Dokumentacja Specyfikacji Wymagań (SRS)

Projekt: Analiza Text Mining oraz grupowanie dokumentów tekstowych na podstawie zawartości tematycznej
Wersja dokumentu: 1.0
Data: 01.06.2025
Autor: [Stanisław Drąg, Szymon Rzeczkowski]

1. Wprowadzenie:

Niniejszy dokument opisuje specyfikację wymagań dla skryptu R, który realizuje pełną analizę text mining oraz grupowanie dokumentów przy użyciu algorytmu klastrowania na podstawie zawartości plików txt. System wykorzystuje techniki czyszczenia tekstu, eksploracyjnej analizy danych, a także algorytm klastrowania, w ramach uczenia maszynowego nienadzorowanego. System generuje wizualizacje globalnej częstości słów i częstości słów w poszczególnych klastrach w postaci chmur słów, wizualizację klastrów dokumentów w formie wykresu oraz wykres przypisania dokumentów do klastrów.

2. Cele systemu:

- Wczytanie dokumentów tekstowych (pliki .txt) z odpowiednim kodowaniem (UTF-8)
- Przetwarzanie i oczyszczanie tekstu (normalizacja, tokenizacja)
- Usunięcie zbędnych słów
- Zliczanie częstości występowania słów i wizualizacja w formie chmury słów
- Odkrywanie struktur tematycznych poprzez przeprowadzanie algorytmu klastrowania.
- Wizualizacja częstości dla poszczególnych klastrów w formie chmury
- Wizualizacja dopasowania dokumentów do klastrów

Docelową grupą użytkowników systemu są analitycy danych, badacze albo studenci - osoby zajmujące się eksploracją dużych zbiorów danych tekstowych. System umożliwia względnie szybkie poznanie i zrozumienie tematyki dokumentów oraz zakresu poruszanych w nich zagadnień, bez konieczności ich ręcznego przeglądania.

3. Wymagania funkcjonalne:

- **Wczytywanie danych:**
  - Skrypt powinien umożliwiać wczytanie danych z lokalnego folderu, zawierającego pliki .txt.
  - Skrypt powinien obsługiwać kodowanie UTF-8.
- P**rzetwarzanie i oczyszczanie tekstu:**
  - Skrypt powinien konwertować cały tekst na małe litery.
  - Skrypt ma usuwać symbole (@, | , ~ , adresy URL oraz znaczniki RT i *via,* a także inne niepożądane pozostałości.
  - Skrypt ma usuwać predefiniowaną listę słów stop w języku polskim oraz zapewniać możliwość jej rozszerzenia.
  - Skrypt powinien usuwać słowa, które nie wnoszą zróżnicowania w danych (np. słowa występujące z taką samą częstością we wszystkich dokumentach lub tylko w jednym)
- **Analiza częstości:**
  - Skrypt powinien generować macierze częstości.
  - Skrypt powinien zawierać funkcjonalność, pozwalającą na wyświetlenie listy najczęściej występujących słów.
- **Wizualizacja danych:**
  - Skrypt powinien umożliwiać wizualizację wyników (wykresy ggplot2, chmury słów)

    - Skrypt powinien generować chmury częstości słów (globalną i dla każdego ze stworzonych klastrów).
    - Skrypt powinien generować wykres obrazujący przyporządkowanie dokumentów do klastrów.
- **Agregacja danych:**
    - Skrypt powinien grupować dokumenty na podstawie ich zawartości tematycznej za pomocą algorytmu klastrowania.
    - W skrypcie powinna być dostępna funkcjonalność do **doboru optymalnej liczby klastrów** (np. poprzez metodę sylwetki).
    - Wyniki klastrowania powinny być prezentowane w interaktywnej tabeli.

4. Wymagania niefunkcjonalne:

- **Wydajność:**
    - System powinien przetwarzać korpus dokumentów zawierający 100 pozycji w czasie nie dłuższym niż 20 sekund.
- **Bezpieczeństwo:**
    - System powinien zapewnić poprawność danych wejściowych.
- **Niezawodność:**
    - Skrypt powinien poprawnie obsługiwać różne formaty danych tekstowych.
    - Skrypt powinien poprawnie obsługiwać brakujące wartości.
- **Użyteczność:**
    - Wizualizacje tworzone przez system powinny być czytelne.
    - Wybór folderu z plikami powinien być intuicyjny.
- **Kompatybilność:**
    - Skrypt powinien być kompatybilny z R w wersji 4.0 lub nowszej.
    - Skrypt powinien korzystać z bibliotek tm, cluster, ggplot2, dplyr, wordcloud, factoextra, DT.

5. Interfejsy użytkownika:

- Wejście:
  - Wybór folderu plików tekstowych .txt. poprzez okno dialogowe.
- Wyjście:
  - Chmura słów dla macierzy TDM.
  - Wizualizacja klastrów (cluster).
  - Tabela przypisania dokumentów do klastrów (cluster).
  - Chmury słów dla każdego klastra (cluster i wordcloud).
  - Wizualizacja przypisania dokumentów do klastrów (ggplot2).

6. Wymagania dotyczące danych:

- Dane to zbiór plików tekstowych, zebranych w jednym folderze. Każdy plik jest traktowany jako jeden dokument.
- Skrypt zakłada, że dane tekstowe są w języku polskim.
- Skrypt nie obsługuje analizy i klastrowania dla danych tekstowych z innych źródeł niż pliki txt.
- Skrypt nie obsługuje plików o rozmiarze powyżej 100 MB.

7. Słownictwo dokumentacji:

- **Korpus:** Zbiór elementów danych poddawany analizie
- **Stopwords:** Często występujące słowa, usuwane, gdyż nie niosą za sobą wartości tematycznej
- **Stemming:** Proces redukowania słów do ich rdzenia
- **Bag of Words:** Model tekstu, wykorzystujący nieuporządkowany zbiór słów
- Klastrowanie: Metoda analizy danych, polegająca na grupowaniu obiektów ze względu na podobieństwo
- **Metoda Sylwetki (Silhouette Method):** technika służąca do oceny jakości klastrowania, poprzez mierzenie, jak dobrze dany obiekt dopasowany jest do

własnego klastra. Pomaga dobrać odpowiednią liczbę klastrów.

- **Chmura słów:** Graficzne przedstawienie częstości słów, gdzie rozmiar odpowiada częstości występowania danego wyrazu.

8. Przypadki użycia (Use Cases):

- **Użytkownik:**
  - wczytuje folder plików .txt
  - uruchamia analizę
  - wybiera zasugerowaną przez system optymalną liczbę klastrów
  - wyświetla wyniki
  - generuje wykresy oraz własny raport html
  - przeprowadza analizę w oparciu o wygenerowane materiały
- **System:**
  - przetwarza tekst
  - oczyszcza tekst
  - generuje globalną chmurę częstości słów
  - sugeruje optymalną liczbę klastrów, korzystając z Metody Sylwetki
  - wykonuje algorytm klastrowania i przydziela dokumenty do klastrów
  - generuje chmury częstości słów dla poszczególnych klastrów
  - system wyświetla tabelę z przypisaniem dokumentów do klastrów

- **Testowe przypadki użycia:**
  - Test z folderem zawierającym pliki .txt o podobnej tematyce
  - Test z folderem zawierającym pliki .txt ze zróżnicowaną tematyką
  - Test z folderem zawierającym pliki .txt z polskimi znakami
  - Test z pustym folderem

9. Scenariusze użytkownika (User Stories):

Scenariusz 1: Porównanie narracji politycznej z przekazem medialnym

Jako: Analityk ekonomii politycznej

Chcę: Porównać dominujące tematy w wypowiedziach polityka (np. z transkrypcji filmów na jego kanale YouTube) z przekazem występującym w tradycyjnych mediach.

Aby: Zidentyfikować zbieżności i różnice pomiędzy wypowiedzią polityka a medialną narracją. Ocenić wpływ mediów, ich działanie względem przekazu danego polityka.

Kryteria akceptacji:

- Użytkownik może wczytać dane tekstowe, będące transkrypcją filmu z serwisu YouTube, do pliku tekstowego.
- Skrypt przeprowadza analizę text mining oraz algorytm klastrowania
- Skrypt przeprowadza analizę częstości dla poszczególnych klastrów, wydzielając w ten sposób tematy dominujące w dokumentach.
- Użytkownik może modyfikować liczbę klastrów.
- Użytkownik może generować podsumowania w formie chmury słów oraz wykresu dopasowania dokumentów do klastrów.

Scenariusz 2: Odkrywanie tematów w zbiorze raportów badawczych

Jako: Badacz

Chcę: Automatycznie pogrupować zbiór raportów badawczych na podstawie ich treści

Aby: W łatwy i szybki sposób określić dominujące tematy oraz móc w prostszy sposób przeprowadzić dalszą analizę raportów.

Kryteria akceptacji:

- Użytkownik może wczytać raporty badawcze jako pliki .txt.
- Skrypt przeprowadza analizę text mining oraz algorytm klastrowania, przydzielając raporty do konkretnych obszarów tematycznych.

- Skrypt przeprowadza analizę częstości dla poszczególnych klastrów, wydzielając w ten sposób tematy dominujące w dokumentach.
- Użytkownik może modyfikować liczbę klastrów.
- Użytkownik może generować podsumowania w formie chmury słów oraz wykresu dopasowania dokumentów do klastrów.