

PAN 2014 Author Profiler - podręcznik użytkownika

Jacek Kowalski, Paweł Lipski

12 maja 2014

1 Pobieranie kodu

```
git clone https://github.com/tilius/author-profiler.git
```

Kod był uruchamiany pod wersją 2.7.3 Pythona.

2 Instalacja bibliotek obliczeniowych

2.1 LIBLINEAR

Bibliotekę można pobrać z <http://www.csie.ntu.edu.tw/~cjlin/liblinear/> (sekcja Download LIBLINEAR).

Po rozpakowaniu wystarczy wydać polecenie make. Powstały plik wykonywalne train oraz predict należy umieścić np. w /usr/local/bin (tak, aby były one dostępne przez PATH) pod następującymi nazwami:

- train → linear-train
- predict → linear-predict

2.2 LIBSVM

Domyślnie aplikacja wykorzystuje LIBLINEAR. Użycie LIBSVM do obliczeń można wymusić poprzez przekazanie opcji `--libsvm` do polecenia train.py.

Bibliotekę można pobrać z <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> (sekcja Download LIBSVM).

Po rozpakowaniu wystarczy wydać polecenie make, a następnie umieścić trzy pliki wykonywalne (svm-train, svm-scale oraz svm-predict) np. w /usr/local/bin (tak, aby były one dostępne przez PATH).

3 Instalacja NLTK

Zgodnie z <http://www.nltk.org/install.html>, zakładając, że w systemie jest dostępny pip, należy uruchomić z shella:

```
sudo pip install -U numpy
sudo pip install -U pyyaml nltk
```

Następnie należy ściągnąć dane NLTK (dokładniej Part Of Speech Tagger). Po wejściu w interaktywną powłokę Pythona należy wpisać:

```
import nltk
nltk.download("maxent_treebank_pos_tagger")
```

4 Uruchamianie

4.1 Trenowanie

```
train.py [-h] -i corpus_dir -o model_dir [--disjoint] [--libsvm]
```

Zakłada się, że `corpus_dir` jest katalogiem, w którym znajdują się pliki XML zgodne z formatem przewidzianym dla PAN 2014. Upřednio utworzone przez `train.py` pliki w `model_dir` zostaną nadpisane.

Niestandardowa opcja `--disjoint` uruchamia trenowanie w trybie osobnego trenowania i klasyfikacji dla płci oraz wieku. Odpowiedni obiekt klasyfikatora wraz z wszystkimi niezbędnymi rezultatami zostanie w takim przypadku zapisany do folderu `model_dir`. Z tego względu nie przekazuje się tej opcji do `classify.py`. Niestandardowa opcja `--libsvm` wykorzystuje bibliotekę LIBSVM (zamiast domyślnej LIBLINEAR). Również tutaj nie ma potrzeby przekazywania tej opcji do `classify.py` - informacja o bibliotece zostanie zaszyta w plikach modelu.

4.2 Klasyfikacja

```
classify.py [-h] -i corpus_dir -m model_dir -o output_dir [--truth truth_file]
[--accuracy accuracy_output_file]
```

Zakłada się, że `corpus_dir` jest katalogiem, w którym znajdują się pliki XML zgodne z formatem przewidzianym dla PAN 2014. Upřednio utworzone przez `classify.py` pliki w `output_dir` zostaną nadpisane.

Niestandardowa (i nieobowiązkowa) opcja `--truth truth_file` wskazuje plik zawierający faktyczne wyniki klasyfikacji - przyjmujemy format zgodny z tym używany w plikach PAN2013. Taki plik można też wygenerować za pomocą skryptu `scripts/make-truth.sh`, podając w parametrze katalog - wynik zostanie zapisany do pliku `truth.dat` w tym katalogu. Niestandardowa (i nieobowiązkowa)

opcja `--accuracy accuracy_output_file` wskazuje plik, do którego zostanie zapisana linia z informacją o trafności klasyfikacji (używane w skryptach).

5 Wyniki

Wszystkie testy były uruchamiane na Ubuntu 12.04 wirtualizowanym w VMware Player na Windows 7, na procesorze i7 przy włączonym VT-X.

Styl rodzaj danych (fragment korpusu treningowego PAN14), na jakich uruchamiane były testy

N_{total} całkowity rozmiar danego fragmentu PAN14

N_{train}, N_{test} rozmiar korpusu treningowego i testowego, wydzielonych z wybranego fragmentu PAN14

N_{maj} rozmiar klasy większościowej w obrębie korpusu testowego

linear użyto biblioteki LIBLINEAR

svm użyto biblioteki LIBSVM

joint trenowanie i klasyfikacja były przeprowadzane dla klas będących kombinacją płci i wieku

disjoint trenowanie i klasyfikacja były przeprowadzane oddzielnie dla płci i wieku

Styl	N_{total}	N_{train}, N_{test}	N_{maj}	Konfiguracja	Dokładność	Czas
Blog	147	75	17	linear, joint	28.00% (21/75)	1677 sec
				linear, disjoint	26.67% (20/75)	2833 sec
				svm, joint	16.00% (12/75)	1695 sec
				svm, disjoint	16.00% (12/75)	2711 sec
Reviews	4470	200	39	linear, joint	16.00% (32/200)	316 sec
				linear, disjoint	12.00% (24/200)	544 sec
				svm, joint	12.50% (25/200)	319 sec
				svm, disjoint	11.50% (23/200)	543 sec
		1000	133	linear, joint	28.30% (283/1000)	1590 sec
				linear, disjoint	28.90% (289/1000)	2895 sec
				svm, joint	13.30% (133/1000)	1683 sec
				svm, disjoint	13.30% (133/1000)	2912 sec
		2000	261	linear, joint	36.60% (732/2000)	3565 sec
				linear, disjoint	34.05% (681/2000)	6212 sec
				svm, joint	(skipped)	N/A
				svm, disjoint	(skipped)	N/A
Social media	7746	150	39	linear, joint	58.67% (88/150)	1536 sec
				linear, disjoint	56.00% (84/150)	2622 sec
				svm, joint	26.00% (39/150)	3257 sec
				svm, disjoint	25.33% (38/150)	5646 sec

6 Wnioski

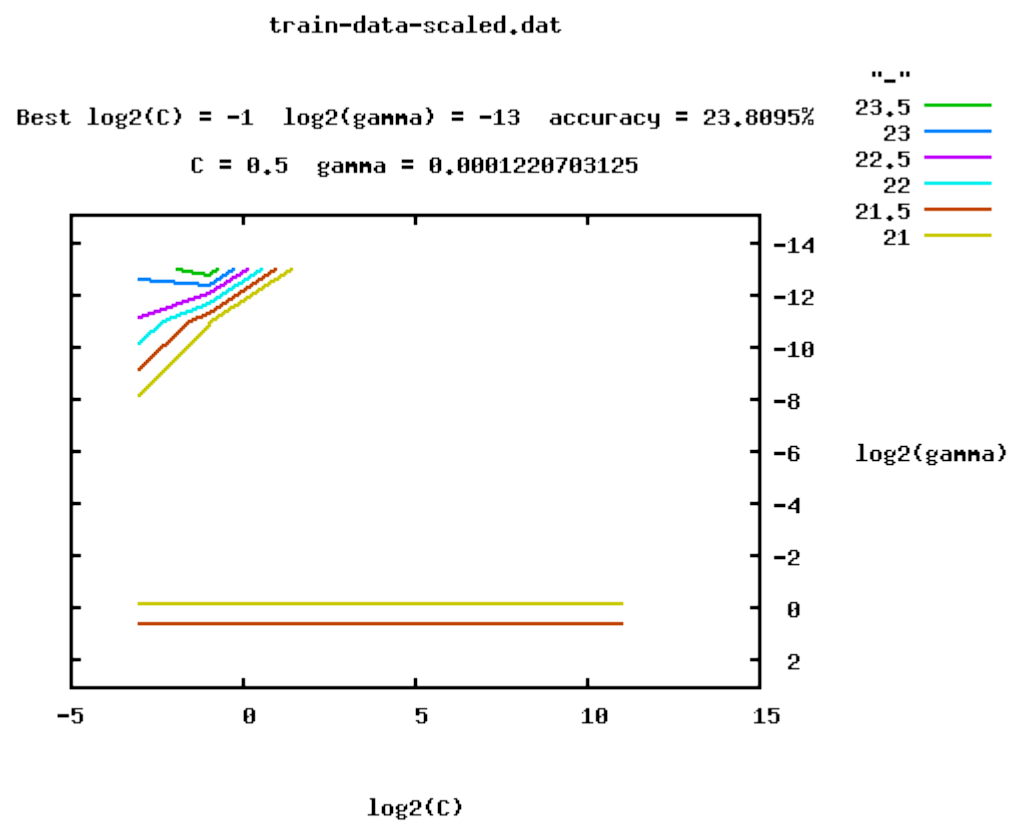
Użycie LIBLINEAR przynosiło zdecydowanie lepsze rezultaty niż użycie LIB-SVM. Ta druga w niektórych przypadkach zwracała dla wszystkich klasyfikowanych obiektów klasę większościową, a w jeszcze innych jej dokładność była nawet mniejsza niż naiwnej klasyfikacji wg klasy większościowej.

Nie miało większego znaczenia dla dokładności, czy trenowanie i klasyfikacja były przeprowadzane dla klas będących kombinacją płci i wieku, czy oddzielnie dla płci i wieku. W tym drugim przypadku jednak znacznie więcej czasu pochłaniały obliczenia, które przeprowadzać trzeba było dla obu wielkości oddzielnie.

Zdecydowanie największą część czasu zajmowały obliczenia (tagowanie części mowy) wykonywane przez NTLK.

TODO - coś o tym, co się udało

7 Wykresy



Rysunek 1: Wykres zależności dokładności od parametrów C oraz γ