Alpha Beta Information Loss

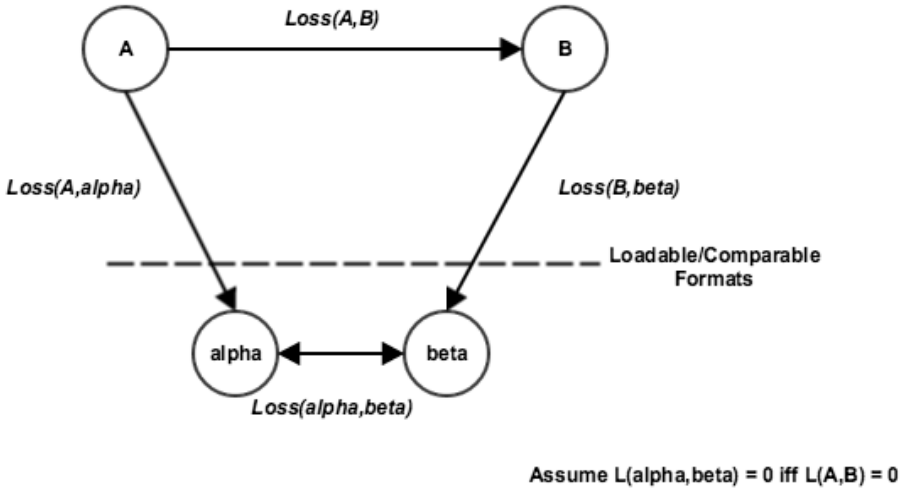
The method described in McHenry 2009 (pdf) does not scale as it:

- requires a dataset exemplifying the data distribution of an archive,
- requires a dataset made up of file types that can be directly opened for the before and after comparison,
- and requires significant computation to fill in the weights of the I/O-graph.

We propose an alternative approach that can accommodate an unknown sample set and whose computation can be distributed out over time (i.e. with each incoming job request). The approach works as follows:

- Keep a registry of file types that can be directly opened by the comparison tool(s)
- For each job request converting from format A to format B
  - Find a format alpha that can be reached from A that is within the set of loadable formats
  - Find a format beta that can be reached from B that is within the set of loadable formats
  - If both alpha and beta exist carry out the conversions from A to alpha and B to beta
  - Compare the files of type alpha and beta. If the difference between alpha and beta is below some threshold record this edge as a good edge within the I/O-graph

The above algorithm assumes that a conversion to alpha and beta resulting in any information loss incurred from the conversion from A to B being ***undone*** is ***HIGHLY UNLIKELY*** (proof required).



We implement this means of measuring information loss as follows:

- In PolyglotStewardAMQ create a new method convertWithLoss(...) that carries out the above algorithm,
  - [x] **BD-1313** - Create convertWithLoss method in PolyglotStewardAMQ BLOCKED
    - Modify Polyglot.java to add a function convertWithLoss(...) that calls convert by default
    - Create a list of loadable formats by the comparison tools
    - Call the DAP with a conversion request to alpha (make sure this request doesn't also attempt information loss estimation),
      - [x] **BD-1317** - Add a flag to /convert endpoint in Polyglot that disables infomration loss estimtion BLOCKED
    - Call the DAP with a conversion request to beta (make sure this request doesn't also attempt information loss estimation)
    - Add comparison tools to DTS, [x] ~~**BD-1300**~~ - Versus Extractor clean up DONE
    - Call the DTS with an extraction request for alpha
    - Call the DTS with an extraction request for beta
    - Add helper methods descriptor_set_distance and descriptor_distance (porting from https://opensource.ncsa.illinois.edu/bitbucket/projects/BD/repos/bdcli/browse/bd.py), [x] **BD-1314** - Add helper methods descriptor_set_distance and descriptor_distance BLOCKED
    - Use descriptor_set_distance to compare extracted JSON from alpha and beta, if a match is found mark edge as good in I/O-graph (e.g. 1 vs 0)
      - Save as a record in mongo document in the form: Application, A, B, 0/1
- Add code and flag to PolyglotStewardAMQ.conf to load edge weights from mongo on Polyglot load,
  - [x] **BD-1315** - Add code and flag to PolyglotStewardAMQ.conf to load edge weights from mongo on Polyglot load BLOCKED
- Add endpoint to PolyglotRestlet.java that uses edge weights to determine best path,
  - [x] **BD-1316** - Add endpoint to PolyglotRestlet.java that uses edge weights to determine best path BLOCKED