

Aufgabe 04

Finale Abgabe bis 18.12 mit Präsentationen des Zwischenstandes vorher

Beispiele für Funksprüche

- “speed bird eight seven alfa sierra identified climb flight level two three zero”. The expected output would be “BAW87AS”.

Here the callsign was in the very first words.

- “gruess gott austrian triple seven sierra identified climb flight level two three zero”. The expected output would be “AUA777S”.

Here the callsign was not in the very first words.

- “bonjour air france eighty seven double alfa identified climb flight level two three zero”. The expected output would be “AFR87AA”. The numbers are not correctly pronounced (eighty seven said and not eight seven)

Beispiele für Funksprüche

- “easy foxtrot six alfa identified” or “easy jet foxtrot six alfa identified” The expected output would be in both cases “EZYF6A”. Some three letter codes code have multiple combinations of word sequences.

Beispiele für Funksprüche

- “speed bird twenty nine seventeen standby lufthansa four double alfa after air france taxi via november november eight to delta four eight one”.

The expected output would be “BAW2917” and “DLH4AA”. “air france” is not used as a callsign.

Often you have then the words “break break” or “standby” in the utterance, but you cannot rely on, e.g. when the speech recognizer did not recognize these words.

Beispiele für Funksprüche

sky travel golf hotel dobry den praha radar radar contact descend flight level one six zero

proceed papa romeo five three zero no speed restrictions

We expect TVS2GH

You need the information, which aircraft are in the air.

Part 1 of the exercise until end of November

2019-02-15__11-32-02-00:

good morning

Csgn: AUA774X BAW52CV

NO_CALLSIGN NO_CONCEPT

2019-02-15__11-32-24-00:

b_air six one identified climb level one twenty

ABP61 INIT_RESPONSE

ABP61 CLIMB 120 FL

2019-02-15__11-32-40-00:

B air six one praha radar radar contact climb flight level one two zero

Csgn: ABP61

ABP61 INIT_RESPONSE

ABP61 CLIMB 120 FL

Implement a function, which inputs such a file from disk
And implement an evaluation function, which
evaluates your performance.

Part 1 of the exercise until end of November

ReadUtteranceCheckCallsign

```
2019-02-15__11-22-40-00:
    " speed bird five two charlie victor standby break break b_air six one praha radar radar contact climb flight level
    one two zero"
Erwartet: ABP61 BAW52CV , Extrahiert: BAW52CV ABP61

2019-02-15__11-32-02-00:
    " speed bird five two bravo victor praha radar radar contact climb flight level one two zero "
Erwartet: BAW52CV , Extrahiert: NO_CALLSIGN #####

2019-02-15__11-32-24-00:
    " scandinavian one seven six seven praha radar radar contact break break speed bird five two bravo victor descend
    flight level one hundred"
Erwartet: BAW52CV (-) SAS1767 (-) , Extrahiert: SAS1767 (-) #####

2019-02-15__11-33-40-00:
    " roger"
Erwartet: NO_CALLSIGN , Extrahiert: NO_CALLSIGN

2019-02-15__11-35-24-00:
    " scandinavian one seven six seven praha radar radar contact "
Erwartet: SAS1768 , Extrahiert: SAS1767 (-) #####
Datei: '..\Data\WordSeqPlusCmdsContext2.txt'
Richtig erkannte Callsigns: 4 ( 57.1429% )
Falsch erkannte Callsigns: 1 ( 14.2857% )
Abgelehnte Callsigns: 2 ( 28.5714% )
```

Part 1 of the exercise until end of November

ReadUtteranceCheckCallsign:

parameter 1 specifies the full filename of the file you want to read in from disk (this could be e.g. the string "R:\VorlesungsUnterlagen\Betreuer\Uebungen\CodeDerAufgabenLoesungen\Aufgabe04\Data\UtterancesWithAnnotationsShort.txt"

and parameter 2 is a boolean parameter. If set to true,

- the one word filename,
- the keyword sequence and expected and
- extracted callsigns

are printed to cout.

If the function detects a deviation between extracted and expected callsign it outputs after the extracted callsign the string "####".

If the parameter is false, no output is printed to cout.

The third output parameter (new class *Evaluation* to be implemented by you) counts the number of read callsigns and also the number of correctly and wrongly extracted callsigns. If parameter 2 is true, also output these values to cout. If you extract NO_CALLSIGN, but a callsign is expected, count this as a rejection, i.e. the class also needs a rejection counter

Attention. It could

Part 1 of the exercise until end of November

The third output parameter (new class *Evaluation* to be implemented by you) counts the number of read callsigns and also the number of correctly and wrongly extracted callsigns. If parameter 2 is true, also output these values to cout. If you extract NO_CALLSIGN, but a callsign is expected, count this as a rejection, i.e. the class also needs a rejection counter Attention. It could happen that you expect only one callsign, but you extract three different ones and vice versa.