NAME

trainer
identifier

SYNOPSIS

./trainer **[TRAIN D] [STORE_TRAIN P D]**
./identifier **[TRAIN D] [TEST P D] [STORE_TEST P D]**

[] = files **containing** directories to **TEXT CORPORA**
[] = **PATH** and **FILENAME** where the directories and **PROFILES should be stored**

DESCRIPTION

trainer	- generates the profiles for every possible text corpus
identifier	- generates profiles for the test files, then compares them with training profiles

OPTIONS

inside Identifier.cpp

print()	- printing without distance
print_withDist()	- printing with distance

DATA

/data/

test_data/…	- contains 21 text files with ~4kb and 1 empty file
train_data/…	- contains 20 text files with ~10kb
test_dir.txt	- contains the directories of all *test* files mentioned above
train_dir.txt	**-** contains the directories of all *train* files mentioned above

BUGS

GenerateProfile.hpp

There are a few characters which could not be filter out and causing a new line. So it will bring some calculation failures in the IdentifyLanguage.hpp and more lines in the profile then it should be (6 of 20 languages had this problem). There are still just too few of these characters to make a difference at the end of the calculation (still all 20 languages of 20 correctly determined by the program).

.prof

There are non representing characters uni, bi and trigrams the profiles, because languages like Chinese need more than a char size to show correctly, even a string is not enough.

COMPILERS TESTED

Mac OSX g++
Windows cl from Visual Studio

SOURCES

For the test corpus: Wikipedia Politik and Wikipedia Politologie (given languages)
For the train corpus: Wikipedia Oper and Wikipedia Opera (given languages)

William B. Cavnar and John M. Trenkle. N-Gram-Based Text Categorization, Environmental Research Institute of Michigan
—> https://www.let.rug.nl/vannoord/TextCat/textcat.pdf

AUTHOR

Le Duyen Sandra Vu