**Data structures and functions for frame scoring**

**Assumption:**

The following data structures already exist:

* masterframes, a structure in which the frame structures used in this particular application are specified;
* three cell arrays of frames, which have been constructed in the decoding loop over all test instances:
  + predicted\_frames
  + unambiguous\_ref\_frames
  + ambiguous\_ref\_frames (optional)

**Function 1: instances\_for\_confmats = collect\_instances\_for\_confmats(masterframes, predicted\_frames, unambiguous\_ref\_frames, ambiguous\_ref\_frames)**

In this function, all the instances on the basis of which the confusion matrices are made, are assembled:

instances\_for\_confmats:

* .num\_instances
* .frame\_level
  + .class\_labels{class\_index}
  + .predicted\_classes{instance\_num}
  + .predicted\_class\_indices{instance\_num}
  + .unambiguous\_ref\_classes{instance\_num}
  + .unambiguous\_ref\_class\_indices{instance\_num}
* .slotvalue\_level
  + .frame(frame\_num)
    - .frame\_name
    - .slot\_names
    - .slot(slot\_num)
      * .slot\_name
      * .class\_labels{class\_index}
      * .predicted\_classes{instance\_num}
      * .predicted\_class\_indices{instance\_num}
      * .unambiguous\_ref\_classes{instance\_num}
      * .unambiguous\_ref\_class\_indices{instance\_num}
      * .ambiguous\_ref\_classes{instance\_num}
      * .ambiguous\_ref\_class\_indices{instance\_num}

For each instance, the class label indices in all relevant frames (predicted\_frames, ambiguous\_ref\_frames and unambiguous\_ref\_frames) are assembled, both on frame level and on slot value level.

The structure ‘instances\_for\_confmats’ is the input for the functions that create the confusion matrices.

The ‘ambiguous\_ref\_frame’ in the input arguments is optional. If it is empty, no “ambiguous\_ref\_values” fields are made.

**Function 2: confmat\_structs = make\_confmats(instances\_for\_confmats)**

This function produces a cell array of confusion matrices (confmats), which are structured as follows:

confmats{confmat\_num}:

* .frame\_level
  + .class\_labels{class\_index}
  + .confmat\_global(class\_index\_reference, class\_index\_predicted)
  + .confmats\_per\_instance(class\_index\_reference, class\_index\_predicted, instance\_num)
* .slotvalue\_level
  + .frame(frame\_num)
    - .frame\_name
    - .slot\_names
    - .slot(slot\_num)
      * .slot\_name
      * .class\_labels {class\_index}
      * .confmat\_global(class\_index\_reference, class\_index\_predicted)
      * .confmats\_per\_instance(class\_index\_reference, class\_index\_predicted, instance\_num)

Within function 2, two sub-functions are used:

**Function 2a: confmat\_struct = make\_confmat\_unambiguous(instances\_for\_confmat)**

**Function 2b: confmat\_struct = make\_confmat\_ambiguous(instances\_for\_confmat, confmat\_struct\_unambiguous)**

Structure of function 2:

confmats{1} = make\_confmat\_unambiguous(instances\_for\_confmats);

if isfield(instances\_for\_confmats.slotvalue\_level.frame(1).slot(1), 'ambiguous\_ref\_classes')

confmats{2} = make\_confmat\_ambiguous(instances\_for\_confmats, confmats{1});

end

**Function 3: scores\_structs = compute\_scores(confmat\_structs, Fbeta)**

This function makes a cell array of scores (‘scores’) on the basis of a cell array of confusion matrices (‘confmats’). So for each confusion matrix in the cell array, a scores structure is created. Each ‘scores’ structure looks as follows:

.scores\_structs{scores\_num}:

* .frame(frame\_num)
  + .frame\_name
  + .frame\_level\_scores
    - .num\_positives\_reference
    - .num\_positives\_prediction
    - .num\_true\_positives
    - .precision
    - .recall
    - .Fscore
  + .slot\_level\_scores
    - .num\_classifications
    - .num\_filled\_reference
    - .num\_filled\_prediction
    - .num\_empty\_reference
    - .num\_empty\_prediction
    - .num\_correct\_filled
    - .num\_correct\_empty
    - .num\_correct
    - .precision
    - .recall
    - .Fscore
    - .accuracy
    - .errorrate
    - .truenegrate
  + .slotvalue\_level\_scores
    - .sum\_positives\_reference
    - .sum\_positives\_prediction
    - .sum\_true\_positives
    - .micro\_precision
    - .micro\_recall
    - .micro\_Fscore
    - .macro\_precision
    - .macro\_recall
    - .macro\_Fscore
  + .binary\_slot\_filling\_scores
    - .num\_classifications
    - .num\_filled\_reference
    - .num\_filled\_prediction
    - .num\_empty\_reference
    - .num\_empty\_prediction
    - .num\_correct\_filled
    - .num\_correct\_empty
    - .num\_correct
    - .micro\_precision
    - .micro\_recall
    - .micro\_Fscore
    - .macro\_precision
    - .macro\_recall
    - .macro\_Fscore
    - .accuracy
    - .errorrate
    - .truenegrate
  + .slot(slot\_num)
    - .slot\_name
    - .slot\_level\_scores
      * .num\_classifications
      * .num\_filled\_reference
      * .num\_filled\_prediction
      * .num\_empty\_reference
      * .num\_empty\_prediction
      * .num\_correct\_filled
      * .num\_correct\_empty
      * .num\_correct
      * .precision
      * .recall
      * .Fscore
      * .accuracy
      * .errorrate
      * .truenegrate
    - .slotvalue\_level\_scores
      * .sum\_positives\_reference
      * .sum\_positives\_prediction
      * .sum\_true\_positives
      * .micro\_precision
      * .micro\_recall
      * .micro\_Fscore
      * .macro\_precision
      * .macro\_recall
      * .macro\_Fscore
    - .binary\_slot\_filling\_scores
      * .num\_classifications
      * .num\_filled\_reference
      * .num\_filled\_prediction
      * .num\_empty\_reference
      * .num\_empty\_prediction
      * .num\_correct\_filled
      * .num\_correct\_empty
      * .num\_correct
      * .micro\_precision
      * .micro\_recall
      * .micro\_Fscore
      * .macro\_precision
      * .macro\_recall
      * .macro\_Fscore
      * .accuracy
      * .errorrate
      * .truenegrate
    - .slotvalue(slotvalue\_num)
      * .value
      * .slotvalue\_level\_scores
        + .num\_positives\_reference
        + .num\_positives\_prediction
        + .num\_true\_positives
        + .precision
        + .recall
        + .Fscore
* .global
  + .frame\_level\_scores
    - .num\_classifications
    - .num\_correct
    - .accuracy
    - .errorrate
  + .slot\_level\_scores
    - .num\_classifications
    - .num\_filled\_reference
    - .num\_filled\_prediction
    - .num\_empty\_reference
    - .num\_empty\_prediction
    - .num\_correct\_filled
    - .num\_correct\_empty
    - .num\_correct
    - .precision
    - .recall
    - .Fscore
    - .accuracy
    - .errorrate
    - .truenegrate
  + .slotvalue\_level\_scores
    - .sum\_positives\_reference
    - .sum\_positives\_prediction
    - .sum\_true\_positives
    - .micro\_precision
    - .micro\_recall
    - .micro\_Fscore
    - .macro\_precision
    - .macro\_recall
    - .macro\_Fscore
  + .binary\_slot\_filling\_scores
    - .num\_classifications
    - .num\_filled\_reference
    - .num\_filled\_prediction
    - .num\_empty\_reference
    - .num\_empty\_prediction
    - .num\_correct\_filled
    - .num\_correct\_empty
    - .num\_correct
    - .micro\_precision
    - .micro\_recall
    - .micro\_Fscore
    - .macro\_precision
    - .macro\_recall
    - .macro\_Fscore
    - .accuracy
    - .errorrate
    - .truenegrate

**Wrapper function: stats = collect\_stats\_for\_frames(Fbeta, masterframes, predicted\_frames, unambiguous\_ref\_frames, ambiguous\_ref\_frames)**

(collects the stats for all the instances)

stats.instances\_for\_confmats = collect\_instances\_for\_confmats(masterframes, predicted\_frames, unambiguous\_ref\_frames, ambiguous\_ref\_frames);

stats.confmat\_structs = make\_confmats(stats.instances\_for\_confmats);

stats.scores\_structs = compute\_scores(stats.confmat\_structs, Fbeta);

**And a wrapper around the previous wrapper function, to include the stats for specific reference frames:**

**stats = collect\_all\_stats(Fbeta, masterframes, predicted\_frames, unambiguous\_ref\_frames, ambiguous\_ref\_frames)**

(collects the stats for all instances, but also separate stats for instances with specific reference frames)

num\_frames = length(masterframes.framenames);

num\_instances = length(predicted\_frames);

for frame\_num = 1:num\_frames

instance\_numbers\_with\_ref\_frame{frame\_num} = [];

for instance\_num = 1: num\_instances

if strcmp(unambiguous\_ref\_frames{instance\_num}.thisframe, masterframes.framenames{frame\_num})

instance\_numbers\_with\_ref\_frame{frame\_num}(end+1) = instance\_num;

end

end

end

stats.all\_instances = collect\_stats\_for\_frames(Fbeta, masterframes, predicted\_frames, unambiguous\_ref\_frames, ambiguous\_ref\_frames);

for frame\_num = 1:num\_frames

stats.ref\_frame{frame\_num} = collect\_stats\_for\_frames(Fbeta, masterframes,

predicted\_frames(instance\_numbers\_with\_ref\_frame{frame\_num}), unambiguous\_ref\_frames(instance\_numbers\_with\_ref\_frame{frame\_num}), ambiguous\_ref\_frames(instance\_numbers\_with\_ref\_frame{frame\_num}));

end