet

'alternative for precise: Set ws = ThisWorkbook.Sheets ("EP1-prefinal") ' Change to your actual sheet name

' Set the range from A10 to AI6000

Set rng = ws.Range("A10:AI6000")

' Set sensitivity

sensitivity = 2

' Loop through each row of blocks

For Each rowBlock In rng.Rows

' Check if the current row satisfies the conditions

If rowBlock.Row Mod 5 <> 0 Then

' Your processing code here

GoTo NextRowBlock

End If

' Print the current row number for debugging purposes

Debug.Print "Current Row: " & rowBlock.Row

' Initialize candidate collection

Set candidateCollection = New Collection

' Loop through each 5x5 block within the row

For j = 1 To rowBlock.Columns.Count Step 5 ' Start from the 1st column, increment by 5

' Set the current block range

Set blockRange = rowBlock.Columns(j).Resize(5, 5)

' Compare each comparison value to find those above 0

For i = 1 To blockRange.Columns.Count Step 5

If Not IsEmpty(blockRange.Cells(2, i).Value) Then

If IsNumeric(blockRange.Cells(2, i).Value) And blockRange.Cells(2, i).Value > 0 Then

compareValue = blockRange.Cells(2, i).Value

candidateCollection.Add compareValue

End If

End If

Next i

Next j

' If there are fewer than 2 items in the candidate collection, or if the comparison collection is empty, move to the next row block

If candidateCollection.Count < 2 Then

Set candidateCollection = Nothing

Set comparisonCollection = Nothing

referencevalue = 0

GoTo NextRowBlock

End If

' Find the lowest value in the candidate collection

Dim minValue As Double

minValue = candidateCollection(1)

' Loop through the candidate collection to find the minimum value, that is higher than 0

For k = 2 To candidateCollection.Count

If 0 < candidateCollection(k) And candidateCollection(k) < minValue Then

minValue = candidateCollection(k)

End If

Next k

' Set the reference value as the minimum value plus sensitivity

referencevalue = minValue + sensitivity

' Initialize comparison collection

Set comparisonCollection = New Collection

' Loop through each 5x5 block within the row again

For l = 1 To rowBlock.Columns.Count Step 5 ' Start from the 1st column, increment by 5

' Set the current block range

Set blockRange = rowBlock.Columns(l).Resize(5, 5)

' Compare each comparison value to the reference value

For m = 1 To blockRange.Columns.Count Step 5

If Not IsEmpty(blockRange.Cells(2, m).Value) Then

If IsNumeric(blockRange.Cells(2, m).Value) And blockRange.Cells(2, m).Value > 0 Then

compareValue = blockRange.Cells(2, m).Value

' Check if comparison value is higher than reference value

If compareValue > referencevalue Then

' Add the cell to the collection

Set targetCell = blockRange.Cells(2, m)

comparisonCollection.Add targetCell

End If

End If

End If

Next m

Next l

' If the comparison collection is empty, move to the next row block

If comparisonCollection.Count = 0 Then

Set candidateCollection = Nothing

Set comparisonCollection = Nothing

referencevalue = 0

GoTo NextRowBlock

End If

' Set the active cell to the first column of the current row

rowBlock.Cells(1, 1).Activate

' If there are cells above the reference value, move and insert the empty block

' Move to column one and process the collection

For Each targetCell In comparisonCollection

'Go to the target cell

targetCell.Activate

' Shift the 5x5 block starting from targetCell down

targetCell.Resize(5, 5).Offset(-1, 0).Insert Shift:=xlDown

Next targetCell

NextRowBlock:

' Clear the comparison collection for the next row

Set comparisonCollection = Nothing

' Clear the candidate collection for the next row

Set candidateCollection = Nothing

'reset referencevalues and counters to be sure

referencevalue = 0

minValue = 0

i = 0

j = 0

k = 0

l = 0

m = 0

' Set the active cell to the first column of the current row

rowBlock.Cells(1, 1).Activate

'Move down one more row to restart in next row

Next rowBlock

End Sub

5 optional: Clear empty blocks

VBA script:

Sub RemoveEmptyBlocksInColumn()

Dim ws As Worksheet

Dim rng As Range

Dim rowBlock As Range

Dim blockRange As Range

Dim emptyBlockCount As Integer

Dim startingcolumn As Integer

Dim columnRange As Range

' Set the worksheet to the active sheet

Set ws = ActiveSheet

' Set the range from A10 to AI6000

Set rng = ws.Range("A10:AI6000")

emptyBlockCount = 0

startingcolumn = Application.WorksheetFunction.RoundDown((ActiveCell.Column / 5), 0) \* 5 + 1

Set columnRange = ActiveSheet.Range(Cells(1, startingcolumn), Cells(1, startingcolumn + 4))

' Loop through each row of blocks

For Each rowBlock In rng.Rows

' Check if the current row satisfies the conditions

If rowBlock.Row Mod 5 = 0 Then

' Set the current block range

Set blockRange = rowBlock.Offset(0, startingcolumn - rowBlock.Column).Resize(5, 5)

Debug.Print "Active Range:" & blockRange.Address

' Check if the entire block is empty or contains only zeros

If IsBlockEmpty(blockRange) Then

' Increment the counter for consecutive empty blocks

emptyBlockCount = emptyBlockCount + 1

' Delete the entire block

Debug.Print "Empty blocks deleted in row " & rowBlock.Row

blockRange.Delete Shift:=xlUp

Else

' Reset the counter if a non-empty block is found

emptyBlockCount = 0

End If

' If five consecutive empty blocks are found, exit the loop

If emptyBlockCount = 5 Then Exit For

End If

Next rowBlock

End Sub

Function IsBlockEmpty(blockRange As Range) As Boolean

' Check if the entire block is empty or contains only zeros

Dim cell As Range

' Loop through each cell in the current block

For Each cell In blockRange.Cells

' Set the current cell range

Set cell = cell.Resize(1, 1)

If Not IsError(cell.Value) Then

If Not IsEmpty(cell.Value) Then

If IsNumeric(cell.Value) Then

If cell.Value <> 0 Then

' The cell has a non-zero numeric value

IsBlockEmpty = False

Exit Function

End If

Else

' The cell has a non-numeric value

IsBlockEmpty = False

Exit Function

End If

End If

End If

Next cell

' If no non-empty cells are found, the block is considered empty

IsBlockEmpty = True

End Function

6 optional: check if subtitles are complete

VBA script

Current column, row 10+5…

This script goes through all subtitle numbers

(current column, row 10 and every following fifth row)

ignores empty cells and alerts to any numbers missing between the first (1) and the last.

(might miss missings at the end).

Sub CheckSubtitleNumbers()

Dim ws As Worksheet

Dim subtitleColumn As Long

Dim subtitleCell As Range

Dim subtitleContent As Variant

Dim subtitleNumbers As Collection

Dim missingSubtitles As Collection

Dim expectedSubtitleNumber As Long

Dim i As Long

' Set the worksheet to the active sheet

Set ws = ActiveSheet

' Initialize the collection for subtitle numbers

Set subtitleNumbers = New Collection

' Initialize the collection for missing subtitles

Set missingSubtitles = New Collection

' Set the subtitle column to the active column

subtitleColumn = ActiveCell.Column

' Loop through subtitle rows

For i = 0 To 1200

' Set the current subtitle cell

Set subtitleCell = ws.Cells(10 + 5 \* i, subtitleColumn)

' Check if the subtitle cell is not empty

If Not IsEmpty(subtitleCell.Value) Then

' Get the content of the subtitle cell

subtitleContent = subtitleCell.Value

' Check if the content is numeric and greater than 0

If IsNumeric(subtitleContent) And subtitleContent > 0 Then

' Check if the content matches the expected subtitle number

If subtitleContent <> expectedSubtitleNumber + 1 Then

' Add the missing subtitle numbers to the collection

For j = expectedSubtitleNumber + 1 To subtitleContent - 1

If Not CollectionContains(missingSubtitles, j) Then

missingSubtitles.Add j

End If

Next j

End If

' Update the expected subtitle number

expectedSubtitleNumber = subtitleContent

End If

' Add the numeric content to the collection

subtitleNumbers.Add subtitleContent

End If

Next i

' Display the missing subtitles in a message box

If missingSubtitles.Count > 0 Then

Dim errorMessage As String

errorMessage = "The following subtitles are missing or have incorrect values in column " & Split(Cells(1, subtitleColumn).Address, "$")(1) & ":" & vbCrLf & vbCrLf

' Concatenate the missing subtitles

For Each Number In missingSubtitles

errorMessage = errorMessage & "Subtitle " & Number & " missing in row " & 10 + 5 \* (Number - 1) & vbCrLf

Next Number

MsgBox errorMessage, vbExclamation, "Subtitle Check"

Else

MsgBox "All subtitles in the current block are correct!", vbInformation, "Subtitle Check"

End If

End Sub

Function CollectionContains(coll As Collection, key As Variant) As Boolean

On Error Resume Next

CollectionContains = Not coll(key) Is Nothing

On Error GoTo 0

End Function

9 Shorten code testing time

place in the loop and remove the apostrophes to test on first 200 rows.

'Stop after row 200 for testing

If rowBlock.Row >= 200 Then

Exit Sub

End If