23.05.2022, Kraków

Narzędzia do stylometrii (analiza stylu artykułów – analiza Autorstwa) z wykorzystaniem pakietu stylo w języku R

P. P.
P. S.

Cel projektu

Celem projektu była analiza narzędzi do stylometrii. Projekt skonstruowano przy pomocy pakietu “stylo” w języku R. Narzędzia stworzone z wykorzystaniem ww. paczki potrafią dokonywać różnych analiz z zakresu stylistyki obliczeniowej, przypisywania autorstwa itp.

Zakres pracy

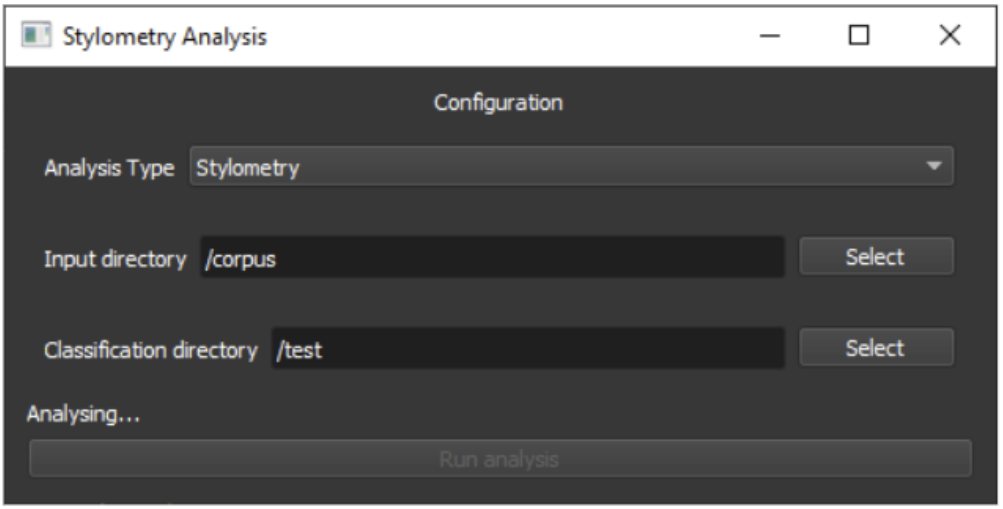
W zakres prac potrzebnych do zrealizowania założonego celu wchodziło dobranie odpowiednich tekstów różnych autorów i próba przeprowadzenia na nich eksperymentu. W projekcie wykorzystaliśmy głównie książki polskiego autorstwa. Jest to 31 książek, 5-ciu autorów. Aby projekt był bardziej przyjazny dla użytkownika końcowego, opakowaliśmy paczkę “stylo” w wygodny do użytkowania interfejs graficzny. Dzięki temu użytkownik ma możliwość szybkiej zmiany konfiguracji początkowych parametrów aplikacji. Dodatkowo w zakres prac wchodzi przedstawienie w postaci wykresów (głównie statystycznych) rezultatów analizy na wybranych przez użytkownika tekstach.

Opis rozwiązania

Pakiet “stylo” posiada wiele różnych opcji, jednak dla kogoś bez zaplecza technicznego użytkowanie w takiego narzędzia może wydawać się problematyczne. Dlatego po wstępnej analizie możliwości pakietu “stylo”, zdecydowaliśmy się na skorzystanie z graficznej biblioteki PyQt, aby ułatwić użytkownikowi korzystanie z narzędzia. Biblioteka w prosty sposób pomaga tworzyć nowe widoki poprzez wykorzystanie gotowych elementów. Przykład kodu służącego do stworzenia głównego okna aplikacji:

```python
class MainWindow(QWidget):

    def __init__(self):
        super().__init__()
        initPalette(self)
        self.setWindowTitle("Stylometry Analysis")
        self.resize(500, 220)

        self._layout = QFormLayout()
        self._header = QLabel("Configuration")
        self._header.setAlignment(Qt.Qt.AlignHCenter)
        self._type = TypeWidget()
        self._input = FileWidget("Input directory")
        self._input2 = FileWidget("Classification directory")
        self._statusLabel = QLabel("Ready to start")
        self._runButton = QPushButton("Run analysis")

        self._layout.addRow(self._header)
        self._layout.addRow(self._type)
        self._layout.addRow(self._input)
        self._layout.addRow(self._input2)
        self._layout.addRow(self._statusLabel)
        self._layout.addRow(self._runButton)
        self.setLayout(self._layout)

        self._runButton.clicked.connect(self._runAlgorithm)

        self.show()
```

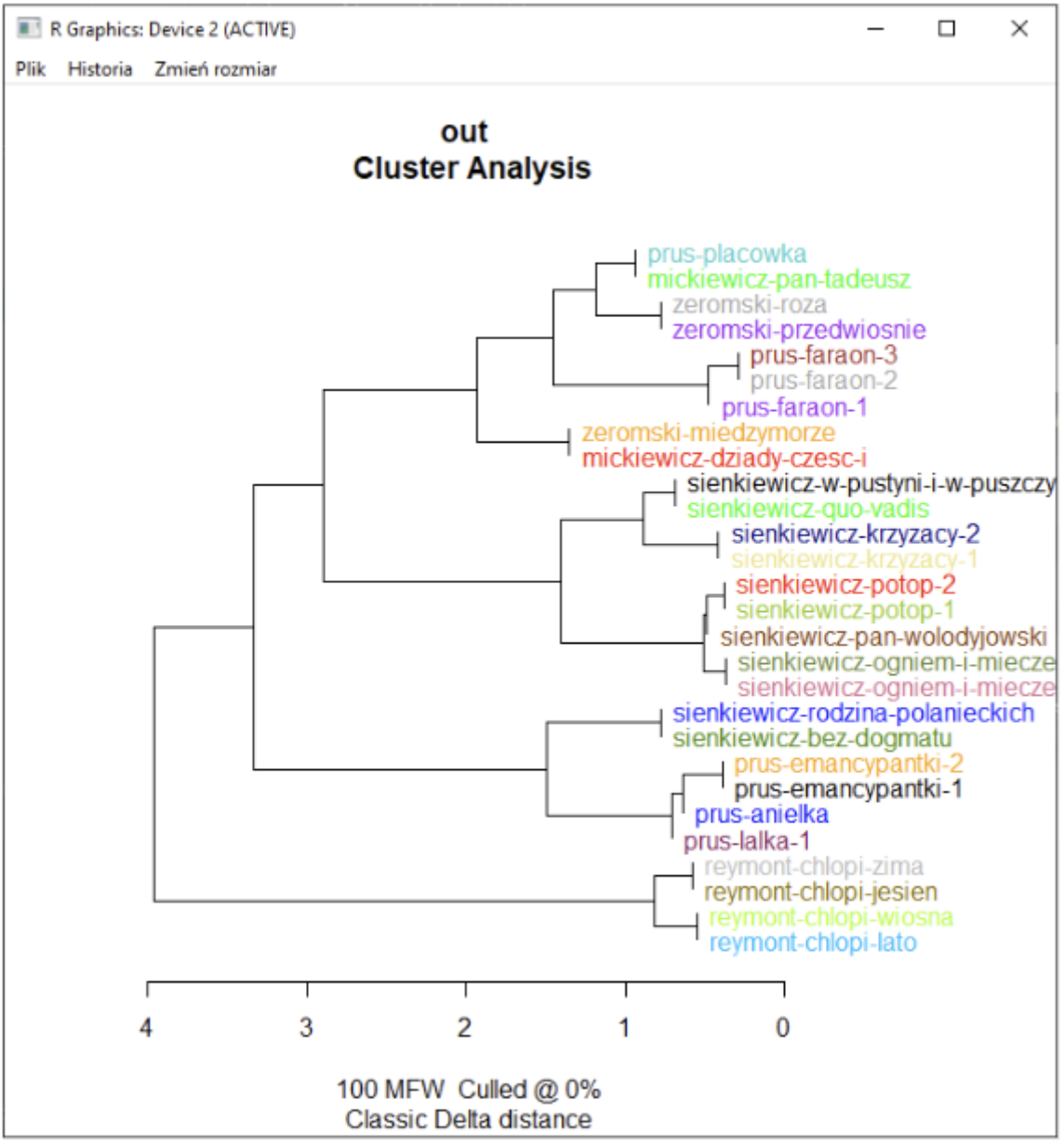
Rezultatem tego prostego kawałka kodu jest:



Użytkownik podczas korzystania z programu ma do wyboru kilka opcji. Pierwszą z nich jest typ analizy, mamy tutaj do wyboru dwie możliwości:

- **analiza stylometryczna** - przedstawia podstawowe funkcje paczki stylo oraz wykresów z nią związanych,
- **analiza klasyfikacyjna** - określa podobieństwo tekstów pomiędzy wskazanymi przez użytkownika katalogami, w których owe teksty się znajdują.

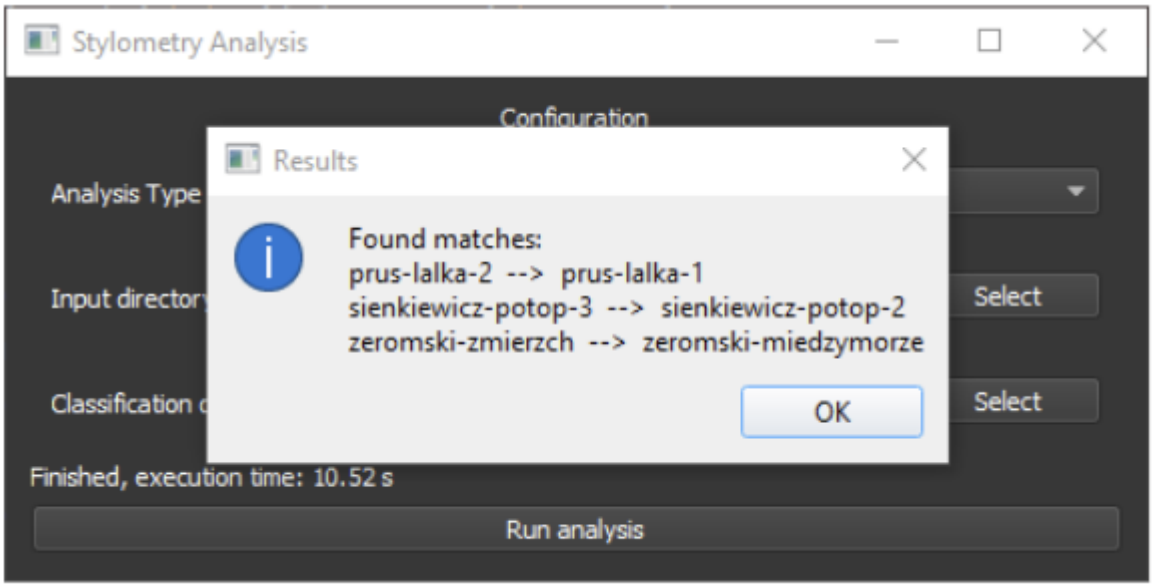
Kolejną opcją jest możliwość wybrania miejsca, z którego ładowane są nasze teksty oraz teksty do porównania (wymagane przy wyborze analizy typu klasyfikacja). Podczas wyboru analizy pierwszego typu, użytkownikowi zostanie pokazany szereg wykresów, dotyczących analizowanych tekstów. Przykładowo jednym z pierwszych wykresów jakie użytkownik zobaczy jest analiza klastrowa, pokazująca zgrupowane teksty tych samych autorów.



Program posiada opcję pomiaru czasu. Dodatkowo analiza tekstów pozwala na wygenerowanie listy z najczęściej używanymi słowami w badanym tekście.


```
"mickiewicz-dziady-czesc-i" "mickiewicz-pan-tadeusz" "prus-anielka" "prus-emancypantki-1" "prus-emancypantki-2" "prus-faraon-1" "prus-faraon-2" "prus-faraon-3"
"prus-lalka-1" "prus-lalka-2" "prus-placowka" "reymont-chlopi-jesien" "reymont-chlopi-lato" "reymont-chlopi-wiosna" "reymont-chlopi-zima" "sienkiewicz-bez-dogmatu"
"sienkiewicz-krzyzacy-1" "sienkiewicz-krzyzacy-2" "sienkiewicz-ogniem-i-mieczem-1" "sienkiewicz-ogniem-i-mieczem-2" "sienkiewicz-pan-wolodyjowski" "sienkiewicz-
potop-1" "sienkiewicz-potop-2" "sienkiewicz-potop-3" "sienkiewicz-quo-vadis" "sienkiewicz-rodzina-polanieckich" "sienkiewicz-w-pustyni-i-w-puszczy" "zeromski-
miedzymorze" "zeromski-przedwiosnie" "zeromski-roza" "zeromski-zmierzch"
"i" 3.50934732699246 3.11992628022291 3.20942341342666 2.69025872027792 2.67401194639596 3.32562174667438 3.43023096523192 3.2990120605594 2.99495855763479
2.79345111460549 3.37914442938915 5.16605166051661 4.0744033187867 3.88880822881649 4.24748246542063 2.99374198313056 3.92053379575527 4.38278521057141
3.70944842471543 3.75124065795166 3.6267008921896 3.43539955190441 3.57164299580395 3.37412108483244 4.07624963694883 3.51173824869034 4.03945608114854
4.10066877859908 3.4326710816777 2.56899122485395 3.79346680716544
"się" 1.50869137422106 2.19111559670601 2.68447782851778 2.79913233546371 2.82488935656588 2.47848127105093 2.68226526754158 2.51635873749038 2.73092369477912
2.79036783522955 2.95052954677433 2.92190677008811 3.69457464576262 3.66990920157697 3.48340582265585 2.5521825319703 2.55023183925811 2.48138140117314
2.57691771097488 2.60007974143416 2.64728604894582 2.46288438969544 2.54164359961922 2.5814994064469 2.54165456496097 2.71810630203609 2.73656676184581
2.05326762876921 2.2700515084621 1.87235706778128 2.68703898840885
"w" 2.75500163988193 2.48365439466409 1.97814860654688 1.77272840604041 2.02717340843676 1.98176436566529 1.9055316584016 2.03361560174493 2.11996923865675
1.88850861776586 1.84719188120455 1.75013178703216 1.74694251387317 1.84835658204805 2.05190496715124 2.16737279900494 2.01587062238474 2.15432017399328
2.06071349434966 2.10296824763957 2.15882470218811 2.03612668138434 2.11336664376297 2.26554652543147 2.30455992270715 2.10495319423798 2.22439647770612
3.19723102194063 2.87711552612215 3.11163256825793 3.21390937829294
"nie" 2.32863233847163 1.37346965641318 2.07204131279075 2.21902711867224 2.05038531769366 1.70618854829381 1.80415901805529 1.90050038491147 1.87985986499188
2.24077328646749 1.99236788983215 1.90978236312975 1.78196217876192 1.80703380649773 2.25174039679741 2.44567963617989 2.20812756813812 2.08347063863442
1.92776423664968 2.06055259117245 1.98624333349948 2.32747088609673 2.14205186020293 2.03999634736554 2.01293351273495 2.13962412907087 1.95183802240371
0.615980288630764 1.8616629874908 1.48126420767031 1.26448893572181
"na" 1.60708428993113 1.91612912662542 1.75195254150484 1.77688388753605 1.65036674816626 1.82526451876297 1.60278363790789 1.6310623556582 1.82090062377168
1.63799216847039 2.01448994828969 1.98358310113713 1.92338774850493 2.11192888015278 1.90692632211381 1.46460916546819 2.04508613865119 1.92611876359322
1.93268828323116 1.86204731890635 1.81802322683547 1.80797544501801 1.72711686890856 1.81901196237787 1.72723446881909 1.3991859454479 1.9877797967293
2.11779889710196 1.98798135884229 1.94813130942778 2.68703898840885
```

Wybierając klasyfikację typu drugiego, użytkownik niejako dopasowuje teksty z jednego katalogu do drugiego. W przykładzie poniżej, w folderze testującym znalazły się 3 książki nie znajdujące się w folderze głównym. Jak możemy zaobserwować, program poprawnie dokonał klasyfikacji, ponieważ wybrał dzieła tych samych autorów.



Jak można zaobserwować program posiada sporo opcji, a w kodzie sprowadza się to głownie to wywołania dwóch funkcji z pakietu “stylo”.

```
        if type == "Stylometry":
            R.stylo(**{
                "gui": False,
                "path": os.getcwd() + "/out",
                "corpus.dir": input1,
                "corpus.format": "plain",
                "corpus.lang": "Polish",
            })
            self._finished()
        else:
            R.classify(**{
                "gui": False,
                "path": os.getcwd() + "/out",
                "training.corpus.dir": input1,
                "test.corpus.dir": input2,
                "corpus.format": "plain",
                "corpus.lang": "Polish",
            })
```

Ciekawy zabieg, który udało nam się uzyskać jest uruchomienie pakietu “stylo”, (napisanego w języku R) w Pythonie. Było to możliwe dzięki skorzystaniu z biblioteki rpy2 - R in Python (https://rpy2.github.io/). Przykład użycia: