1. Software

W opracowaniu skryptów wykorzystano język programowania R w wersji 3.6.3 (https://www.r-project.org/), zarówno do ekstrakcji treści stron internetowych (web scraping), jak również do analizy semantycznej wyodrębnionych treści. Jako zintegrowane środowisko programistyczne wykorzystano R Studio w wersji 1.2.5033 (https://rstudio.com/).

a) W ramach zadania ekstrakcji treści ogłoszeń z portali internetowych, w szczególności wykorzystano następujące pakiety R:

- 'rvest' w wersji 0.3.6, kod źródłowy pobrany z repozytorium CRAN opracowany przez Hadley Wickham (2020). Pakiet zawiera szeroki zestaw narzędzi do pozyskiwania informacji ze stron internetowych (http://rvest.tidyverse.org/)

- 'purrr' w wersji 0.3.4, kod źródłowy pobrany z repozytorium CRAN opracowany przez Lionel Henry i Hadley Wickham (2020). Pakiet rozszerza możliwości R związane z tzw. programowaniem funkcyjnym (opartym na funkcjach) poprzez zapewnienie kompletnego i spójnego zestawu narzędzi do pracy z funkcjami i wektorami. Kluczowa z punktu widzenia zadania jest rodzina funkcji map(), które pozwalają zastąpić wiele pętli kodem, który jest zarówno bardziej zwięzły jak i łatwiejszy do odczytania.

b) W ramach zadania analizy semantycznej zgromadzonych treści, konieczne było doinstalowanie następujących pakietów:

- 'tidytext' w wersji 0.2.5, kod źródłowy pobrany z repozytorium CRAN opracowany przez Julia Silge i in. (2020). Korzystanie z uporządkowanych danych może ułatwić wiele zadań związanych z wydobyciem tekstu, m.in. zwiększyć jego skuteczność i spójność z narzędziami, które są już w powszechnym użyciu. Takich narzędzi dostarcza pakiet 'tidytext'. Duża część infrastruktury potrzebnej do eksploracji tekstu z uporządkowanymi arkuszami danych (data frame) w R istnieje już w pakietach takich jak 'dplyr', 'broom', 'tidyr' i 'ggplot2'. W tym pakiecie udostępniono funkcje i wspierające zestawy danych, które umożliwiają konwersję tekstu do i z formatów tidy oraz płynne przełączanie się między narzędziami tidy a istniejącymi pakietami text mining.

c) Ponadto konieczne było wykorzystanie dodatkowych pakietów, usprawniających pracę w ramach obu zadań:

- 'dplyr' w wersji 0.7.8. Pakiet usprawnia manipulację dużymi zbiorami danych.

- 'wordcloud' w wersji

- ‘RColorBrewer’ w wersji

d) do ekstrakcji sekcji kompetencje z portalu LinkedIn posłużono się narzędziami Selenium opartmi na automatyzacji działania przeglądarki internetowej (https://www.selenium.dev/). W szczególności wykorzystano implementację działającą w środowisku r (pakiet ‘rselenium’)

e) do ekstrakcji treści stron opartych na JavaScript (pracuj.pl) posłużono się narzędziem do renderowanie www Docker – splash. Dokumentacja techniczna dotycząca instalacji i uruchomienia narzędzia znajduje się pod adresem: https://splash.readthedocs.io/en/stable/

d) W celu identyfikacji selektorów Kaskadowych Arkuszy Stylów (CSS selectors) wykorzystano narzędzie SelectorGadget (www.selectorgadget.com) dla przeglądarki Google Chrome. SelectorGadget jest narzędziem open source, które sprawia, że generowanie i odkrywanie CSS na skomplikowanych stronach jest bardzo proste. Po zainstalowaniu rozszerzenia Chrome, a następnie przejściu do dowolnej strony, w prawym dolnym rogu strony otworzy się okienko. Po kliknięciu na interesujący element strony SelectorGadget wygeneruje wtedy minimalny selektor CSS dla tego elementu. Selektor taki jest następnie wykorzystany w skrypcie R do ekstrakcji treści z pola na stronie www odpowiadającemu jego nazwie.

e) Tam gdzie identyfikacja poszczególnych pól była niemożliwa, pracowano bezpośrednio na kodzie strony internetowej z wykorzystaniem narzędzie developerów dla przeglądarek Google Chrome (wersja 94.0.4606.81, 64-bitowa) i Mozilla Firefox (wersja 93.0, 64-bitowa).

f) Instalacja oprogramowania odbyła się na komputerze z zainstalowanym 64 bitowym systemem operacyjnym Windows 10 Pro.

Szczegółowy raport wszystkich niezbędnych pakietów, ich wersji i oprogramowania znajduje się poniżej:

R version 4.1.0 (2021-05-18)

Platform: x86_64-w64-mingw32/x64 (64-bit)

Running under: Windows 10 x64 (build 19042)

Matrix products: default

locale:

[1] LC_COLLATE=Polish_Poland.1250 LC_CTYPE=Polish_Poland.1250 LC_MONETARY=Polish_Poland.1250 LC_NUMERIC=C

[5] LC_TIME=Polish_Poland.1250

system code page: 1252

attached base packages:

[1] tcltk stats graphics grDevices utils datasets methods base

other attached packages:

[1] ggraph_2.0.5 igraph_1.2.6 wordcloud_2.6 RColorBrewer_1.1-2 SnowballC_0.7.0 tm_0.7-8

[7] NLP_0.2-1 forcats_0.5.1 tidyr_1.1.3 ggplot2_3.3.4 tidytext_0.3.1 dplyr_1.0.6

[13] magick_2.7.2 splashr_0.6.0 purrr_0.3.4 rvest_1.0.0

loaded via a namespace (and not attached):

[1] ggrepel_0.9.1 Rcpp_1.0.6 lubridate_1.7.10 lattice_0.20-44 assertthat_0.2.1 digest_0.6.27

[7] utf8_1.2.1 ggforce_0.3.3 slam_0.1-48 R6_2.5.0 httr_1.4.2 pillar_1.6.1

[13] rlang_0.4.11 curl_4.3.1 rstudioapi_0.13 Matrix_1.3-3 labeling_0.4.2 readr_1.4.0

[19] stringr_1.4.0 selectr_0.4-2 htmlwidgets_1.5.3 polyclip_1.10-0 munsell_0.5.0 compiler_4.1.0

[25] janeaustenr_0.1.5 pkgconfig_2.0.3 askpass_1.1 stevedore_0.9.3 htmltools_0.5.1.1 openssl_1.4.4

[31] tidyselect_1.1.1 gridExtra_2.3 tibble_3.1.2 graphlayouts_0.7.1 viridisLite_0.4.0 fansi_0.5.0

[37] crayon_1.4.1 withr_2.4.2 MASS_7.3-54 grid_4.1.0 jsonlite_1.7.2 gtable_0.3.0

[43] lifecycle_1.0.0 magrittr_2.0.1 formatR_1.11 scales_1.1.1 tokenizers_0.2.1 HARtools_0.0.5

[49] cli_2.5.0 stringi_1.6.1 farver_2.1.0 viridis_0.6.1 xml2_1.3.2 ellipsis_0.3.2

[55] generics_0.1.0 vctrs_0.3.8 tools_4.1.0 glue_1.4.2 tweenr_1.0.2 hms_1.1.0

[61] parallel_4.1.0 colorspace_2.0-1 tidygraph_1.2.0

2. Hardware

Skrypty były uruchamiane na komputerze wyposażonym w 6-rdzeniowy procesor AMD Ryzen 5600X, 16GB pamięci RAM, kartę graficzną Radeon RX 6600XT oraz dysk SSD 1TB. Do działania wymagany jest stały szerokopasmowy dostęp do Internetu. Z uwagi na duże obciążenie łącza podczas automatycznego otwierania linków i pobierania danych, szybkość transferu istotnie wpływa na czas wykonania zadania.