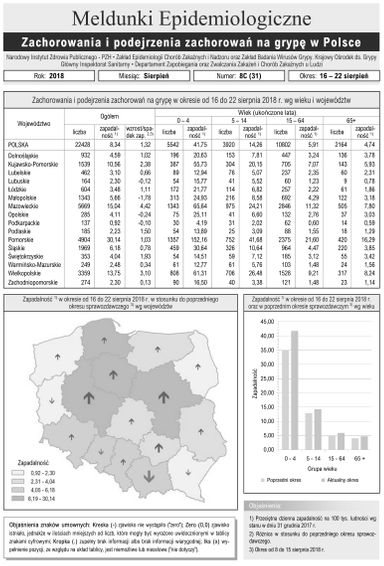
Cierpienia zbieracza danych

*Piotr KRZYŻANOWSKI**

* Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski

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 – Państwowy Zakład Higieny. Ale co zrobić, jeśli nie chcielibyśmy zawracać 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?



Meldunki Epidemiologiczne

Zachorowania i podejrzenia zachorowań na grypę w Polsce

Narodowy Instytut Zdrowia Publicznego - PZH • Zakład Epidemiologii Chorób Zakaźnych i Nadzoru oraz Zakład Badania Wirusów Grypy, Krajowy Ośrodek ds. Grypy
Główny Inspektorat Sanitarny • Departament Zapobiegania oraz Zwalczania Zakażeń i Chorób Zakaźnych u Ludzi

Rok: **2018** | Miesiąc: **Sierpień** | Numer: **8C (31)** | Okres: **16 – 22 sierpień**

Zachorowania i podejrzenia zachorowań na grypę w okresie od 16 do 22 sierpnia 2018 r. wg wieku i województw

Województwo	Ogółem			Wiek (ukończone lata): 0 – 4		5 – 14		15 – 64		65+	
	liczba	zapadalność [1]	wzrost/spadek zap. [1]	liczba	zapadalność [1]	liczba	zapadalność [1]	liczba	zapadalność [1]	liczba	zapadalność [1]
POLSKA	22428	8,34	1,32	5542	41,75	3920	14,26	10802	5,91	2164	4,74
Dolnośląskie	932	4,59	1,02	196	20,63	153	7,61	447	3,24	136	3,78
Kujawsko-Pomorskie	1539	10,56	2,38	387	55,73	304	20,15	705	7,07	143	5,93
Lubelskie	462	3,10	0,66	89	12,94	76	5,07	237	2,35	60	2,31
Lubuskie	164	2,30	-0,12	54	15,77	41	5,52	60	1,23	9	0,78
Łódzkie	604	3,48	1,11	172	21,77	114	6,82	257	2,22	61	1,86
Małopolskie	1343	5,66	-1,78	315	24,93	216	8,58	692	4,29	122	3,18
Mazowieckie	5669	15,04	4,42	1343	65,64	975	24,21	2646	11,22	505	7,80
Opolskie	285	4,11	-0,24	75	25,11	41	6,60	132	2,76	37	3,03
Podkarpackie	137	0,82	-0,10	30	4,19	31	2,02	62	0,60	14	0,59
Podlaskie	185	2,23	1,50	54	13,89	25	3,09	88	1,55	18	1,29
Pomorskie	4904	30,14	1,03	1357	152,16	752	41,68	2375	21,60	420	16,29
Śląskie	1969	6,18	0,78	459	30,64	326	10,64	964	4,47	220	3,85
Świętokrzyskie	353	4,04	1,93	54	14,51	59	7,12	185	3,12	55	3,42
Warmińsko-Mazurskie	249	2,48	0,34	61	12,77	61	5,76	103	1,48	24	1,56
Wielkopolskie	3359	13,75	3,10	808	61,31	706	26,48	1526	9,21	317	8,24
Zachodniopomorskie	274	2,30	0,13	90	16,50	40	3,38	121	1,48	23	1,14

Zapadalność [1] w okresie od 16 do 22 sierpnia 2018 r. w stosunku do poprzedniego okresu sprawozdawczego [2] wg województw

Zapadalność [1] w okresie od 16 do 22 sierpnia 2018 r. oraz w poprzednim okresie sprawozdawczym [2] wg wieku

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

Wystarczy jednak wejść pod adresem `wwwold.pzh.gov.pl/oldpage/epimeld/index_p.html` i znajduje się przebogaty zestaw bieżących i archiwalnych raportów o zachorowaniach na różne okropne choróbska, w tym to, które nas interesuje:

`wwwold.pzh.gov.pl/oldpage/epimeld/grypa/index.htm`

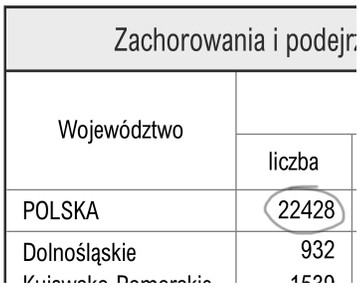
Bingo! Tam jest archiwum obejmujące szczegółowe, pięknie opracowane i na bieżąco uzupełniane raporty: od roku 2000, po 48 rocznie, czyli w sumie bez mała tysiąc! Ze względu na liczbę będziemy musieli interesujące nas dane wydobyć maszynowo. Jednak jest pewien mały…

…kłopot: raporty są w plikach PDF. To nic strasznego – można się pocieszyć, pamiętając, że wokół jest wiele programów pozwalających wyłuskać 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 tabelek – 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…



Zachorowania i podejr

Województwo	liczba
POLSKA	22428
Dolnośląskie	932
Kujawsko-Pomorskie	1539

…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!

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 obiecają 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 okrężną 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 Linuksem 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 łopatki), a wadą – że zupełnie nie wiemy, jak skuteczne są linuksowe programy w takich zastosowaniach. Po przejrzeniu 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
```

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...

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

... `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 pętle ładujące kolejne raporty z archiwum WWW – i dane o grypie będą nasze!

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.

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).