# How to use ET - EmotionTool to analyse the CEVVT

## 0. General description

ET1.0 (Bräuer & Streubel, 2024), is a Python-based computer program that was developed for the automatic encoding of CEVVT data. ET runs within the (Windows) command window. The programme uses a word list to check whether the participants' answers contain emotion words. If emotion words are found, ET reduces them into emotion categories based on their common root word (e.g., "angry", and "angry" were reduced to the category "anger"). New words can be added to the word list during coding.

Reference:
Bräuer, J. S., & Streubel, B. (2024). EmotionTool (ET1.0). Zenodo. https://doi.org/10.5281/zenodo.11244222

## 1. Installation

### a) Windows

1) Download and unzip the EmotionTool-1.0 zip folder
2) If Python is not installed on your computer, an embeddable package for Windows can be used so that ET can access the necessary Python scripts and libraries without having to install Python. To do this, download the latest version of the embeddable package at https://www.python.org/downloads/windows/, unzip it and copy all files into a folder ET1.0\phyton\ you have to create for this purpose (without any further subfolders)

### b) Mac

1) Download and unzip the EmotionTool-1.0 zip folder
2) If Python is not installed on your computer, you can easily install Python. To do this, download the latest version at https://www.python.org/downloads/

## 2. Input & Output

### a. Word Lists

The folder **ET1.0\list** contains the following lists with which the data will be compared:

- ***emotion_words.csv***
  Word list on the basis of which ET codes the data; contains 3 columns: *emotion* (words that are considered emotion words → are searched in the data by ET), *reduction* (reduction/emotion category of the emotion word), *valence* (valence of the emotion word: positive, negative, neutral).
- ***modifier.csv***
  list of modifiers (i.e., additional descriptions of an emotion word like "a bit", "very", ...) that are searched for by ET; contains one column: *modifier*
- ***negation.csv***
  list of negations (e.g., “not”) that are searched for by ET; contains one column: *negation*

All file and column names can be changed in settings.py (see 3b)

### b. Input file

The folder **ET1.0\in** contains the input file for ET (see point 3a)

### c. Output-file

The output file will be stored in the folder **ET1.0\out**; the file name is defined in settings (see 3b); ET will automatically add date and time the file was created to the file name

ET codes (and adds as columns in the output file):

***Emotion:***
Emotion word coded according to the list "emotion_words.csv" (see 1a)

***Reduction:***
Reduction of the emotion word to emotion category based on word stem (e.g., "get angry" and "anger" are reduced to the category "anger"); a reduction is assigned to each emotion word in the list "emotion_words.csv" (see 1a)

***Modifiers:***
Modifiers according to the list "modifier.csv" (see 1a)

***Negation:***
Negations of emotion words; according to the list "negation.csv" (see 1a); negations of emotion words are reduced to the opposite valence of the negated emotion word as indicated in the list "emotion_words" (e.g., "not sad" => positive; "not happy" => negative)

***Problems:***
If ET detects one or more of the following pre-defined problems, the repsective item is not coded, but requires review and manual coding by a trained user:

*"0": "Match in a marked column",*
This problem marks items in which an emotion word appears in a previously defined column (see 3b), e.g., within the answer to the comprehension question. This allows you to manually check whether an emotion word refers to what the vignette's protagonist's feeling or to something else

*"1": "No reduction found",*
With this problem, ET indicates items where an emotion word was found for which no reduction (yet) exists.

*"2": "More than one emotion found",*
With this problem, ET indicates items with more than one emotion word. In this way, a manual decision can be made according to previously defined rules (see 4) as to which emotion word is to be coded, reduced and thus included in the scoring.

*"3": "No emotions found",*
With this problem, ET indicates items in which no emotion word has been found. This allows you to manually check whether there is an emotion word that ET has not yet know.

*"4": "More than one negation found".*
With this problem, ET indicates items with more than one negation. This allows you to manually check if the negations found refer to the emotion word.

### d. Settings
The file "settings.py" contains all necessary settings like file paths for input and output files, columns that are relevant for the coding, etc. (see 3a)

### e. IMPORTANT
Save data and lists as "utf-8-sig" (in German: „CSV UFT-8 (durch Trennzeichen getrennt)") before you let ET working with them (Excel should use the semicolon ";" as a separator; see also settings)

## 3. Procedure

### a. Convert CEVVT transcripts to the correct format
- use template "CEVVT_data_masterfile_TEMPLATE_ET_eng.csv", which is available in the folder **ET1.0\in** to transcribe data
- each line represents a vignette, so that each participant has 20 lines

- Columns G-L represent the questions and possible follow-up questions of the CEVVT (assigning the participant's answers to the (follow up) questions can help later during manual coding to decide which answers are to be coded)
- save input file as .csv (utf-8-sig)

## b. Make all relevant settings

- Open "settings.py" in editor
- Save and close "settings.py" after making changes
- The following settings are necessary and possible:

```
# Filepath of the input-file
INPUT_FILE_URL = "in/CEVVT_data_masterfile_05122023.csv"
```
→ Insert the name of the input file

```
# Filepath of the Emotion-Dictionary
# IMPORTANT: Please name the columns of the dictionary 'emotion',
  'reduction', and 'valence'!

EMOTION_WORDS_URL = "lists/emotion_words.csv"
```
→ Insert the name of the file containing the word list with emotion words, reductions and valence

```
# If a word has a valence on the key side of the dictionary (the word
  before the :), the word on the other side is used as a reduction, if
  there is one negation before the emotion term.
VALENCE_PAIRS = {
    "positive":"negative",
    "negative":"positive",
    "neutral":"neutral"
    }
```
(Here it is specified that ET reduces to the opposite valence in the case of a negation of an emotion word)

```
# Filepath of the negation-wordlist
NEGATIONS_URL = "lists/negation.csv"
```
→ Name of the file containing the list of negations
```
# Filepath of the modifier-wordlist
MODIFIER_URL = "lists/modifier.csv"
```
→ Name of the file containing the list of modifiers

```
# Filepath of the output-file
# IMPORTANT: MUST BE A CSV-FILE
OUTPUT_FILE_URL = "out/ETout_CEVVT_data_masterfile.csv"
```
→ Insert the name of the output file that is created in the ET\out\ folder and contains the encoded data

```
# Columns which are to be searched for emotions by ET
LABELS_TO_LOOK_THROUGH = [
    'Comprehension question',
    'Main question',
    'Follow-up question 1',
    'Follow-up question 2',
    'Follow-up question 3',
    'Follow-up question 4'
    ]
```
→ Here you can specify in which columns of the input file ET searches for emotion words

```
#Columns which raise an error, if a match is found in them
LABELS_RAISING_PROBLEMS = ['Comprehension question ']
```

→ If emotion words are found in the columns defined here, ET declares this as problem “0” and the coding can be checked manually
(This could be, for instance, relevant for children's data, as one column represents the answer to the comprehension question and this may contain emotion words that do not relate to the feeling of the protagonist of the vignette)

```
# Columns, which should be shown next to the default columns, when
"clicking through the lines".
LABELS_TO_SHOW = ["ID subject", "vignette"]
```

→ Here you can specify the contents of which columns from the input file are to be provided additionally during manual coding (these can facilitate manual coding through context)

```
# Name of the coder, which is noted in the output-file
CODER = "JSB"
```

→ The initials of the person who encodes the data manually can be entered here
- Is given in the column "Coder" in the output file

```
# Column-names for the ET specific columns.
# This is a dictionary format (key:value). The key (e.g. "emotion") is
the name of the column inside the program. The value (e.g. "Emotion") is
the name of the column in your input.file.
# IMPORTANT!!!: Don't change the keys (e.g. "emotion")! Only change the
values (e.g. "Emotion")! Changing the keys will result in an error and
break the program!
ET_LABELS = {
    "emotion":"Emotion",
    "reduction":"Reduction",
    "modifier":"Modifier",
    "negation":"Negation",
    "problems": "Problems",
    "coder": "Coder"
    }
```

→ here you can change the values (Entries after the colon) if you want to change the column names of the output file

```
WORDLIST_LABELS = {
    "emotion": "emotion",
    "reduction": "reduction",
    "valence": "valence",
    "modifier": "modifier",
    "negation": "negation"
}
```

→ here you can change the values (Entries after the colon) if you have changed the column names of your word lists

```
NO_EMOTION_FOUND_LABEL = "99"
```

→ here you can change the indicator to be used in the event of missing values (see 3d)

### c. Double-click on "start.cmd" to start ET

- After starting the programme, the windows command window opens and ET automatically compares the CEVVT input data with the word lists (see 2a) and encodes emotion words, reductions, modifiers and negations (see 2c)
- In the command window ET displays
  - entries from the first lines of all word lists and the input file as view examples
  - how many and what type of problems (see point x) occurred during encoding (see Figure 1)

- ET creates the output file in the folder **ET1.0\out**; this contains the date and time of its creation in the file name
- The user can now choose whether she wants to
  [0] exit ET and, if necessary, make changes to the lists or the input file or
  or
  [1] check the items that have not yet been coded and code them manually
- When ET creates new files (output files, word lists), these are each given a date and time stamp in the file name. Older files with the same name are moved to an archive folder

## d. Manual coding of problem items

- The command window will now show the user the items in which ET detected one or more of the above-mentioned problems (see figure 2)
- It is shown for every article:
  - all columns in the input file that have been specified to be searched by ET ("`Columns which are to be searched for emotions by ET`" in settings.py)
  - additional columns in the input file defined in settings.py ("`columns, which should be shown next to the default columns, when "clicking through the lines"`")
  - Problems found (see 2c)
  - Possible encodings (emotion, reduction, modifier, negation) proposed by ET, sorted by the column of the input file where the emotion word was found

- the user has the following options:
  [1] skip → Item is skipped
  [2] own input → a new emotion word and reduction (and possibly modifiers or negation) can be added to the word list (see more under 3e)
  [3] 99 → if no emotion word occured, "99" can be coded as an indicator for missing value
  [4] ... [x] → Selection of one of the codes suggested by ET
- After making an entry, the user can select
  [0] Done → ET is finished; output file is saved
  [1] Next → Edit next item
  [2] Redo → Edit item again

## e. Make an own input

- If the user selects "own input", the following entries are required. The first 4 entries refer to the corresponding entries in the output file, the last 3 entries refer to the list "word_valence.csv"
  - ***Emotion*** Emotion word (in combination with modifiers and negation if necessary) to be coded in the "Emotion" column of the output
  - ***Reduction:*** Reduction, to be coded in the "Reduction" column of the output
  - ***Modifier:*** Modifier to be coded in the "Modifier" column of the output
  - ***Negation:*** Negation to be coded in the "Negation" column of the output
  - ***Stripped Emotion:*** new emotion word to be entered in the word list (modifiers or negations are omitted here)
  - ***Stripped Reduction:*** new reduction that is assigned to the new emotion word in the word list
  - ***Valence***: Valence of the new emotion word to be entered in the word list (positive/negative/neutral)
- If not applicable, the entry can be left blank (just press Enter)
- After each entry, you are asked whether the input is correct [1 = Yes] or not and should be entered again [0 = No].
- If a new entry is added to one or more of the word lists,

- ET automatically checks the input file in the background to see whether the new word appears there and recodes if necessary
- the old list is moved to the **ET1.0\lists\archiv** folder


*Figure 1. After starting ET*

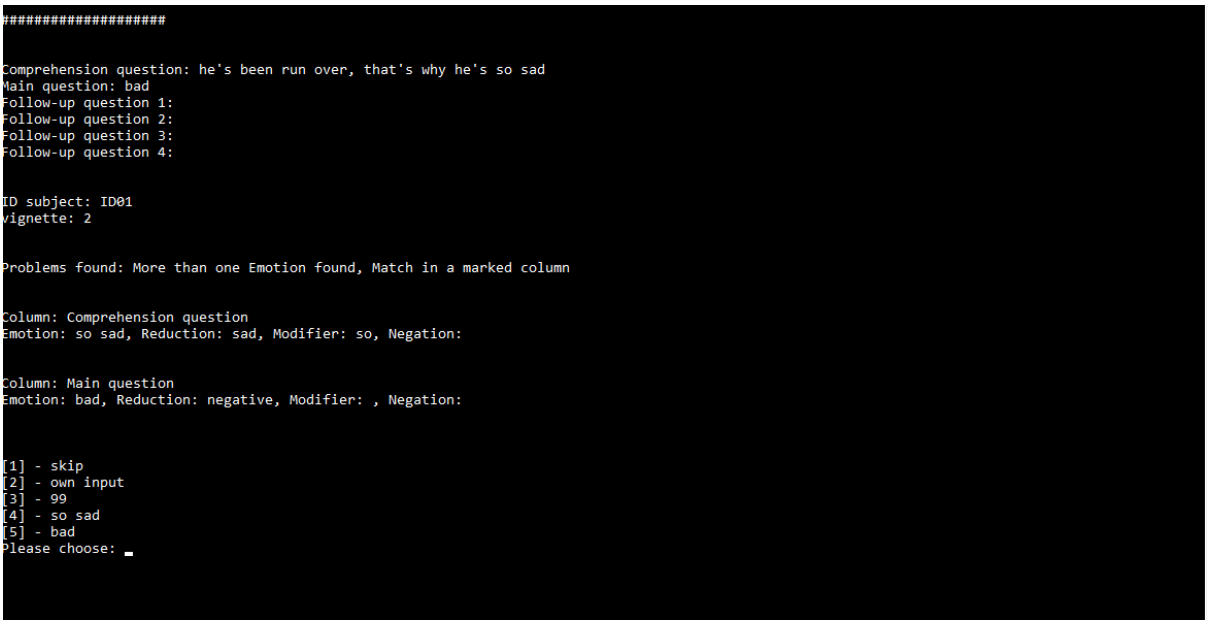


*Figure 2. Manual coding of items*

## 4. Notes on coding

### a. Problem "0":"Match in a marked column"

- Check whether the emotion word refers to the feeling of the protagonist child in the vignette or is the answer to the comprehension question
- Vignettes in which the answer to the comprehension question often contains an emotion word that does not refer to the protagonist child in the vignette are
    - 49 ("What does Nina/Tobias think of her/his friend's bike?)
    - 51 ("What does Sarah/Phil think about the shell?")
- However, in vignette 13 ("Why is Tamara/Julian's head red?"), the protagonist child's feeling is often already mentioned in the vignette (e.g., "because it's embarrassing").
- If the emotion word in the answer to the comprehension question describes the protagonist child's feeling in the vignette, this can be coded as an emotion word (This case may also occur under problem 2)

### b. Problem "2":"More than one Emotion found"

- select the most specific emotion word that the child mentions in question 1 to follow-up question 3 (e.g., sad > bad; guilty > sad)
- if an emotion word at the same level of specificity is mentioned in follow-up questions 1-3 (e.g., good and not bad or good and great), the first mentioned word is coded
- if several emotion words at the same level of specificity are mentioned in question 1 (e.g., sad and anxious or sad and angry), emotion word in follow-up question 4 should be coded
- if follow-up question 4 is not been asked in such a case, the first emotion word mentioned is coded

### c. Problem "3":"No Emotions were found

- check whether the participant child's answers contain an emotion word that is not yet in the word list, select "own input" and enter the emotion word, the reduction and, if necessary, the modifier and negation
- check in emotion_words.csv to see if there is already a suitable reduction and use this (ATTENTION: emotion_words.csv must be closed, otherwise ET cannot make a new entry there)
- select [3] 99, if there is no emotion word