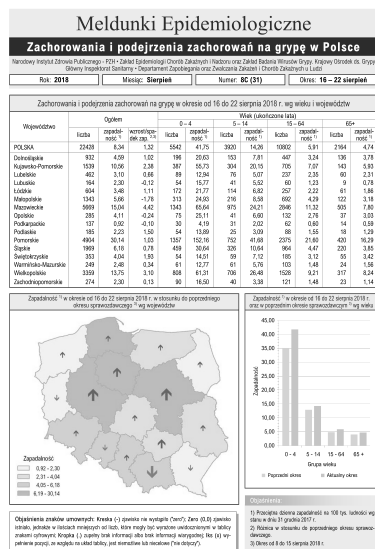


# Cierpienia zbieracza danych

Piotr KRZYŻANOWSKI\*

\*Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski



Pierwsza strona cotygodniowego raportu o zachorowaniach na grypę z NIZP-PZH

Jest kwiecień 2019 roku. Tej zimy grypa przeszła bez rozgłosu; ciekawe, jak wyglądała zachorowalność w poprzednich latach? Oczywiście, najprościej bezpośrednio zwrócić się do ludzi, którzy takie informacje zbierają, np. w Pracowni Monitorowania i Analizy Sytuacji Epidemiologicznej w Zakładzie Epidemiologii Chorób Zakaźnych i Nadzoru w Narodowym Instytucie Zdrowia Publicznego – Państwowym Zakładzie Higieny. Ale co zrobić, jeśli nie chcielibyśmy zwracać im głowy swoją skromną osobą i nagabywać o udostępnienie części bazy danych, a potem – gdy już się zgodzą (a na pewno się zgodzą, bo to bardzo mili ludzie) – zanudzać ich pytaniami o to, jak w tej bazie dostać się do tego fragmentu, który jest nam potrzebny... Może jednak lepiej na początek spróbować samodzielnie wykorzystać to, co i tak już Pracownia publikuje na swoich stronach internetowych?

I rzeczywiście, pod adresem [wwold.pzh.gov.pl/oldpage/epimeld/index\\_p.html](http://wwold.pzh.gov.pl/oldpage/epimeld/index_p.html) znajduje się przebogaty zestaw bieżących i archiwalnych raportów o zachorowaniach na różne okropne choróbka, w tym to, które nas interesuje:

[wwold.pzh.gov.pl/oldpage/epimeld/grypa/index.htm](http://wwold.pzh.gov.pl/oldpage/epimeld/grypa/index.htm)

Bingo! Tam jest archiwum obejmujące szczegółowe, pięknie opracowane i na bieżąco uzupełniane raporty: od roku 2000, po 48 rocznie, czyli w sumie bez mała tysiąc! Ze względu na liczbę będziemy musieli interesujące nas dane wydobyć maszynowo. Jednak jest pewien mały...

...kłopot: raporty są w plikach PDF. To nic strasznego – można się pocieszyć, pamiętając, że wokół jest wiele programów pozwalających wyluskać czysty tekst z pliku w formacie PDF. Faktycznie, po ściągnięciu któregoś z raportów, np. [G\\_18\\_08C.pdf](#), wystarczy wyciągnąć z niego tekst linuksowym programem `pdftotext G_18_08C.pdf`.

Ponieważ raporty mają skomplikowaną strukturę – składają się z ramek i złożonych tabel – efekt eksportu jest daleki od naszych oczekiwań (pokazujemy niewielki urywek pliku złożonego z 1500+ linii):

32	Warmińsko–Mazurskie
33	Wielkopolskie
34	Zachodniopomorskie
35	
36	22428
37	932
38	1539
39	462
40	164
41	604
42	1343

Możemy jednak zauważyć, porównując wynik z oryginałem w PDF, że w linii 36. jest dokładnie ta jedyna liczba, którą chcemy wyciągnąć z raportu: to całkowita liczba zachorowań w danym tygodniu. Niestety, po ściągnięciu kilkunastu innych raportów musimy przyznać się do...

...porażki: ta liczba nie pojawia się zawsze w tej samej linii. A skoro tak, to skąd mamy wiedzieć, gdzie jej szukać? Może będzie to największa z liczb zapisanych w raporcie? – wszak inne wartości to dane cząstkowe, dotyczące podziału zachorowań na województwa lub na grupy wiekowe. Nie można jednak wykluczyć, że gdzieś w raporcie znajdzie się – na przykład dla porównania – większa liczba, dotycząca zeszłorocznych zachorowań... Co więc z tym fantem zrobić?

Możemy spróbować obrócić na swoją korzyść fakt, że raport ma piękną, systematyczną (choć skomplikowaną) strukturę. Interesująca nas liczba zawsze pojawia się na pierwszej stronie, w głównej tabeli, obok nagłówka „POLSKA”. Kilka testów – i przekonujemy się, iż w każdym raporcie komórka z całkowitą liczbą zachorowań znajduje się niemal idealnie w tym samym miejscu na stronie!

Zachorowania i podejrz.	
Województwo	liczba
POLSKA	22428
Dolnośląskie	932
Kujawsko-Pomorskie	1539

Zatem należy nauczyć się wyciągać tekst z pliku PDF, który znajduje się w zadanym obszarze strony, tylko, że...

...nie bardzo wiadomo, jakim programem to zrobić. Zanim sięgniemy po ostateczność i zaczniemy szukać mrocznych internetowych serwisów, które obiecują nam, że zrobią to za nas bezboleśnie, zauważmy, że w łatwy sposób można automatycznie wyciąć zadany fragment z obrazka (np. zdjęcia) w formacie JPG lub innym podobnym, na przykład PNG (który w przeciwieństwie do JPG nie stosuje kompresji stratnej). Gdybyśmy więc poszli nieco okreśną drogą i wykonali trzy kroki:

1. najpierw skonwertowali nasz PDF do formatu PNG,
  2. następnie obcięli plik PNG do interesującego nas obszaru zawierającego właściwą komórkę tabeli,
  3. a na koniec odczytali z pliku PNG znajdującą się w nim liczbę (tu: 22 428),
- to dalej byłoby już z górki.

Krok pierwszy jest prosty. Pod Linuxem jest wiele narzędzi konwersji „wszystkiego na wszystko” (wspomnijmy chociażby program `convert` z pakietu `ImageMagick`); po kilku eksperymentach i doczytaniu instrukcji decydujemy się na `pdftoppm`.

Krok drugi – jak na samym początku stwierdziliśmy – *powinien* być prosty. Jeśli nie `ImageMagick`, to inne narzędzie pozwoli nam z linii komend przyciąć obrazek jak trzeba. Po kilku eksperymentach i dalszym doczytaniu instrukcji okazuje się, że potrafi to... `pdftoppm`.

Krok trzeci budzi najwięcej wątpliwości, bo tym, co chcemy w nim zrobić, jest tzw. OCR – czyli maszynowe rozpoznawanie tekstu na obrazku. Zaletą jest to, że obrazek będzie zawierać tylko drukowane czcionki, w dodatku – same cyfry (żadnego odręcznego pisma, które mogłoby nas położyć na łopatkę), a wadą – że zupełnie nie wiemy, jak skuteczne są linuksowe programy w takich zastosowaniach. Po przejrzaniu paru stron internetowych wybieramy aplikację `tesseract` (podobno jedną z najlepszych).

Mamy więc następujący ciąg technologiczny:

```
1 pdftoppm -l 1 -png -r 900 -x 1578 -y 2850 -W 486 -H 160 G.18.08C.pdf >
  report.png
2 tesseract report.png report
```

Skąd wiedziałem, jakie podać parametry przycięcia? – wystarczyło na pliku pośrednim `report.png` użyć narzędzia zaznaczania w `GIMP`-ie.

Pierwsza linia załatwia nam za jednym zamachem dwa punkty planu, konwertując jedynie pierwszą stronę (opcja `-l 1`) pliku PDF do formatu PNG i następnie przycinając go jak należy. Druga linia wyprodukuje plik tekstowy `report.txt`, w którym znajduje się nasza liczba (i jakiś śmieć, którego na szczęście łatwo się pozbyć). Niestety, po ściągnięciu kilkunastu raportów ze strony `NIZP-PZH` okazuje się, że...

...`tesseract` nie umie odczytać prawidłowo wszystkich liczb. Między innymi niektóre cyfry „7” odczytuje jako „/”; bywa też, że niektóre cyfry gubi (co skądinąd do dziś budzi moje zdumienie: w drukowanym, wyrazistym i prostym tekście taka głupia wpadka?). Więc może coś mniej wyrafinowanego będzie skuteczniejsze w odczytaniu prostej liczby? Wypróbujmy zatem kolejne linuksowe narzędzie, tym razem będzie to program `gocr`. Po lekturze manuala cieszymy się, że może on współpracować z plikami w formacie PPM, natywnymi dla `pdftoppm`. Lecz po przetestowaniu kolejnych kilkunastu raportów o grypie...

...z przerażeniem stwierdzamy, że `gocr` też nie umie odczytać prawidłowo wszystkich liczb. Na przykład, „0” miesza mu się z literą „O”... Na szczęście, po żmudnej wzrokowej inspekcji wyników wygląda na to, że jest to jedyny jego problem – dla nas zupełnie nieistotny, bo przecież wiemy, że gdzie `gocr` widzi „O”, tam *musi* być zero. W ten sposób silnik maszynki jest gotowy: wystarczy jeszcze dobudować pętlę ładującą kolejne raporty z archiwum `WWW` – i dane o grypie będą nasze!

Być może opisane przeszkody wkrótce znikną. `NIZP-PZH` realizuje wielki grant, `EpiBaza`, którego celem jest udostępnienie zasobów Ogólnopolskiego Systemu Nadzoru Epidemiologicznego i Środowiskowego nad Bezpieczeństwem Ludności. Być może pojawi się wygodne API na potrzeby takich zadań, jak opisane powyżej? Pewnie wtedy stare archiwum zniknie z serwera i niniejszy artykuł zostanie jedynym dowodem na to, jak trudne mogło być kiedyś życie zbieracza danych (o grypie).

API (*application programming interface*): ściśle zdefiniowany sposób komunikacji (zestaw komend, język zapytań itp.) umożliwiający „rozmowę” między programami komputerowymi, które nawet nie muszą znać kodów źródłowych swoich interlokutorów.