

POLITECHNIKA WROCŁAWSKA

ZASTOSOWANIE INFORMATYKI W GOSPODARCE

Relacje pomiędzy bytami w tekstach literackich, drugi kamień milowy.

Lider grupy:

Przemysław WUJEK 234983

Skład grupy:

Paweł CZARNECKI 234974

Łukasz ŁUPICKI 257536

Dawid PIECHOTA 235851

Bartosz RODZIEWICZ 226105

Wojciech WÓJCIK 235621

Prowadzący:

dr inż. Tomasz WALKOWIAK

27 maj 2020

1 Wczytywanie tekstów z plików

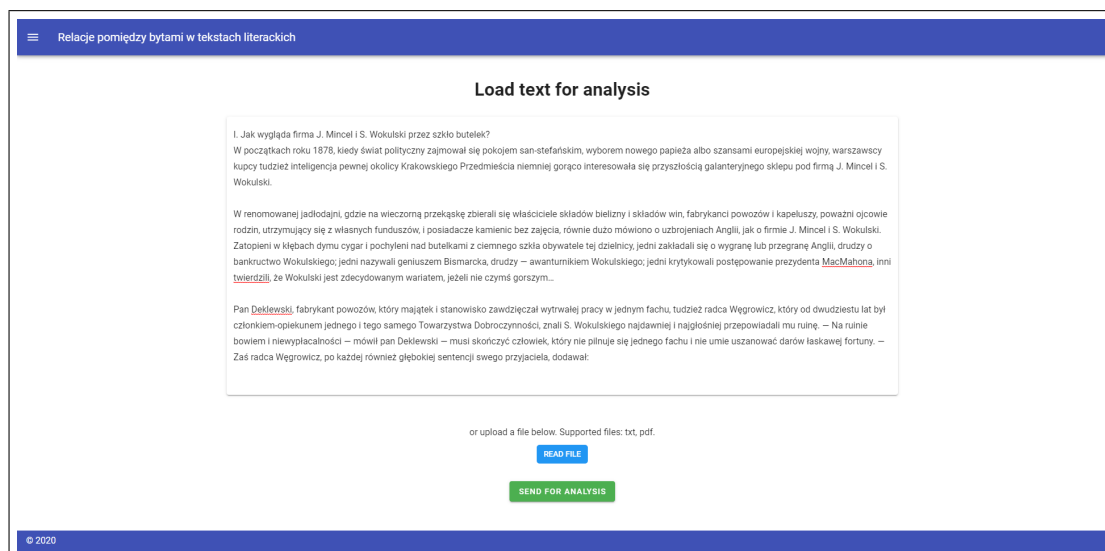
Interfejs użytkownika umożliwia wczytywanie tekstu podanego przez użytkownika ręcznie, oraz poprzez wczytanie pliku.

1.1 Obsługiwane formaty

Program aktualnie wspiera wczytywanie tekstu z plików tekstowych (np. txt) oraz plików PDF. Ekstrakcja tekstu z pdf jest wykonywana przez bibliotekę `pdf.js`. Operacja ta jest wykonywalna lokalnie na komputerze użytkownika. Plik pdf musi zawierać tekst, czytanie tekstu z obrazów (OCR) nie jest wspierane.

1.2 Ręczna modyfikacja

Po wczytaniu tekstu jest on uzupełniany do pola, w którym użytkownik może zmienić jego treść, usunąć część tekstu lub całkowicie zrezygnować z wysłania. Możliwe jest również ręczne uzupełnienie tego pola z pominięciem wczytywania pliku.



Rysunek 1: Interfejs wczytywania tekstu

1.3 Dane

Cały tekst książki pakowany jest w JSONa i wysyłany na RESTowy endpoint wystawiony przez backend.

2 Dane dotyczące postaci z tekstu

2.1 Dane

Dane zwracane z przeanalizowanego tekstu zwracają wagi połączeń pomiędzy bytami oraz liczbę wystąpień postaci na podstawie opisanej w wcześniejszym dokumencie metod wędrujących okien. Te dane wykorzystywane są w funkcji parsującej dane dla frontendu.

2.2 Format

Dane zwracane przez funkcję parsującą zawierają się w formacie danych JSON. Format prezentuje się następująco:

```
{
  "nodes": [
    {
      "name": "Pawel",
      "class": "rare"
    },
    {
      "name": "Przemek",
      "class": "rare"
    },
    {
      "name": "Wojtek",
      "class": "rare"
    },
    ...
  ],
  "links": [
    {
      "source": 0,
      "target": 1,
      "value": 244,
      "type": "straight"
    },
    {
      "source": 0,
      "target": 2,
      "value": 108,
      "type": "dotted"
    },
    {
      "source": 0,
      "target": 3,
      "value": 164,
      "type": "straight"
    },
    ...
  ]
}
```

Sekcja *nodes* odpowiada za przekazanie danych wszystkich węzłów bytów wraz z ich nazwami(name) oraz sklasyfikowaną po stronie backendu częstością występowania(class). Częstość występowania zależna jest od maksymalnej wartości występowania postaci.

Sekcja *links* odpowiada za informacje dotyczące własności połączeń. Opisuje ona między którymi wierzchołkami następuje połączenie(source i target), wartość value, która określa moc połączenia(im większa tym silniejsze połączenie) oraz type, które określa jaki graficzny styl będzie miało połączenie wierzchołków. Styl *straight* określa mocne połącze-

nia, a styl *dotted* słabe. Moc połączeń jest klasyfikowana odnosząc się do maksymalnej wartości połączenia.

3 Zapis wyników do plików

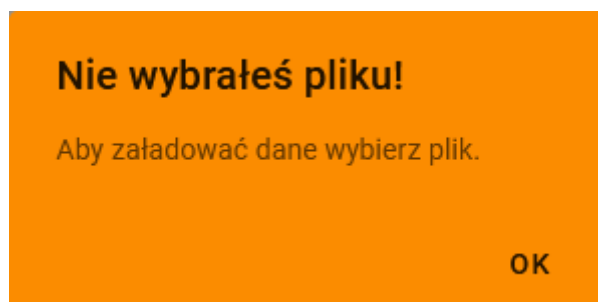
3.1 Zapis modelu wykresu do pliku

Do zapisania aktualnego stanu modelu do pliku wykorzystane zostało API przeglądarki do wygenerowania pliku, który następnie jest pobierany. Struktura pliku jest identyczna do tej opisanej w rozdziale 2.2. Przy wczytywaniu z pliku użytkownik wybiera plik z dysku z danymi wykresu.



Rysunek 2: Element UI odpowiadający za wczytanie danych grafu

W przypadku gdy ktoś niepoprawnie wypełni pole z wyborem pliku i kliknie przycisk Wczytaj dane z pliku zostanie mu wyświetlona informacja ukazana na rysunku 3



Rysunek 3: Komunikat informujący o braku pliku do wczytania

3.2 Eksport wykresu do pliku graficznego

Eksport wykresu odbywa się poprzez generowanie pliku PNG przez przeglądarkę. Jest on generowany na podstawie obecnej zawartości tagu *jsvg* w którym znajduje się renderowany wykres. Plik graficzny zawiera legendę oraz wszystko co jest widoczne na wykresie w czasie kliknięcia przycisku *Zapisz graf do pliku graficznego*. Takie działanie umożliwia ustawienie grafu w odpowiadającym użytkownikowi stanie i eksport tego grafu, jego części lub zbliżenia na daną część do pliku graficznego.



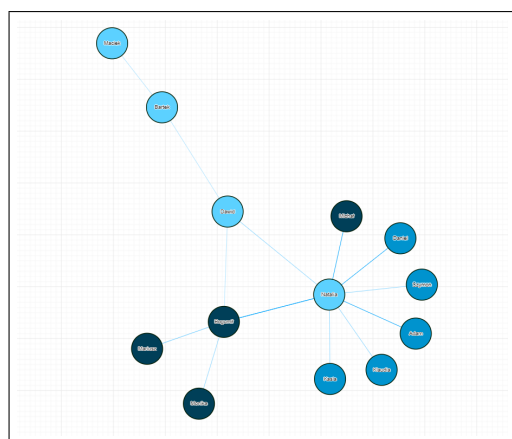
Rysunek 4: front

4 Prezentacja danych

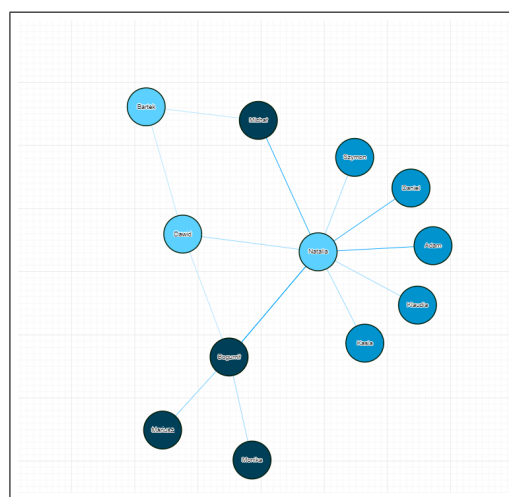
4.1 Coś o całości aplikacji maybe

4.2 Łączenie mylnie rozdzielonych bytów

Użytkownik posiada możliwość ręcznego złączenia dwóch bytów w jeden [rys 6]. Funkcjonalność ta jest przydatna w przypadku gdy *jak to się nazywało?* mylnie rozróżni jeden byt jako dwa osobne. Łączenie bytów polega na usunięciu pierwszego z nich oraz przypisaniu odpowiednich połączeń do bytu drugiego. Wynik takiej operacji został przedstawiony na rysunku 5. Wszelkie zmiany dokonane przez użytkownika mogą zostać przywrócone przyciskiem *Przywróć zmiany*.

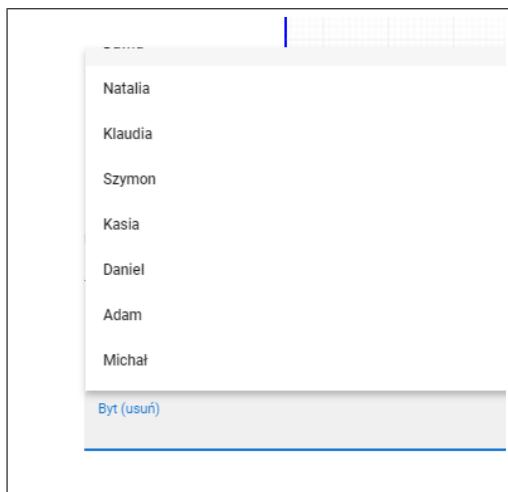


(a) Przed złączeniem

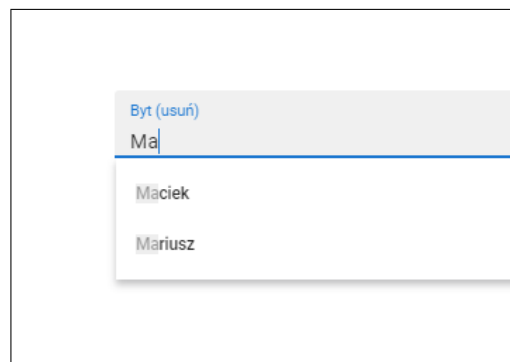


(b) Po złączeniu

Rysunek 5: Efekt złączenia bytów



(a) Lista



(b) Autouzupełnianie

Rysunek 6: Wybór bytów do złączenia

5 Link do repozytorium projektu

<https://github.com/Isild/ZIWG>