ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2015
Seria: ORGANIZACJA I ZARZĄDZANIE z. 86 Nr kol. 1946

Marcin LEJA, Ireneusz J. JÓŹWIAK
Politechnika Wrocławska
Wydział Informatyki i Zarządzania
marcin.leja93@gmail.com; ireneusz.jozwiak@pwr.edu.pl

STRATEGIA PRZETWARZANIA DOKUMENTÓW TEKSTOWYCH OPARTA NA HEURYSTYCZNEJ ANALIZIE DANYCH

Streszczenie. W artykule przedstawiono problem związany z przetwarzaniem dokumentów tekstowych przez człowieka. Zaproponowano heurystyczne podejście, inspirowane sposobem, w jaki ludzki mózg przetwarza dokumenty tekstowe, które może zostać wykorzystane do usprawnienia tego procesu. Przedstawiony algorytm rozpoznaje frazy na podstawie zdefiniowanego zbioru znanych fraz oraz cech indywidualnych danej frazy. Efektem działania algorytmu jest zbiór rozpoznanych fraz oraz odpowiadająca im pozycja w tekście.

Słowa kluczowe: przetwarzanie dokumentów tekstowych, analiza tekstu.

TEXT DOCUMENT PROCESSING STRATEGY BASED ON HEURISTIC DATA ANALYSIS

Summary. The paper presents the problem of processing text documents. It proposes a heuristic approach, inspired by the way the human brain processes text documents, which can be used to facilitate this process. The algorithm recognizes phrases based on a defined set of known phrases and individual characteristics of the phrase. The result of the algorithm is a set of identified phrases, and the corresponding position in the text.

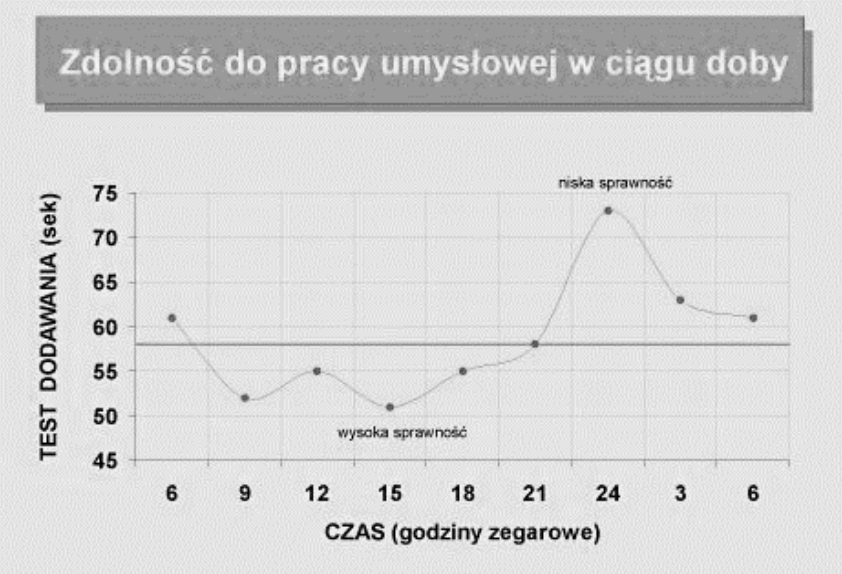
Keywords: text document processing, text analysis.

1. Wprowadzenie

Przetwarzanie dokumentów tekstowych, w celu pozyskania zawartych w nich informacji, jest dla wielu ludzi zadaniem, które wykonują nawet kilkadziesiąt razy dziennie. Przykładem takiego dokumentu może być lista produktów do zakupienia w sklepie, ogłoszenie sprzedaży samochodu czy też CV kandydata na pewne stanowisko w firmie. Bez względu na to, jaki jest

to rodzaj dokumentu, istotna jest efektywność przetwarzania i jakość pozyskanych, w jego wyniku informacji. Dla człowieka, efektywność i jakość przetwarzania są ściśle związane z liczbą słów na sekundę, którą człowiek potrafi przeczytać oraz mocą zbioru rozumianych słów. W wielu przypadkach jednak przetwarzania wymaga tak duża liczba dokumentów, że przetworzenie ich, nawet bardzo szybko czytającej osobie, z dużą mocą zbioru rozumianych słów, zajęłoby zbyt wiele czasu. Oczywiście jednym z rozwiązań tego problemu jest zatrudnienie wielu osób. Praktyka pokazuje jednak, że zasoby ludzkie są ograniczone m.in. ze względu na ich wysoki koszt. Dodatkowo, wydajność oraz jakość przetwarzania tekstu, wykonywane przez człowieka, może ulegać pogorszeniu. Potwierdzenie tej tezy można znaleźć analizując wyniki eksperymentu przeprowadzonego przez Centralny Instytut Ochrony Pracy – Państwowy Instytut Badawczy, które zostały przedstawione na poniższym rysunku.

W ramach eksperymentu przeprowadzono test dodawania, gdzie na kryterium oceny składały się czas oraz poprawność rozwiązania zadania. Najkrótszy czas oznacza najlepszy wynik testu.



Rys. 1. Zdolność do pracy umysłowej w ciągu doby
Fig. 1. The capacity for mental work during the day
Źródło: Centralny Instytut Ochrony Pracy – Państwowy Instytut Badawczy. Zdolność do pracy umysłowej w ciągu doby. http://archiwum.ciop.pl/15705.html (10.09.2015).

Niestety stwierdza się brak bezpośredniego rozwiązania przedstawionego problemu w postaci oprogramowania lub algorytmu.

Z wyżej wymienionych powodów wynika cel niniejszego artykułu: **usprawnienie procesu przetwarzania dokumentów tekstowych**.

W ramach poszukiwania rozwiązania podjęto próbę wykorzystania algorytmów wyszukiwania wzorca w tekście. Pomimo zadowalającej wydajności, podejście oparte na tych algorytmach jest obciążone ograniczeniem, co zostanie dokładnie przedstawione w kolejnym

rozdziale. W odpowiedzi na to ograniczenie zaproponowano heurystyczny algorytm, który pozwala na rozpoznawanie fraz, wykorzystując dodatkowe informacje, zawarte w tekście.

2. Algorytmy wyszukiwania wzorca w tekście

Jednym z możliwych rozwiązań problemu przetwarzania dokumentów tekstowych, przedstawionego w poprzednim rozdziale, jest wykorzystanie algorytmów wyszukiwania wzorca w tekście. Każda fraza z bazy wiedzy zawierającej tylko frazy dotyczące danego wycinka rzeczywistości zostaje potraktowana, jako wzorzec do wyszukania w danym dokumencie tekstowym. W przypadku odnalezienia frazy, która odpowiada zadanemu wzorcowi, zapisana zostaje informacja o rozpoznaniu danej frazy oraz jej pozycji w tekście. Wynikiem jest zbiór znanych fraz, które występują w dokumencie tekstowym oraz pozycja ich występowania, co umożliwi zaprezentowanie wyników użytkownikowi w graficzny sposób. Jest to istotne, ponieważ przetwarzanie dokumentów oparte na tym rozwiązaniu wymaga nadzoru człowieka, który pozostaje odpowiedzialny za ostateczny wynik, tzn. zapisanie informacji zawartych w dokumencie tekstowym w wymaganej formie reprezentacji danych. Nadzór użytkownika jest wymagany, ponieważ nie ma gwarancji, że znaleziona fraza występująca w pewnym kontekście, w dokumencie tekstowym ma dokładnie takie znaczenie, jakie jest nadawane w wyniku rozpoznania frazy. W przypadku błędnego rozpoznania frazy, użytkownik może poprawić błąd. Pomimo że proponowane podejście nadal wymaga pewnego nakładu pracy ze strony użytkownika, wydajność pracy przy użyciu tego rozwiązania ulega znacznej poprawie, co zostanie dokładniej przedstawione w dalszej części artykułu.

Przeprowadzono przegląd kilku algorytmów, które zostały następnie zaimplementowane i wykorzystane do przetwarzania dokumentów tekstowych w sposób, który został przedstawiony powyżej. Ponieważ przegląd został wykonany jedynie w celu przetestowania podejścia opartego na algorytmie wyszukiwania wzorca w tekście, więc kryterium wyboru była wysoka dostępność ich specyfikacji w literaturze.

Poniżej opisano kilka algorytmów wyszukiwania wzorca w tekście.

1. Algorytm Knutha – Morrisa – Pratta [6], [3]
2. Algorytm Boyera – Moore'a [1], [7]
3. Algorytm Karpa – Rabina [5], [4],[3]

Ad 1. Algorytm Knutha – Morrisa – Pratta

Opiera się na funkcji prefiksowej, która pozwala określić, w przypadku znalezienia niedopasowania danego znaku wzorca, ile znaków od pozycji wyjściowej porównywania wzorca można bezpiecznie ominąć. Wartości funkcji obliczane są dla wzorca, przed

rozpoczęciem wyszukiwania. Algorytm porównuje tekst ze wzorcem od lewej do prawej strony. Złożoność obliczeniowa algorytmu wynosi O(m+n).

Ad 2. Algorytm Boyera-Moore'a

Bazuje na dwóch heurystykach, które pozwalają określić maksymalne przesunięcie względem znaku, od którego zaczęło się porównywanie, po znalezieniu niezgodności ze wzorcem. Algorytm wybiera maksymalne możliwe przesunięcie, wynikające z ewaluacji obu heurystyk. Wartości obu funkcji obliczane są dla wzorca, przed rozpoczęciem wyszukiwania. Algorytm porównuje wzorzec z tekstem od prawej do lewej strony. Złożoność obliczeniowa w pesymistycznym przypadku wynosi O(m*n), jednak średnio jest zbliżona do O(n), a w najlepszym przypadku nawet O(n/m).

Ad 3. Algorytm Karpa-Rabina

Alternatywnie do algorytmów przedstawionych w Ad 1 i Ad 2 w tym algorytmie nie położono nacisku na uzyskanie możliwie dużego przesunięcia w przypadku braku podobieństwa wzorca do teksu. Bazuje on na przyśpieszeniu porównywania wzorca z tekstem. W tym celu stosowana jest funkcja haszująca, przekształcająca tekst na wartość liczbową. Algorytm wykorzystuje zależność, że jeśli dwie frazy są równe, to wartości funkcji haszującej dla tych fraz też są równe. Wartość funkcji haszującej wyznaczana jest dla wzorca przed rozpoczęciem wyszukiwania.

Algorytm Karpa-Rabina jest wolniejszy od algorytmów przedstawionych w Ad 1 i Ad 2 w przypadku wyszukiwania pojedynczego wystąpienia wzorca w tekście. Jednakże wykazuje się on krótszym czasem działania dla wyszukiwania wielowzorcowego.

Złożoność algorytmu, w pesymistycznym przypadku (o bardzo niskim prawdopodobieństwie wystąpienia) wynosi O((n-m + 1) *m), jednak średnio O(m+n).

Przedstawione algorytmy charakteryzują się zadowalającą szybkością działania, tzn. czas przetwarzania dokumentu o rozmiarze 2 stron formatu A4 wynosi około 2 sekund.

Jednakże stwierdza się, że wykorzystanie ich do przetwarzania dokumentów tekstowych jest obciążone ograniczeniem. Ponieważ wyszukiwanie ograniczamy tylko do fraz, które zawierają się w bazie wiedzy, więc zbiór rozpoznanych fraz może mieć maksymalnie taką moc, jak baza wiedzy. Jednakże, dokumenty tekstowe mogą zawierać frazy, których znaczenie może być rozpoznane na podstawie innych czynników, np. formatu (data itp.) czy kontekstu występowania frazy. W kolejnym rozdziale zaproponowano algorytm, którego powyższe ograniczenie nie dotyczy.

3. Heurystyczna analiza danych

W wielu sytuacjach podejściem do rozwiązywania problemu jest zamodelowanie sposobu rozwiązywania tego problemu przez ludzki mózg. W ramach artykułu przeanalizowano sposób, w jaki człowiek przetwarza tekst i przedstawiono go w postaci algorytmu heurystycznego.

Człowiek, przetwarzając dokument w celu pozyskania informacji, najpierw czyta słowo, a następnie wyszukuje jego znaczenia w swojej bazie wiedzy, analizuje format, w jakim słowo jest zapisane i kontekst jego występowania. Stwierdza się, że takie podejście do przetwarzania dokumentów tekstowych daje szersze możliwości wyszukiwania fraz, ponieważ przeglądane są również frazy, których nie ma w bazie wiedzy i podejmowana jest próba rozpoznania ich znaczenia na podstawie indywidualnych cech danej frazy.

Jak opisano powyżej, człowiek zwykle przetwarza tekst wyraz po wyrazie. Z tego powodu poniższy algorytm korzysta z dodatkowego bytu, który będzie dostarczał kolejne frazy z zadanego tekstu.

Poniżej sformułowano problem, przedstawiono dane wejściowe oraz wyjściowe algorytmu, a także zaproponowano algorytm przetwarzania dokumentów tekstowych.

Sformułowanie problemu

Wyszukanie w tekście *T*, fraz *Bi*, takich, że *Bi* zawiera się w bazie wiedzy *B* oraz próba rozpoznania fraz *Ci* takich, że $\neg (Ci \in B)$, na podstawie formatu i kontekstu występowania frazy *Ci*.

Dane wejściowe algorytmu:

- zbiór znanych fraz - baza wiedzy *B*,
- tekst do przetworzenia *T*.

Dane wyjściowe algorytmu

zbiór rozpoznanych fraz *Bi*, gdzie $Bi \in B$ oraz fraz *Ci* rozpoznanych na podstawie formatu i kontekstu występowania

Algorytm

1. Ustaw wskaźnik na początek tekstu.
2. Dopóki nieprawda, że koniec tekstu, wykonuj:
   2.1. Odczytaj słowo wskazywane przez wskaźnik.
   2.2. Sprawdź, czy istnieją frazy w bazie wiedzy złożone z kilku słów, zaczynające się od słowa.
   2.2.1. Jeśli istnieją, analizuj słowo przyrostowo, rozpoznając najdłuższy możliwy ciąg znaków i przejdź do kroku 2.5.
   2.3. Sprawdź czy słowo zawiera się w bazie wiedzy.
   2.3.1. Jeśli tak, przejdź do kroku 2.5.
   2.4. Spróbuj rozpoznać słowo na podstawie formatu i kontekstu występowania.

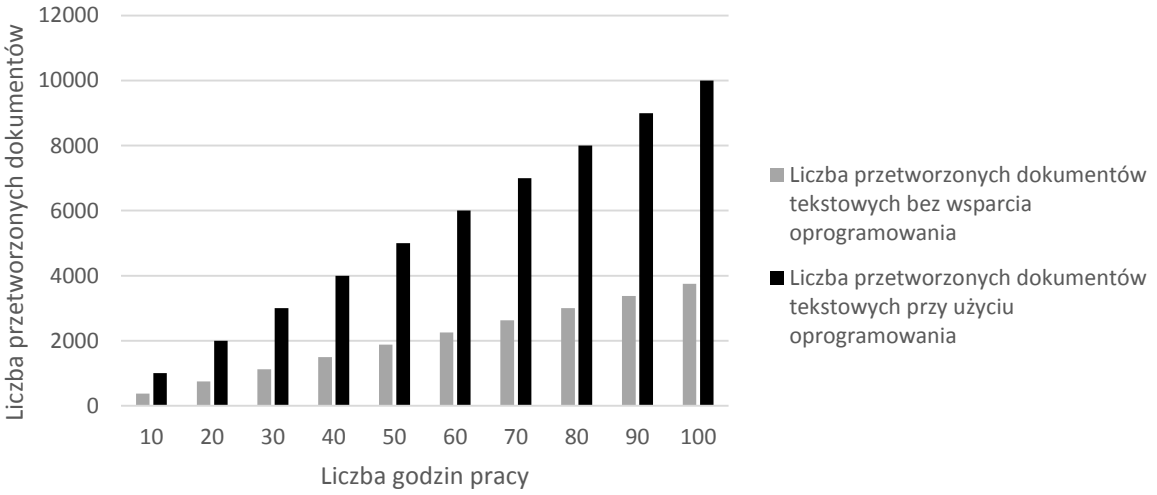
2.5. Zapisz wynik.
2.6. Przesuń wskaźnik o odpowiednia liczbę pozycji.
2.7. Przejdź do kroku 2.
3. Zakończ

4. Wpływ zaproponowanego rozwiązania na wydajność pracy

Aby określić wpływ rozwiązania przedstawionego w poprzednim rozdziale na wydajność pracy, przeprowadzono eksperyment w gronie ekspertów, którzy ze względu na charakter wykonywanej pracy bardzo często zajmują się przetwarzaniem dokumentów tekstowych. Określono, że w ramach prowadzonej działalności, przetwarzane dokumenty mają średnio rozmiar 1,5 strony.

Średni czas przetwarzania dokumentu w standardowy sposób wynosi **8 minut**.

Średni czas przetwarzania dokumentu wykorzystując zaproponowane rozwiązanie informatyczne wynosi **3 minuty**.



Rys. 2. Zestawienie liczby przetworzonych dokumentów przez 5 pracowników w czasie
Fig. 2. Comparison of the number of documents processed by 5 employees in time
Źródło: opracowanie własne.

Z przeprowadzonych badań wynika, że wprowadzenie proponowanego rozwiązania, jako wsparcia procesu przetwarzania dokumentów tekstowych, może przełożyć się na ponad 2-krotne zwiększenie wydajności pracowników.

Na rysunku 2 zwizualizowano powyższe wyniki badań i przedstawiono zestawienie potencjalnej liczby przetworzonych dokumentów tekstowych przez 5 pracowników bez i ze wsparciem oprogramowania. Pominięto spadki wydajności pracowników spowodowane zmęczeniem. Każda godzina pracy została potraktowana niezależnie od innych.

5. Podsumowanie

W pewnych branżach przetwarzanie dokumentów tekstowych jest zadaniem bardzo często wykonywanym. Standardowy sposób przetwarzania wykonywany przez człowieka bez wsparcia oprogramowania jest drogi i narażony na spadek wydajności. Z tego powodu warto ten proces usprawnić. Z przeprowadzonych w ramach artykułu eksperymentów wynika, że wzrost wydajności z tytułu wsparcia tego procesu przez oprogramowanie może wynieść ponad 250%. Uzyskany wynik jednocześnie oznacza, że cel niniejszego artykułu, definiowany jako **usprawnienie procesu przetwarzania dokumentów tekstowych**, został osiągnięty.

Nie oznacza to jednak, że temat usprawniania procesu przetwarzania dokumentów tekstowych został wyczerpany. Warto zastanawiać się nad sposobami uzyskania jeszcze wyższej jakości wyników tego procesu, a ostatecznie nawet jego pełnej automatyzacji.

Bibliografia

1. Boyer R.S, Moore J.S.: A fast string searching algorithm. Communications of the ACM. Vol. 20, No.10, 1977, p. 762-772.
2. Centralny Instytut Ochrony Pracy – Państwowy Instytut Badawczy. Zdolność do pracy umysłowej w ciągu doby. http://archiwum.ciop.pl/15705.html (10.09.2015).
3. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C.: Introduction to Algorithms, MIT Press 2001, p. 790-813.
4. Crochemore M., Rytter W.: Text Algorithms, Oxford University Press, New York 1994, p. 367-369.
5. Karp R.M., Rabin M.O.: Efficient randomized pattern-matching algorithms. IBM J. RES. Dev. Vol. 31(2), 1987, p. 249-260.
6. Knuth D.E., Morris (Jr) J.H., Pratt V.R.: Fast pattern matching in strings, SIAM Journal on Computing, Vol. 6(2), 1997, p. 323-350.
7. Sedgewick R.: Algorithms, Addison-Wesley Publishing Company. 1983, p. 241-254.

Abstract

Nowadays, processing of text documents is a task very often performed. In some sectors, particularly. For example, recruitment companies process a vast amount of CV documents. It turns out that the transformation of a text document can be very time-consuming, hence the need to improve the process. As no direct solution for this problem was found, the aim of this article was to propose solutions that could greatly accelerate this process. As part of the

search for a solution, an approach based on string matching algorithms was tested. However, the ability to recognize phrases with this approach was limited. To bypass this limitation, a heuristic algorithm was proposed. A study was conducted to determine the impact of proposed solution on labor productivity. It has been noted more than twofold increase in productivity.