Design Description

“Bless You” – CBR Based Sneeze Detector

DVA406 - vt’15: Group 1

Table of contents

1. Introduction

In the course DVA406 Intelligent System a mini-project is included as part of the examination. The project should let you define your problem, find a solution and solve it.
The project chosen is: “Bless You” – a CBR-based Sneeze Detector.

2. Background

The current trend to analyze big data is a way to get early indications of events in the society. One such event is the outbreak of an influenza. It is imaginable that using sneeze detectors could be used to get an early indication of such an outbreak.

3. Problem High Level Description

A microphone, placed in e.g. a public library, keeps listening to the sound in the library, when it detects that someone sneezes a counter is incremented. A supervisory system is able to read the sneeze count at cyclic intervals. The read counter values can be used to detect if a flu is in progress.

4. High Level Description

The experimental “Bless You” system prototype features:

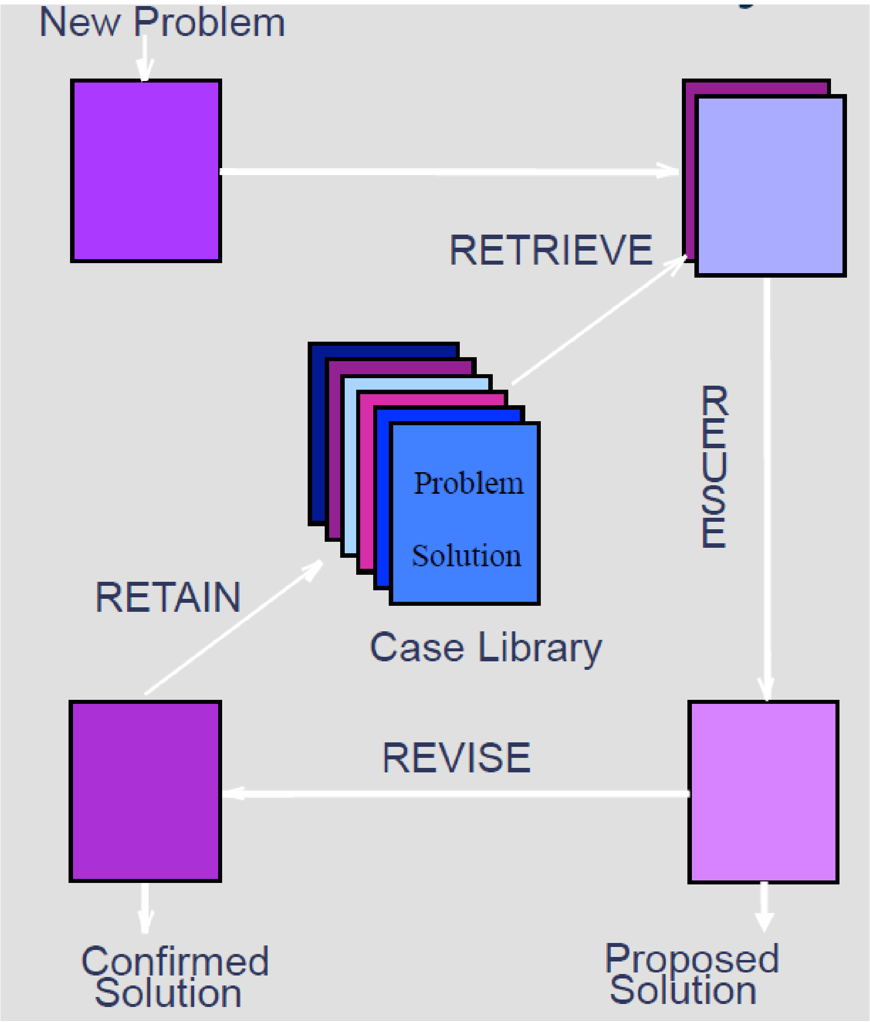
- Sound input in the form of .wav-files.
- A library of sound files of known sneezes used for CBR.
- Detection of sneezes according to a feature detection: such as FFT, Crest Factor, and Wave-lets.
- Response is just a detection sneeze being TRUE or FALSE and an incremented count.

Block Diagram, CBR System

In the figure below:

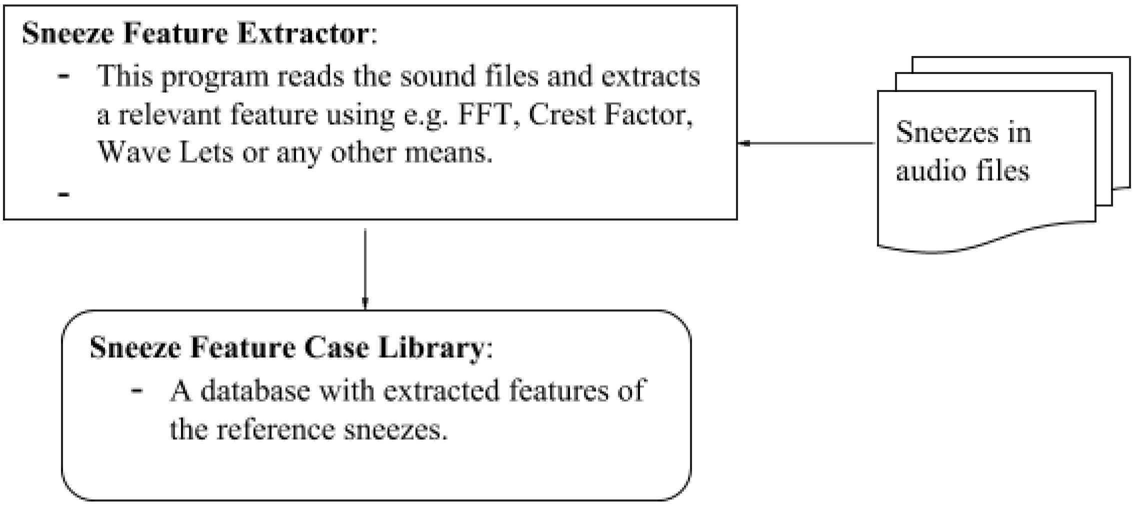
“New Problem” = a new sneeze sound file.

“Case Library” = sneeze features



Block Diagram, Feature Detector

The audio files in the library are analyzed for usable features.



Sneeze Feature Extractor:

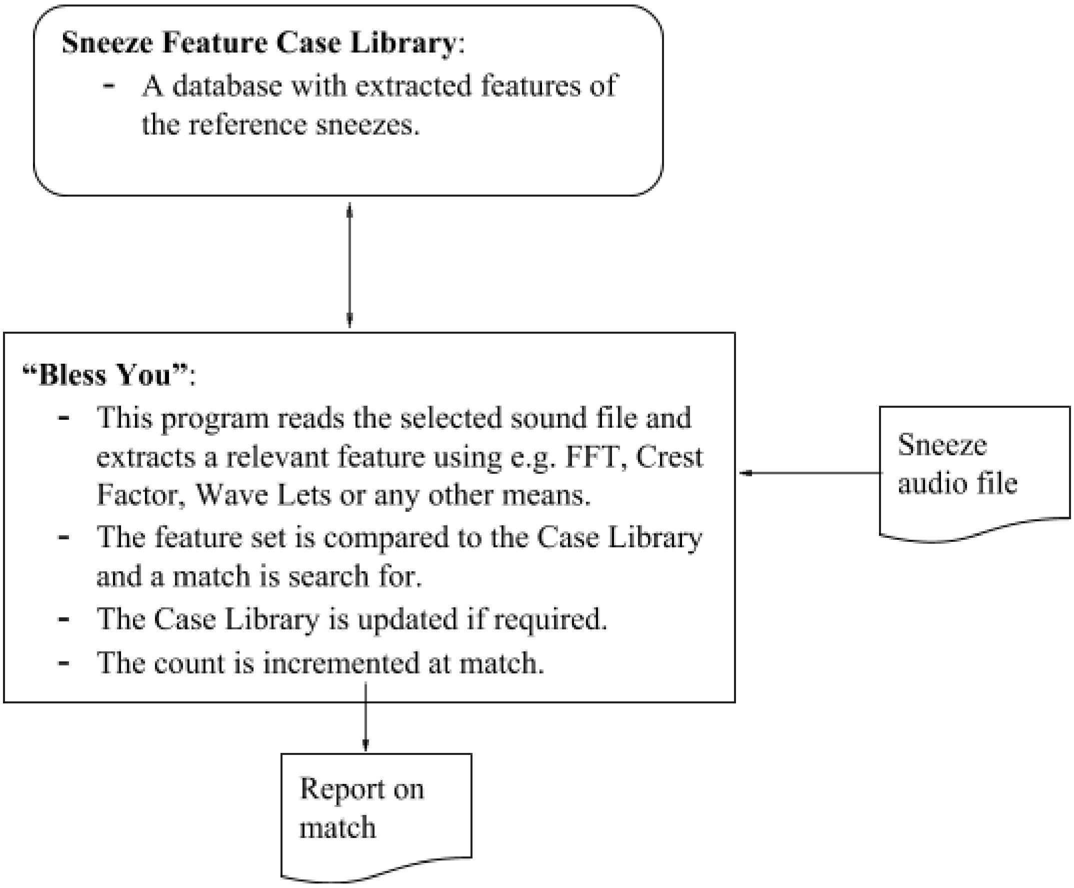
- This program reads the sound files and extracts a relevant feature using e.g. FFT, Crest Factor, Wave Lets or any other means.
-

Sneeze Feature Case Library:

- A database with extracted features of the reference sneezes.

Block Diagram, CBR evaluation

An audio file is evaluated by CBR.



Sneeze Feature Case Library:

- A database with extracted features of the reference sneezes.

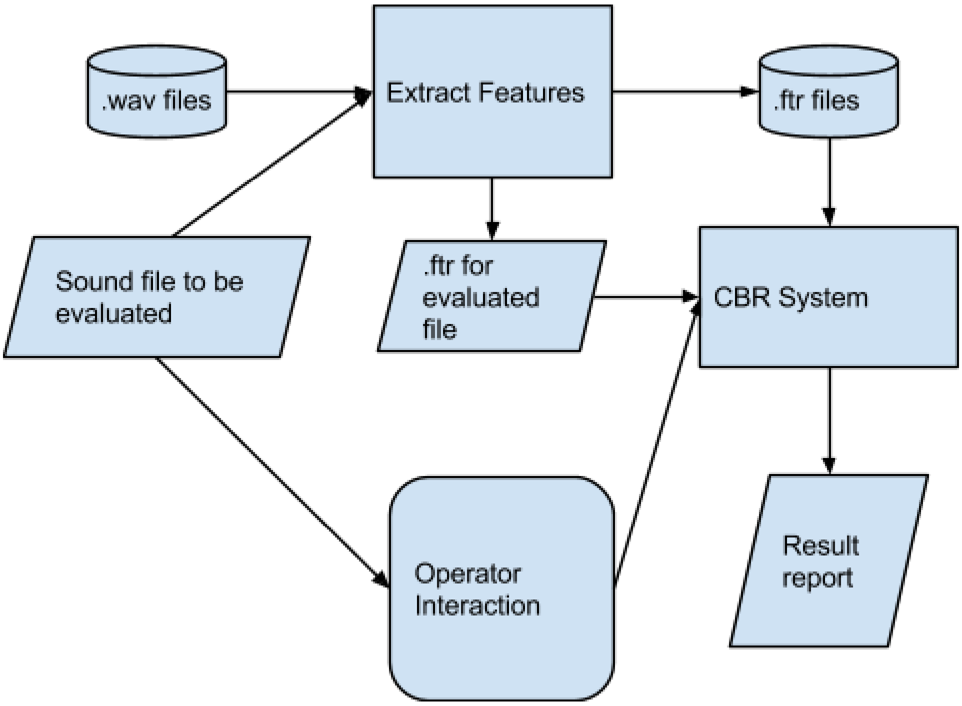
“Bless You”:

- This program reads the selected sound file and extracts a relevant feature using e.g. FFT, Crest Factor, Wave Lets or any other means.
- The feature set is compared to the Case Library and a match is search for.
- The Case Library is updated if required.
- The count is incremented at match.

5. Work Flow

5.1. Overview

Overview of the complete system, it consists of two programs working in collaboration via sets of files. Sound files in `.wav`-formats are used for feature extraction, the result is stored as feature files in `.ftr`-format.
These files are used by the CBR System case library together with a sound file to evaluate if it contains a sneeze or not.



5.2. Extract Features Design

Operations to be performed on each .wav-file.

Command line program: `BlessYou.exe`
Usage:

```
BlessYou P1 [P2]
```

where

P1 = name of text file with names of all .wav-files to be examined
P2 = path to directory for created `.ftr`-files (optional)

Inputs: .wav-files, 16bit, 44kHz
Output: created `.ftr`-files.

Format of list file used as P1: one line per .wav-file:

line = <marker for type of sound> TAB [<path>]<filename of .wav-file>
<path> = <absolute path> | <relative path to directory of the list file itself>
<marker for type of sound> = ‘0’ if not a sneeze sound
‘1’ if a sneeze sound

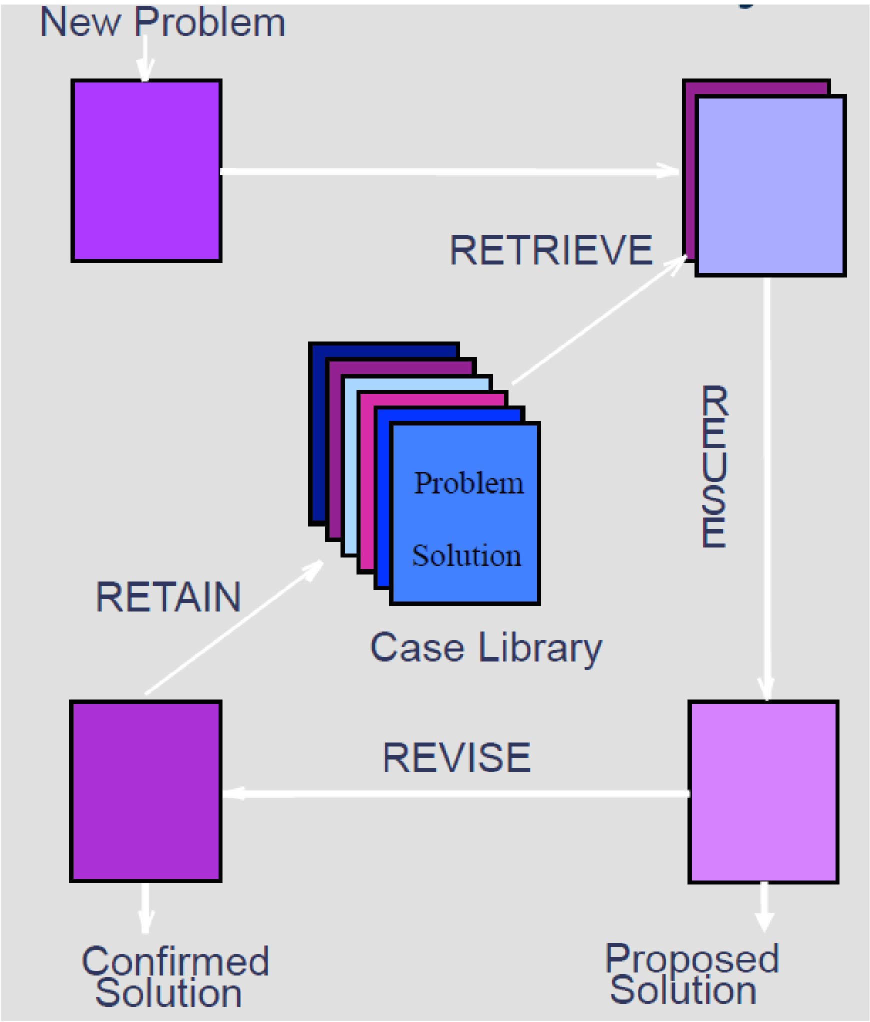
‘?’ if unknown contents.

Flow of computations for each case, i.e. .wav-file.

1. Read .wav file contents (16 bit PCM, 44 kHz)
2. If stereo: calculate sample as average of left and right sample.
3. Normalize: search for largest sample, scale all samples so that the largest sample is 100% of 16 bits.
4. Search for start of possible sneeze: search for a sample with a absolute magnitude of at least 50%. TBA
5. Evaluate length of suspected sneeze, check for a low level 20% TBA after at least 0.3 seconds.
6. Normalize to standard length, L, e.g. 1 second TBA, split into N (e.g. 10/seconds TBA) equal time interval, indexed as t = [0, N-1]
7. The range of samples to be evaluated are now, evaluated time is: 3 seconds TBA
   20% before TBA
   80% after TBA
8. Now the feature extraction can be made, calculated as a number, e.g. 32 TBA, of float values or vector of float values, e.g.:
   a. Peak values
   b. Mean values
   c. RMS values
   d. Peak to peak values
   e. CF (Crest Factor) values
   f. PZ (Passage through Zero> values
   g. FFT values (vector)
   h. Wavelet values (vector)
9. Suggested Feature Vector: [EOlsson76 p.29, equation 2.33]
   FV = [Peak(x), Mean(x), RMS(x), CF(x)]
   where x is a vector of time-based samples .
10. Save the extracted feature values to a file with the same file name as the main file name of the .wav-file. TBA

5.3. CBR System Design

Input: path to directory of `.ftr`-files.
Input Path and file name of sound file to be evaluated:
Output: report on evaluated file being a sneeze or not.



Notes on figure:

New Problem:
New `.wav`-file is read and the features extracted and stored in a `.ftr`-file.

Retrieve:
Match the current `.ftr`-file against the cases in the library using the similarity measurement “Nearest Neighbor” and create a list with the best matches.

[EOlsson76 p.32, equation 2.34]

$$Similarity(N,R) = \sum_{i=1}^{n} w_i * f(N_i, R_i)$$

N is the new case
R is the retrieved case
n is the number of features in each case
i is the current feature
f is the similarity function for attribute i in cases N and R
w is the weight that controls the importance of the attribute i, the sum of all w_i is 1.

Reuse:

Inspect the best matches and see how many of them are sneezes and how many are not, then propose a solution based on the results.

Revise:
If the proposed solution does not match the confirmed solution it needs to revised and hopefully improve the detection system in the process. When an incorrect solution is proposed, the triggering input file is to be added to the case library. This will assist in the classification of similar cases in the future. The weight values for the feature extraction systems may also be modified to help future classification.

Retain:
Evaluate the significance of all samples and possible replace any with less usagfullness.

Proposed Solution:
TBA

Confirmed Solution:
TBA

5.4. *Feature Vector*

The feature vector contains a set of different features, e.g.:
FV = [PeakColumn[t], MeanColumn[t], RMSColumn[t], CFColumn[t], PZColumn[t]], which characterizes the case, where t is an index according to above.

Calculation of feature vector

Input:	Sound sample:	`double sampleArr[]`
	Size of Sound sample:	`int size`
	time interval index:	`int timeIntervalIx`
	time interval size:	`int timeIntervalSize`

The current interval in `sampleArr` under evaluation is `timeIntervalIx * timeIntervalSize` ... `(timeIntervalIx + 1) * timeIntervalSize -1`

Algorithm

```
PeakColumn[t] = max( | sampleArr[current interval] | )
MeanColumn[t] = average( | sampleArr[current interval] | )
RMSColumn[t] = rms( sampleArr[current interval] )
CFColumn[t] = cf( | sampleArr[current interval] | )
```

cf is calulated as S_{max} /rms value.

```
PZColumn[t] = pz( | sampleArr[current interval] | )
```

pz is calulated as number of times zero is passed between samples..

Output
Feature vector, FV.

5.5. *Similarity Functions and Weight values*

The similarity function is according to:

$$Similarity(N,R) = \sum_{i=1}^{n} w_i * f(N_i,R_i)$$

It is used to calculate the best fitting case from the case library to the current sample under investigation.

```
sf   =  f($N_t$, $R_t$)
$N_t$ / $R_t$ = feature value at interval t
PeakColumn  sf = SUMV(| $N_t$ - $R_t$ |) / N
MeanColumn  sf = SUMV(| $N_t$ - $R_t$ |) / N
RMSColumn   sf = SUMV(| $N_t$ - $R_t$ |) / N
CFColumn    sf = SUMV(| $N_t$ - $R_t$ |) / N
```

SUMV is the sum over all t indexes.
N is the number of time intervals.

5.6. Case Base Library maintenance

The purpose of maintenance is to continuously update the library so that the cases stored are the most useful. Keeping too many cases in the library will make the matching slow. One way to prune obsolete cases, i.e. case that seldom are matched, another way is to merge similar cases into a single case.
Details TBA

6. System Performance Evaluation

Initially the system is used to evaluate different methods to extract features. Use the details below to evaluate how well a specific system works.
A specific system is:

i. A selected Feature Vector
ii. A selected Similarity Function.

a. Set up a specific system with a multiple case files that are marked as containing sneeze or not.
b. Set up methods to extract features and compare them.
c. Execute the system:
    i. For each case file (F_x) in the set of all case files.
    ii. Set up the CASE library for the rest of the case files but file (F_x).
    iii. Evaluate the selected case file (F_x) against the system.
    iv. Generate report:
        file (F_x) is in fact a sneeze/none-sneeze
        a. file (F_x) is detected as sneeze with a probability of p_1%
        b. file (F_x) is detected as none-sneeze with a probability of p_2%
        c.
d. Calculate performance values for the current system.
    i. Calculate best limits for p_1 and p_2 to evaluate:

ii. V_{TS} % of correctly detected sneezes (True Sneezes) - search Max

iii. V_{FS} % of incorrectly detected sneezes (False Sneezes) - search Min

iv. V_{TN} % of correctly detected none-sneezes (True Sneezes) - search Max

v. V_{FN} % of incorrectly detected none-sneezes (False None-Sneezes) - search Min

e. Calculate a single performance value for the current system.

1. $V_{TOT} = w_1 (V_{TS} + V_{TN})/2 - w_2 (V_{FS} + V_{FN})/2$

ii. Suggestion:

1. $w_1 = 75\%$ $w_2 = 25\%$

7. Feature Vector Evaluation

Evaluate NNCnt (e.g. 5 TBA) closest neighbors using the calculated similarity value to:

Is Sneeze:	number of closest Sneeze neighbors / NNCnt
Is None-Sneeze:	number of closest None-Sneeze neighbors / NNCnt

8. DUMMY

9. References

Ref.	Document Title
EOlsson76	Fault Diagnosis of Industrial Machines using Sensor Signals and CASE-based Reasoning Doctoral Dissertation nr. 76, MdH Västerås, Sweden 2009

History

Date	Version	Status	Notes	Author
2015-02-18	1	working	Document introduced.	All
2015-02-19	1	working	ch.5 reworked	All
2015-02-25	1	dratft	First draft	All