EN-Deep ML-Process Scenario File Format

The description of a machine learning / natural language processing process / experiment:

- consists of **Tasks** (derived classes of the Task class)
- is specified in a scenario text file by the individual task clauses, whose input and output files specify the dependencies among them
- assumes to be launched in one working directory, where all the input data are located and the output data are saved --> file descriptions do not contain paths in general

Task clause

Each task clause consists of the following fields:

`task [id];` -- unique task id

`algorithm: [class];` -- full name of the used Java class (must be derived from en_deep.mlprocess.Task)

`params: [name]=["value"], [...];` -- a list of parameters required by the particular task

`in: ["file1"] [...];` -- list of input files / wildcard patterns

`out: ["file1"] [...];` -- list of output files / wildcard patterns

Example:

```
# this is a commentary
task sttoarff; # id of the subtask
    algorithm: # used Java class
        en_deep.mlprocess.manipulation.StToArff;
    params: lang_conf="st-en.conf", # its settings
        divide_senses, one_file,  # parameters with no value (binary)
        generate="Children, DepPath, HeadPos", # parameters
        cluster_file="clusters-plain.txt";     # with values set
    in: "train.txt"; # inputs listing
    out: "train.arff"; # outputs listing
end;
```

Parameters

- the `params` clause is not compulsory (no parameters will be set)
- the parameters' values are not compulsory (they will be considered boolean, empty string will be substituted for their value) and may be enclosed in quotes to be able to contain spaces

Wildcards

- inputs and outputs may contain wildcards, which expand to file names – possibly the whole task is expanded to many tasks for the individual files.

Expanding mode “*”

- „*“ is for finding the files with the same expansion for every occurrence in the inputs, e.g. „data-*.dat“ and „params-*.dat“ expand to „data-exp1.dat“ „params-exp1.dat“ / „data-

exp2.dat“ „params-exp2.dat“

- with sub-selection, there's a possibility of matching multiple patterns and combining them

- for every expansion (combination), one task will be created
- „*“ is propagated into outputs and inputs of depending tasks (transitively)

Listing mode “*”

- “**” in the inputs of a task lists all the files that match as the inputs to this very task
  - “**” and “*” may be combined: first, “**” is applied, then “*”
- “**” in the outputs means that the task produces a previously unknown number of outputs

Sub-Selection (using more variables)

- every wildcard may contain a sub-selection specification, using “|” characters:
- the sub-specification may contain several variables ($0-$9), which leads to a Cartesian product expansion
  - e.g. “file-*|$0-$1|.dat” (additional variables) matches everything that starts with “file-”, ends with “.dat” and contains “-” in between. If other patterns repeat the $0-$9 variables, they must match the same strings.
- $0 is reserved for the listing mode, $1-$9 are used as expanding mode wildcards
- $1-$9 must be used in consecutive order from right to left, no number may be skipped
- “*” on the output of a task expands to all valid variables joined with a “_” (e.g. “$1_$2_$3” if $1-$3 are valid).
- if “*” is combined with $0-$9 and the task doesn't depend on another expanded task, “*” is equivalent to “$1”. If the task depends on other tasks, “*” is equivalent to all valid variables joined with a “_”.