Hierarchical classification for ImageCLEF 2008 Medical Image Annotation

Thomas Deselaers, Jayashree Kalpathy-Cramer, Henning Müller, and Thomas M. Deserno
http://www.imageclef.org/2008/medaat

The IRMA coding system consists of four independent axes with three to four positions, each in $\{0, \ldots, 9, a, \ldots, z\}$, where '0' denotes '*unspecified*' to determine the end of a path along an axis. The technical code (T) describes the imaging modality; the directional code (D) models body orientations; the anatomical code (A) refers to the body region examined; and the biological code (B) describes the describes the biological system examined.

The code is strictly hierarchical – each sub-code element is connected to only one code element. The element to the right is a sub element of the element to the left.

For example:

```
2   cardiovascular system
21  cardiovascular system; heart
216 cardiovascular system; heart; aortic valve
```

The aortic valve is an element of the heart, which in turn is an element of the cardiovascular system.

Let an image be coded by the above 4 *independent* axes, such that we can consider the axes independently and just sum up the errors for each axis independently:

Let $l_1^I = l_1, l_2, \ldots, l_i, \ldots, l_I$ be the *correct* code (for one axis) of an image.

Let $\hat{l}_1^I = \hat{l}_1, \hat{l}_2, \ldots, \hat{l}_i, \ldots, \hat{l}_I$ be the *classified* code (for one axis) of an image.

Where l_i is specified precisely for every position, and in $\hat{l}_i$ it is allowed to say "*don't know*", which is encoded by `*`.

Note that I (the depth of the tree to which the classification is specified) may be different for different images.

Given an incorrect classification at position $\hat{l}_i$ we consider all succeeding decisions to be wrong and given a not specified position, we consider all succeeding decisions to be not specified.

Furthermore, we do not count any error if the correct code is unspecified and the predicted code is a wildcard. In that case, we do consider all remaining positions to be not specified.

We want to penalize wrong decisions that are easy (fewer possible choices at that node) over wrong decisions that are difficult (many possible choices at that node), we can say, a decision at position l_i is correct by chance with a probability of $\frac{1}{b_i}$ if b_i is the number of possible labels for position i. This assumes equal priors for each class at each position.

Furthermore, we want to penalize wrong decisions at an early stage in the code (higher up in the hierarchy) over wrong decisions at a later stage in the code (lower down on the hierarchy) (i.e. l_i is more important than l_{i+1}).

Putting together:

$$\sum_{i=1}^{I} \underbrace{\frac{1}{b_i}}_{(a)} \underbrace{\frac{1}{i}}_{(b)} \underbrace{\delta(l_i, \hat{l}_i)}_{(c)}$$

with

$$\delta(l_i, \hat{l}_i) = \begin{cases} 0 & \text{if } l_j = \hat{l}_j \quad \forall j \leq i \\ 0.5 & \text{if } l_j = * \quad \exists j \leq i \\ 1 & \text{if } l_j \neq \hat{l}_j \quad \exists j \leq i \end{cases}$$

where the parts of the equation account for

- **(a)** accounts for difficulty of the decision at position i (branching factor)
- **(b)** accounts for the level in the hierarchy (position in the string)
- **(c)** correct/not specified/wrong, respectively.

In addition, for every axis, the maximal possible error is calculated and the errors are normed such that a completely wrong decision (i.e. all positions for that axis wrong) gets an error count of 0.25 and a completely correctly predicted axis has an error of 0. Thus, an image where all positions in all axes are wrong has an error count of 1, and an image where all positions in all axes are correct has an error count of 0.

Examples:

```
correct: 318a

classified  error count
318a        0.0
318*        0.0244653860094
3187        0.0489307720188
31*a        0.0824574121058
31**        0.0824574121058
3177        0.164914824212
3***        0.34342152954
32**        0.686843059079
1000        1.0
```