STUDIA INFORMATICA 2013
Volume 34 Number 2B (112)

Artur NIEWIAROWSKI, Marek STANUSZEK
Politechnika Krakowska, Instytut Informatyki

MECHANIZM IDENTYFIKACJI I KLASYFIKACJI TREŚCI[1]

Streszczenie. Artykuł opisuje mechanizm identyfikacji i klasyfikacji treści, oparty na metodzie ważenia terminów, bazującej na odwrotnej częstości dokumentowej, częstości wystąpienia terminu i odległości Levenshteina. Zaproponowany mechanizm zaimplementowano w program analizujący tematy i opisy prac dyplomowych, w celu automatycznego doboru promotorów i recenzentów.

Słowa kluczowe: ekstrakcja słów kluczowych, ważenie terminów, odwrotna częstość dokumentowa, częstość wystąpienia terminu, odległość Levenshteina, termin, analiza danych tekstowych.

THE MECHANISM OF IDENTIFICATION AND CLASSIFICATION OF CONTENT

Summary. This paper presents the mechanism of identification and classification of content, based on terms weighted method with inversed document frequency analysis and Levenstein distance technique. The proposed mechanism is applied in the analysis of topics and descriptions of selected diploma thesis, to automatic selection of supervisors and reviewers.

Keywords: keywords extracting, inversed document frequency, term frequency, Levenshtein distance, text mining, database mining

1. Wprowadzenie do problemu

Mając bazę danych ogólnie rozumianych dokumentów tekstowych, warto zaimplementować mechanizm umożliwiający szybką analizę treści reprezentowanych przez te dane, dla ekstrakcji informacji wykorzystywanych w dalszej selekcji. Na przełomie lat powstało wiele

[1] Prezentowane w pracy wyniki badań zrealizowano w ramach tematu nr F-3/105/DS/2012, finansowanego z dotacji na naukę przez Ministerstwo Nauki i Szkolnictwa Wyższego w 2012 roku.

prac naukowych, badających zagadnienia ekstrakcji słów kluczowych z danych tekstowych, reprezentujących różne metody i różne podejścia w zależności od problemu. W obszarze tematu publikacji warte uwagi są m.in. te, badające: metody ekstrakcji i hierarchizacji wyrazów o kluczowym znaczeniu w dokumentach, wykorzystujące heurystykę i modele probabilistyczne [9]; ekstrakcje terminów – przy wykorzystaniu bazy korpusu dokumentów i sieci Bayesa [8] lub sztucznych sieci neuronowych, stosowane w szczególności w celu szybkiej analizy dokumentów w postaci stron internetowych [11] itd. Wszystkie wymienione metody służą (lub mogą zostać użyte) do skutecznej identyfikacji dokumentów poprzez opisanie ich kluczowymi słowami lub frazami, które z kolei umożliwiają m.in. sprawne wyszukiwanie tych dokumentów na bazie wprowadzonych wyrazów.

Artykuł przedstawia propozycję modelu mechanizmu[2], umożliwiającego identyfikację osób zajmujących się określoną tematyką, na podstawie stworzonych przez nich dokumentów tekstowych i wpisanych przez zewnętrznego użytkownika słów kluczowych, charakteryzujących określoną tematykę. Przedstawione w publikacji rozwiązanie daje możliwość oszacowania autorów na bazie ich dzieł, na podstawie ważności słów kluczowych użytych w zapytaniu, w poszczególnych ich dziełach, co daje możliwość sporządzenia odpowiedniego rankingu autorów. Dodatkowo, mechanizm został zoptymalizowany pod kątem analizy terminów podobnych, w celu bardziej precyzyjnego obliczenia wag poszczególnych terminów, z pominięciem konieczności użycia słownika wyrazów bliskoznacznych, używanego w niektórych publikacjach o podobnym zakresie badań [12].

W ramach testów, opisany mechanizm zaimplementowano w program, umożliwiający oszacowanie problematyki, jaką zajmuje się dany promotor, w celu optymalnego doboru recenzenta w obrębie danego wydziału lub uczelni. Studentom, jako użytkownikom mechanizmu, algorytm identyfikuje propozycję promotora na podstawie zaproponowanego przez studenta tematu (np. takiego tematu, który jeszcze nie jest zdefiniowany w bazie danych). Identyfikacja promotora bazuje na szacowaniu słów kluczowych i ich wspólnego występowania, przy jednoczesnej eliminacji słów nieistotnych.

Koncepcja automatu, pozwalającego na identyfikację treści danych tekstowych i ich odpowiednie kategoryzowanie, została zaimplementowana jako moduł systemu obsługi tematów prac dyplomowych na Wydziale Fizyki, Matematyki i Informatyki Politechniki Krakowskiej i obecnie jest w fazie testów.

[2] W publikacji pojęcie *mechanizm* określa współpracujące ze sobą, w ramach analizy danych, algorytmy.

2. Model analizy danych

Mechanizm identyfikacji treści reprezentowanych przez analizowane dane tekstowe bazuje na ekstrakcji słów kluczowych. W jej procesie wykorzystuje się popularną metodę ważenia terminów[3], bazującą na odwrotnej częstości dokumentowej (ang. *inversed document frequency*) i częstości wystąpienia terminu (ang. *term frequency*) [1-3, 5-6, 8-11], w połączeniu z odległością edycyjną. Automat analizy częstości wystąpienia opiera się na badaniu tzw. odległości Levenshteina (ang. *Levenshtein distance*) [4]. Algorytm ten polega na obliczeniu najmniejszej liczby zmian, niezbędnych do przeprowadzenia na znakach dwóch porównywanych ciągów, dla osiągnięcia ich identyczności[4]. Odległość Levenshteina jest uogólnieniem odległości Hamminga (ang. *Hamming distance*) [7]. W przeciwieństwie do odległości Hamminga, odległość Levenshteina umożliwia porównywanie ciągów o różnej długości, co bezpośrednio przekłada się na liczne zastosowania algorytmu, m.in. w korekcie pisowni, maszynowym tłumaczeniu tekstów czy eksploracji danych tekstowych, a szerzej w mechanizmach wyszukiwarek internetowych, procesorach tekstów itp.

Siłą mechanizmu jest jego adaptacja do wielkości zasobów tekstowych jakimi dysponuje, ich powiększenie pozwala na znaczne podniesienie jego dokładności. Dodatkowe treści wnoszą jednak wiele terminów nieistotnych z punktu widzenia przyjętych kryteriów identyfikacji treści. Stąd też konieczne jest, dla zoptymalizowania procesu identyfikacji, wprowadzenie słownika terminów nieistotnych, czyli takich, które nie mają czy też nie powinny mieć wpływu na wynik procesu analizy treści. Zaproponowana implementacja automatu analizy odległości Levenshteina została użyta dla zminimalizowania błędów porównywania słów (terminów) podobnych, tj. odmienionych lub zawierających błędy ortograficzne. Metoda z użyciem algorytmu odległości transformacyjnej sprawdza się w szczególności, gdy dane tekstowe zawierają odmienione słowa specjalistyczne.

2.1. Ekstrakcja słów kluczowych

W celu oszacowania wystąpienia ustalonego terminu w badanych zestawach danych, zastosowano formułę obliczającą tzw. złożoną wagę tego terminu, składającą się z iloczynu kartezjańskiego odwrotnej częstości dokumentowej i częstości wystąpienia terminu.
Złożoną wagą terminu $tf\text{-}idf_{t,d}$ jest wartość liczbowa obliczona według wzoru (1):

$$tf\text{-}idf_{t,d} = \log \frac{D}{df_t} \times tf_{t,d} \tag{1}$$

[3] *Termin* – wyraz lub wyrażenie o specjalnym znaczeniu w jakiejś dziedzinie (źródło: Słownik Języka Polskiego PWN).
[4] Dopuszczalnymi operacjami na ciągach są: dodanie znaku, usunięcie znaku, podmiana znaku na inny.

gdzie: D – liczba wszystkich badanych dokumentów tekstowych, df_t – częstość dokumentowa, która określa liczbę dokumentów tekstowych dla danego wystąpienia terminu, $tf_{t,d}$ – liczba wystąpień terminu w dokumencie.

2.2. Analiza miary podobieństwa pomiędzy terminami

Dla oszacowania podobieństwa pomiędzy terminami, został wykorzystany algorytm odległości Levenshteina.

Odległością Levenshteina K jest wartość liczbowa $d(N,M)$ powstałej macierzy **d**, wg formuły (2):

$$\begin{aligned} &\mathop{\mathrm{I}}_{i=1}^{N}\mathop{\mathrm{I}}_{j=1}^{M} d(i,j) = \min(d(i-1,j)+1, d(i,j-1)+1, d(i-1,j-1)+\beta) \\ &\begin{cases} \beta = 0 : a(i) \equiv b(j) \\ \beta = 1 : a(i) \neq b(j) \\ d(i,0) = i \\ d(0,j) = j \\ d(0,0) = 0 \end{cases} \end{aligned} \tag{2}$$

gdzie: $\mathop{\mathrm{I}}_{i=1}^{N}$ – symbol oznaczający iterację dla i = (1, ..., N), **d** – macierz o rozmiarach N+1, M+1, utworzona z dwóch porównywanych terminów, N, M – długości badanych terminów, $d(i,j)$ - (i,j) – ty element macierzy **d,** min – funkcja zwracająca wartość najmniejszą z podanych, β – zmienna przybierająca odpowiednio wartości 0 lub 1, $a(i)$ – i-ty element ciągu znaków terminu a, $b(j)$ – j-ty element ciągu znaków terminu b.

Stąd odległość Levenshteina K jest minimalną liczbą operacji przeprowadzających termin porównywany w termin odniesienia:

$$K = d(N,M)$$

Miara podobieństwa dwóch terminów P obliczana jest wg zależności (3):

$$P = 1 - \left(\frac{K}{K_{\max}}\right);\ K_{\max} = \max(N, M), \quad \begin{matrix} K \geq 0, M > 0, N > 0 \\ P \in \langle 0,1 \rangle \end{matrix} \tag{3}$$

gdzie: $K_{\max}$ – wartość długości najdłuższego z badanych terminów i jednocześnie liczba kroków zmieniających, za pomocą operacji podmiany lub wstawienia, jeden termin w drugi dla przypadku, w którym terminy zawierają taki zestaw znaków, że należy wykonać operację podmiany dla wszystkich znaków terminu krótszego i odpowiednią liczbę operacji dodania znaków w tym ciągu, w celu przeprowadzenia go w termin dłuższy[5] (tzn. przypadek, w którym odległość Levenshteina jest równa długości najdłuższego z terminów).

[5] Jeżeli ciągi są tej samej długości, ciągiem krótszym dla opisanej reguły jest dowolny z dwóch ciągów.

Zasada działania zależności (3) została przedstawiona w formie przykładu w tabeli 1. Tabela zawiera cztery zestawienia porównania dwóch ciągów[6] w postaci: obliczonej odległości Levenshteina (K) oraz miary odległości (P).

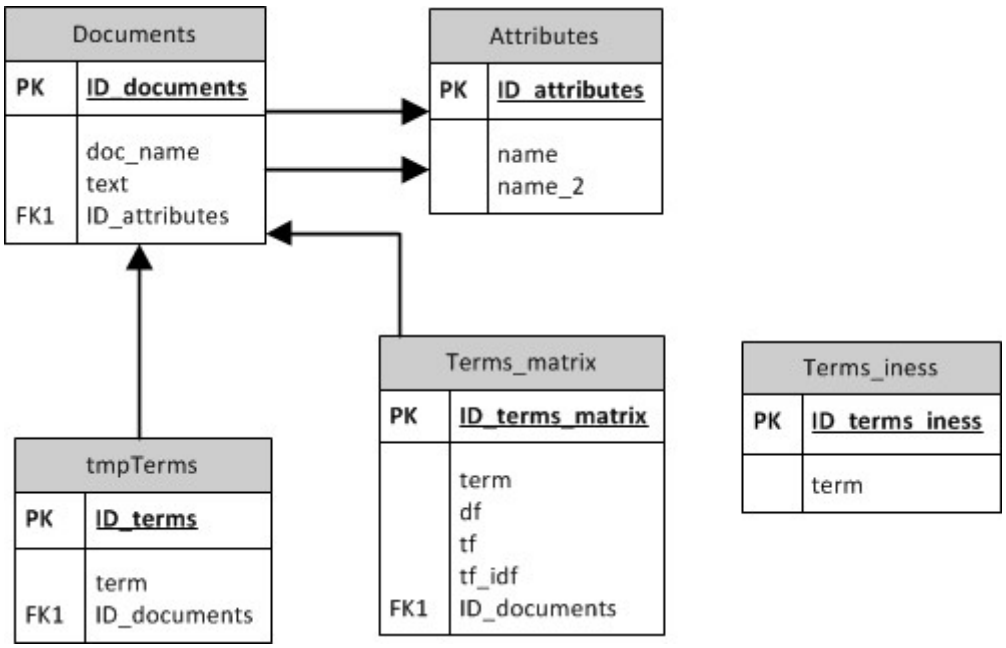
Tabela 1

Przykład użycia zależności (3)

Lp.	Ciąg 1	Ciąg 2	K	P
1	Studium	Studia	2	$\approx$0,71 (tj. $\approx$71% podobieństwa)
2	Studia	Studia	0	1 (tj. 100% podobieństwa)
3	Studia	Berek	6	0 (tj. 0% podobieństwa)
4	XXXX	XXYY	2	0,5 (tj. 50% podobieństwa)

Klasyfikowanie podobieństwa terminów jest przeprowadzane na podstawie porównania wartości miary podobieństwa P do wartości q ustalanej przez użytkownika mechanizmu, będącej wyznacznikiem prawdopodobieństwa występowania terminu w danej dziedzinie.

3. Projekt modułu



Rys. 1. Schemat przedstawiający najważniejsze klasy (tabele) i relacje pomiędzy nimi
Fig. 1. Scheme describes the most important classes (tables) and relationships

Przystępując do realizacji wprowadzonego wyżej mechanizmu, na wstępie będziemy chcieli zidentyfikować strukturę statyczną systemu. Z opisu wynika skład najważniejszych klas (tabel) omawianego mechanizmu analizy danych tekstowych, a więc:

[6] Pojęcie *ciąg* oznacza ciąg tekstowy i w publikacji jest równoważne z pojęciem *termin*.

- *Attributes* (przechowująca informacje o atrybutach[7] dokumentów / tekstów),
- *Documents* (przechowująca informacje o dokumentach),
- *tmpTerms* (przechowująca terminy i informacje, w których dokumentach występują;pełni funkcję tabeli roboczej, tymczasowej),
- *Terms_matrix* (będąca tabelą wyjściową pierwszego etapu analizy danych),
- *Terms_iness* (tabela zawierająca terminy nieistotne; przed rozpoczęciem analizy uzupełniana jest przez użytkownika terminami, które uzna on za nieistotne w dziedzinie analizowanych danych tekstowych).

W trakcie analizy danych tworzone są dodatkowo klasy (tabele tymczasowe): *tmpTM* oraz *tmp_vec_query*, które ze względu na swoje chwilowe zastosowanie, nie zostały ujęte w schemacie (rys. 1). Alternatywnym schematem tabel może być taki, w którym zastosowana została tabela przechowująca unikalne terminy, pochodzące z przeanalizowanych dokumentów, a następnie połączona z tabelami: *Terms_matrix* relacją klucz główny - klucz obcy (zastąpienie kolumny *term* kolumną *ID_terms*) oraz *tmpTerms* analogiczną relacją. Spowoduje to uniknięcie redundancji danych w tabelach. Jednak minusem takiego rozwiązania może być skomplikowanie i nieznaczne wydłużenie procesu analizy danych. W nowej tabeli należałoby umieścić wyłącznie unikalne terminy, pochodzące z analizowanych dokumentów, a następnie odpowiednio skojarzyć wartości klucza głównego, charakteryzujące poszczególne terminy, z terminami w tabeli *tmpTerms*, która ostatecznie jest tabelą tymczasową i usuwaną po wygenerowaniu danych w tabeli *Terms_matrix*. Zapytanie użytkownika, które w etapie 2 (opisanym w dalszej części pracy) komparowane jest z danymi *Terms_matrix*, składa się bezpośrednio z terminów, to oznacza, że w zapytaniu SQL należałoby dołączać lewostronnie tabelę zawierającą unikatowe terminy do tabeli *Terms_matrix*, w celu porównania z zapytaniem. Tabela *Terms_iness* jest osobną tabelą, niezwiązaną relacjami z pozostałymi, zawierającą terminy, które filtrują zapytanie użytkownika, wykluczając terminy nieistotne. Tabela ta nie byłaby połączona z tabelą zawierającą terminy unikatowe.

Analiza danych obejmuje dwa etapy:

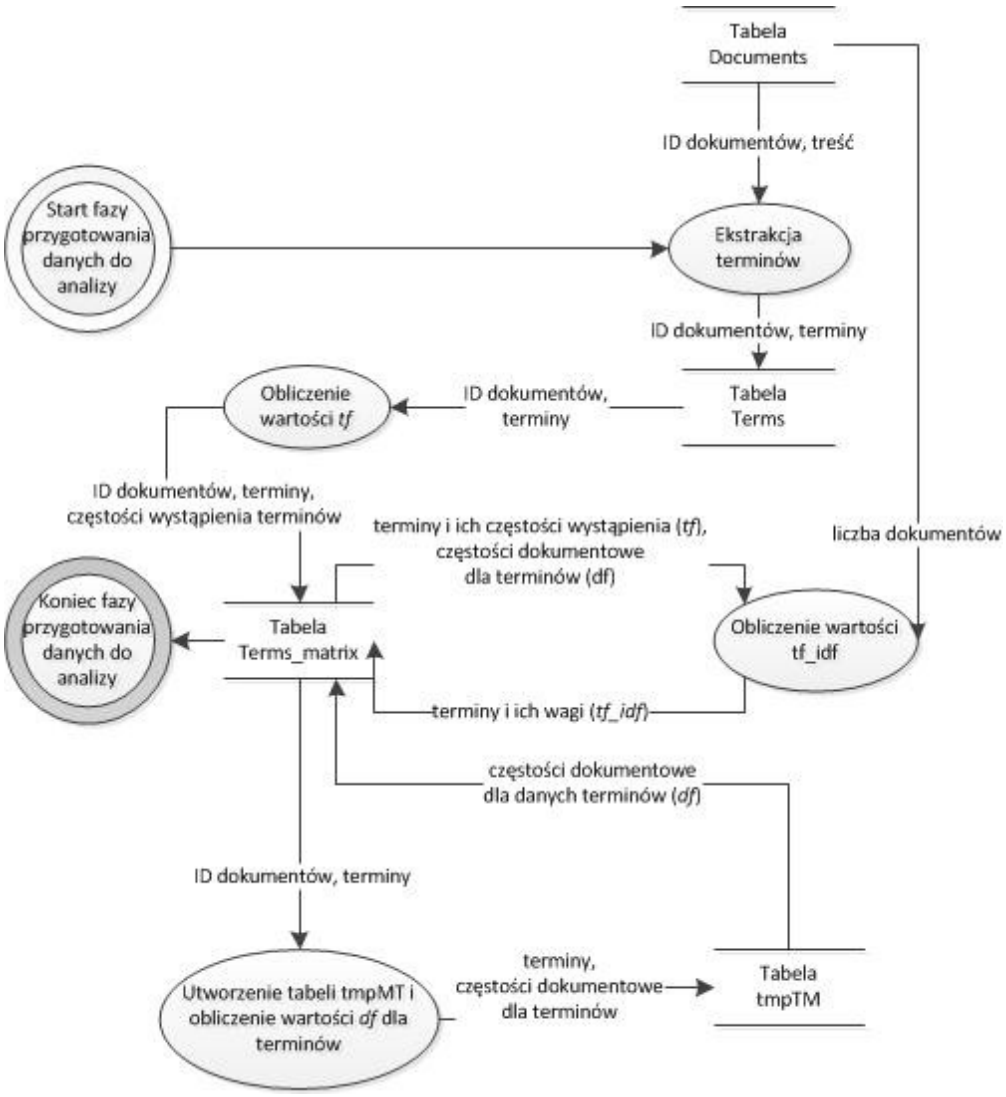
1) periodyczny – tj. okresowej analizy zestawu danych,
2) on-line – tj. w trybie rzeczywistym – po wykonaniu zapytania użytkownika.

3.1. Etap periodyczny

Przed przystąpieniem do etapu on-line dane tekstowe umieszczone w bazie danych należy odpowiednio przygotować, w celu późniejszego przeliczenia wag terminów zawartych w zapytaniu użytkownika (drugi etap).

[7] Atrybutem może być np. autor danego tekstu, wydawnictwo dokumentu itp.

Na podstawie danych zawartych w tabeli *Documents*, uzupełniana jest tabela *tmpTerms*, przechowująca terminy przyporządkowane do odpowiednich dokumentów. Tabela uzupełniania jest danymi okresowo (tzn. co pewien interwał czasu, np. raz dziennie) i przed każdym wypełnieniem danymi należy stare dane usunąć[8] (jednocześnie stosując zabezpieczenie w postaci transakcji o odpowiednim poziomie izolacji, celem zachowania spójności danych). W tabeli umieszczane są terminy wcześniej przefiltrowane, np. niezawierające liczb, czyli wartości niemających wpływu na późniejszą analizę.



Rys. 2. Diagram przepływu danych, przedstawiający etap periodyczny
Fig. 2. Data flow diagram describes periodic stage

Etap przebiega w następujących krokach (zgodnie z diagramem opisanym na rys. 2):

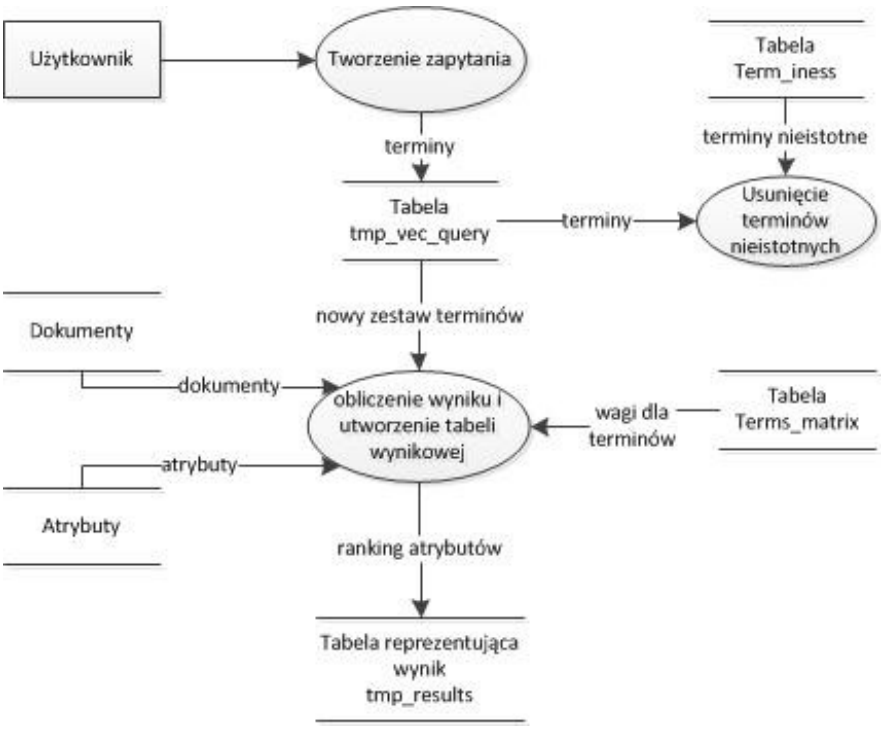
1. Na bazie tabeli *tmpTerms* następuje przeliczenie liczby terminów względem zawartości danych dokumentów i umieszczenie w kolumnie *tf* (reprezentującej częstość wystąpienia terminu) tabeli *Terms_matrix*.

[8] Wynika to ze wzoru (1).

2. Na bazie tabeli *Documents* następuje przeliczenie liczby badanych dokumentów i zapamiętanie w zmiennej.
3. Na bazie tabeli *Terms_matrix* tworzona jest tabela pomocnicza *tmpTM*, tymczasowa, przechowująca informacje o liczbie dokumentów, w których występuje dany termin (tj. częstość dokumentowa).
4. Wykorzystując dane zawarte w tabeli tymczasowej *tmpTM*, następuje obliczenie wagi terminu *tf_idf* (odpowiednia kolumna w tabeli *Terms_matrix*) dla terminów powiązanych odpowiednio z danym dokumentem.

3.2. Etap on-line

Etap on-line analizy danych polega na wykonaniu zapytania użytkownika, czyli zestawieniu zapytania z wcześniej przygotowanymi danymi i zwróceniu odpowiedzi w postaci rankingu atrybutów dokumentów. Etap ten przebiega na bazie tabeli tymczasowej *tmp_vec_query*, przechowującej zapytanie użytkownika oraz tabelę uzupełnioną zestawem, danych w etapie poprzednim – *Terms_matrix*. Zadaniem tej części analizy jest zestawienie dokumentów poprzez terminy je charakteryzujące względem terminów zawartych z zapytaniu użytkownika, a następnie obliczenie sumy wag tych terminów pogrupowanych względem atrybutów dokumentów.



Rys. 3. Diagram przepływu danych, przedstawiający etap on-line
Fig. 3. Data flow diagram describes online stage

Etap przebiega w następujących krokach (zgodnie z diagramem opisanym na rys. 3):

1. Utworzenie wektora zapytania w postaci tablicy tymczasowej *tmp_vec_query*, w celu przechowania terminów wchodzących w skład zapytania użytkownika.
2. Usunięcie z tabeli zawierającej zapytanie użytkownika terminów nieistotnych, określonych w tabeli *Terms_iness*. Terminami nieistotnymi mogą być wyrazy pospolite, np.: „lub", „oraz", „i", „w" lub wyrazy często występujące w danej dziedzinie. Zastosowanie tego typu zabiegu umożliwia znaczne przyspieszenie obliczeń i poprawę dokładności wyniku.
3. Na bazie zapytania *select* tworzona jest tabela tymczasowa *tmp_results*, przechowująca listę atrybutów z wartością będącą wyznacznikiem dla rankingu.

Ostatecznym wynikiem jest ranking potencjalnych atrybutów, utworzony na podstawie sumy wag terminów odpowiednio pogrupowanych względem danego atrybutu tekstu (dokumentu tekstowego).

3.3. Dalsza obróbka danych

Dane wygenerowane w tabeli *tmp_results* można poddać dalszej analizie, w celu doprecyzowania wyniku. Jednym z przykładów jest oszacowanie dolnego pułapu dla wartości rankingu. Ze względu na to, że ta wartość nie jest zamknięta w danym przedziale, punktem odniesienia może być np. liczba pozycji zwracanego wyniku lub pewna część wartości maksymalnej rankingu.

Doprecyzowanie wyniku umożliwia wyświetlenie wyłącznie tych atrybutów tekstów, które charakteryzują znaczną część dokumentów o danej tematyce, z wyłączeniem tych, które charakteryzują teksty w znacznie mniejszym stopniu.

4. Implementacja i testowanie

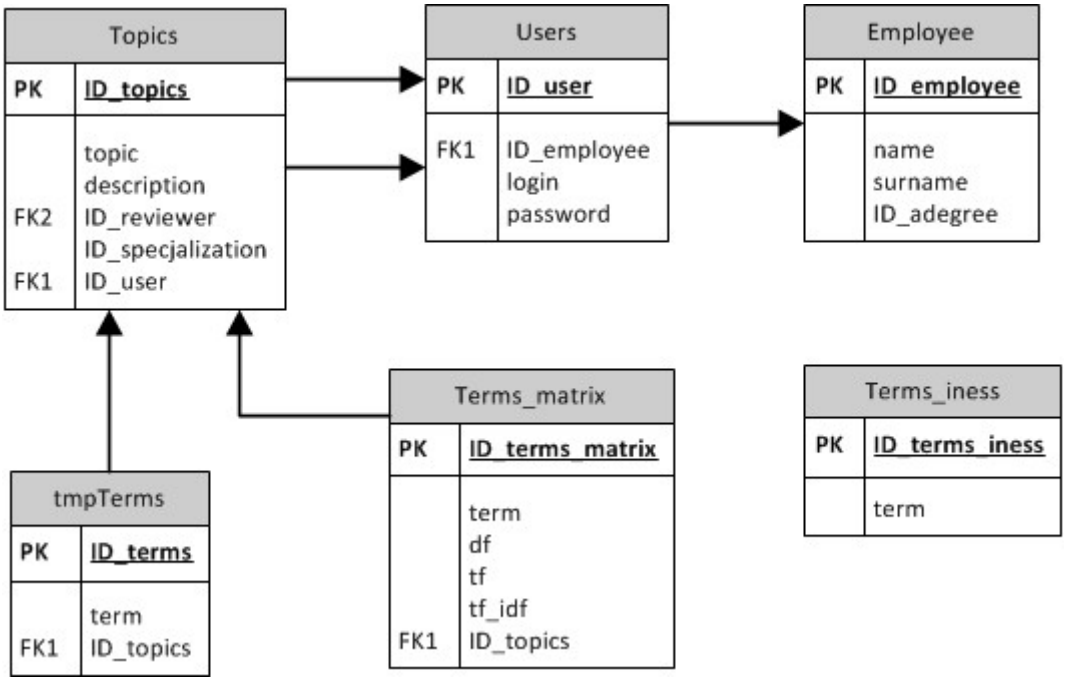
Zaproponowany mechanizm zaimplementowano w systemie analizującym tematy i opisy prac dyplomowych, dla automatycznego doboru promotorów i recenzentów.
W skład najważniejszych tabel systemu analizy danych wchodzą:

- *Users* (przechowująca informacje o użytkownikach systemu),
- *Employee* (przechowująca informacje o pracownikach uczelni; powiązana relacją jeden do wielu z tabelę *Users*),
- *Topics* (przechowująca informacje o tematach i opisach prac dyplomowych),
- *tmpTerms* (przechowująca terminy i informacje o tematach i opisach, w których występują; pełni funkcję tabeli roboczej, tymczasowej),

- *Terms_matrix*,
- *Terms_iness*.

W trakcie analizy danych tworzone są dodatkowo tabele tymczasowe: *tmpTM* oraz *tmp_vec_query*.

Relacje pomiędzy tabelami przedstawia rys. 4.



Rys. 4. Schemat przedstawiający najważniejsze tabele i relacje pomiędzy tabelami
Fig. 4. Scheme describes the most important tables and relationships between tables

4.1. Etap periodyczny

W ramach badanych tematów i opisów uwzględniane są wyłącznie te, będące w relacji z promotorem, a nie z recenzentem.

Na podstawie danych zawartych w tabeli *Topics*, uzupełniana jest tabela *tmpTerms*, przechowująca terminy przyporządkowane do propozycji tematów i opisów prac dyplomowych.

Etap przebiega w następujących krokach[9] (zgodnie z diagramem opisanym na rys. 2):

1. Na bazie tabeli *tmpTerms* następuje przeliczenie liczby terminów względem propozycji tematu oraz opisu i umieszczenie w kolumnie *tf* (reprezentującej częstość wystąpienia terminu) tabeli *Terms_matrix*.
2. Na bazie tabeli *Topics* następuje przeliczenie liczby badanych tematów i ich opisów, a następnie zapamiętanie w zmiennej.
3. Na bazie tabeli *Terms_matrix* tworzona jest tabela pomocnicza *tmpTM*, zawierająca informacje o terminach i ich częstościach dokumentowych.

[9] Implementacja kroków w postaci kodu SQL znajduje się w załączniku.

4. Wykorzystując dane zawarte w tabeli tymczasowej *tmpTM*, następuje obliczenie wartości *tf_idf* dla terminów powiązanych odpowiednio z propozycją tematu pracy dyplomowej.

4.2. Etap on-line

Etap on-line analizy danych zwraca odpowiedź użytkownikowi w postaci rankingu pracowników naukowo-dydaktycznych.

Etap przebiega w następujących krokach[10] (zgodnie z diagramem opisanym na rys. 3):

1. Utworzenie wektora zapytania w postaci tablicy tymczasowej *tmp_vec_query*.
2. Usunięcie z tabeli, zawierającej zapytanie użytkownika, terminów nieistotnych, określonych w tabeli *Terms_iness*. Terminami nieistotnymi mogą być, np. dla kierunku informatyka: „system", „projekt", „aplikacja"; dla kierunku matematyka: „wzór", „równanie", „definicja", „teoria" itd. Zastosowanie tego typu zabiegu umożliwia znaczne przyspieszenie obliczeń i poprawę dokładności wyniku.
3. Na bazie zapytania *select* tworzona jest tabela tymczasowa *tmp_results*, przechowująca odpowiedź w postaci listy pracowników naukowo-dydaktycznych, z wartością będącą wyznacznikiem dla rankingu.

4.3. Algorytm mechanizmu identyfikacji treści w działaniu

Poniżej znajduje się przykładowy zestaw danych tekstowych w postaci propozycji tematów i opisów tematów prac dyplomowych, który zostanie poddany działaniu przez zaproponowany mechanizm.

Tabela 2

Tabela *Topics* z przykładową zawartością danych

ID_topics	topic	description	ID_user
1	Algorytm rozpoznawania treści na stronach WWW	Zaprojektować i zaimplementować algorytm pobierający i analizujący treść, z uwzględnieniem znaczników hipertekstowych	2
2	Algorytm rozwiązywania równań Nernsta-Plancka poprzez przybliżenia różnicowe	Celem pracy będzie implementacja w dowolnym języku programowania wybranych algorytmów rozwiązywania równań Nernsta-Plancka, opisujących transport cząstek naładowanych ośrodkami ciągłymi	1

Po ekstrakcji terminów tabela *tmpTerms* będzie zawierała następujący zestaw danych:

a) dla tematu o *ID_topics*: 1; terminy: algorytm, rozpoznawania, treści, na, stronach, www, zaprojektować i zaimplementować, algorytm, pobierający, i, analizujący, treść, z uwzględnieniem, znaczników, hipertekstowych;

[10] Implementacja kroków w postaci kodu SQL znajduje się w załączniku.

b) dla tematu o *ID_topics*: 2; terminy: algorytm, rozwiązywania, równań, nernsta, plancka, poprzez, przybliżenia, różnicowe, celem, pracy, będzie, implementacja, w dowolnym, języku, programowania, wybranych, algorytmów, rozwiązywania, równań, nernsta, plancka, opisujących, transport, cząstek, naładowanych, ośrodkami ciągłymi.

Etap przygotowania danych do analizy:

1. Tabela *Terms_matrix* będzie zawierała następujący zestaw danych:
   a) dla tematu o *ID_topics*: 1; terminy i wartości *tf*: algorytm (2), rozpoznawania (1), treści (1), na (1), stronach (1), www (1), zaprojektować (1), i (2), analizujący (1) treść (1), z (1), uwzględnieniem (1), znaczników (1), hipertekstowych (1);
   b) dla tematu o *ID_topics*: 2; algorytm (1), rozwiązywania (2), równań (2), nernsta (2), plancka (2), poprzez (1), przybliżenia (1), różnicowe(1), celem (1), pracy (1), będzie (1), implementacja (1), w (1), dowolnym (1), języku (1), programowania (1), wybranych (1), algorytmów (1), opisujących (1), transport (1), cząstek (1), naładowanych (1), ośrodkami ciągłymi (1).
2. Liczba dokumentów: 2.
3. Tabela *tmpTM* będzie zawierała następujący zestaw danych (z uwzględnieniem terminu i liczby propozycji prac dyplomowych, w których występuje): algorytm (2), rozpoznawania (1), treści (1), na (1), stronach (1), www (1), zaprojektować (1), i (1), analizujący (1), treść (1), z, (1), uwzględnieniem (1), znaczników (1), hipertekstowych (1), rozwiązywania (1), równań (1), nernsta (1), plancka (1), poprzez (1), przybliżenia (1), różnicowe (1), celem (1), pracy (1), będzie (1), implementacja (1), w (1), dowolnym (1), języku (1), programowania (1), wybranych (1), algorytmów (1), opisujących (1), transport (1), cząstek (1), naładowanych (1), ośrodkami ciągłymi (1).
4. Tabela *Terms_matrix* będzie zawierała odpowiednio następujące wartości dla *ID_topics* / *term* / *tf_idf*: (1) / algorytm / (0); (1) / rozpoznawania / (0,301); (1) / treści / (0,301); (1) / na / (0,301); (1) / stronach / (0,301); (1) / www / (0,301); (1) / zaprojektować / (0,301); (1) / i / (0,6021); (1) / analizujący / (0,301); (1) / treść / (0,301); (1) / z / (0,301); (1) / uwzględnieniem / (0,301); (1) / znaczników / (0,301); (1) / hipertekstowych / (0,301); (2) / algorytm / (0); (2) / rozwiązywania / (0,6021); (2) / równań / (0,6021); (2) / nernsta / (0,6021); (2) / plancka / (0,6021); (2) / poprzez / (0,301); (2) / przybliżenia / (0,301); (2) / różnicowe / (0,301); (2) / celem / (0,301); (2) / pracy / (0,301); (2) / będzie / (0,301); (2) / implementacja / (0,301); (2) / w / (0,301); (2) / dowolnym / (0,301); (2) / języku / (0,301); (2) / programowania / (0,301); (2) / wybranych / (0,301); (2) / algorytmów / (0,301); (2) / opisujących / (0,301); (2) / transport / (0,301); (2) / cząstek / (0,301); (2) / naładowanych / (0,301); (2) / ośrodkami ciągłymi / (0,301).

Przebieg etapu on-line:

1. Utworzenie zapytania użytkownika, np.: „Algorytm rozwiązujący równanie Nernsta-Plancka”.
2. Po usunięciu terminów nieistotnych, zestaw terminów zapytania będzie następujący: nernsta, plancka.
3. Ranking kształtuje się następująco: dla ID_user: 1 wartość rankingu wynosi: 1,2041, dla użytkownika ID_user: 2 wartość rankingu wynosi: 0. Z tego wynika, że osobą najprawdopodobniej zajmującą się danym tematem jest użytkownik o wartości ID_user: 1.

4.4. Wynik jakości analizy danych mechanizmu zaimplementowanego w program Menedżer Dyplomów

Poniżej (tabela 3) znajdują się wyniki testów działania opisanego mechanizmu, zaimplementowanego w program o nazwie Menedżer Dyplomów, do którego zaimportowano ok. 1 000 tematów prac dyplomowych i ich opisów przez 125 pracowników naukowych Wydziału Fizyki, Matematyki i Informatyki Politechniki Krakowskiej.

W ramach testów brane pod uwagę były następujące parametry: liczba terminów użytych w zapytaniu użytkownika, liczba istotnych terminów w zapytaniu, charakteryzująca tematykę, liczba prawidłowo oszacowanych osób zajmujących się rozpoznaną tematyką, liczba osób, które zostały wytypowane nieprawidłowo, liczba osób pominiętych, które zajmują się tematyką.

Na uwagę zasługuje fakt wysokiej jakości odpowiedzi. Pomimo tego, że znaczna część promotorów proponowała tematy prac dyplomowych z różnych dziedzin informatyki, matematyki i fizyki, to zastosowane sumowanie wag terminów zważonych metodą tf-idf, będących w korelacji z tematami pracowników, skutecznie osłabiło lub wzmocniło ich powiązanie z tematyką reprezentowaną przez słowa kluczowe w zapytaniu użytkownika.

Tabela 3

Tabela przedstawiająca wyniki jakości analizy tematów prac dyplomowych

Numer testu	1	2	3	4	5	6	7	8
Ilość terminów w zapytaniu	2	3	5	8	8	10	11	14
Ilość istotnych terminów w zapytaniu	2	3	3	4	3	6	5	5
Liczba trafionych autorów	8	4	4	4	3	5	3	4
Liczba nietrafionych autorów	0	0	0	1	2	1	0	1
Liczba pominiętych autorów	0	0	0	0	0	2	1	3

W testach można zaobserwować trzy przypadki pominięcia pracowników zajmujących się daną dziedziną oraz cztery przypadki błędnie wytypowanych promotorów. Po przeanalizowaniu błędnych odpowiedzi okazało się, że są one wynikiem użycia popularnej terminologii w temacie i opisie pracy dyplomowej oraz użycia rzadkiego słownictwa, nieistotnego do

określenia dziedziny pracy. Zwiększenie liczby umieszczanych pozycji, reprezentujących prace dyplomowe, przyczyni się do zwiększenia jakości uzyskiwanych odpowiedzi.

Wywiad przeprowadzony wśród pracowników naukowych i studentów korzystających z zaimplementowanych algorytmów, potwierdził dobrą jakość generowanych odpowiedzi.

5. Podsumowanie

Zaproponowany w pracy mechanizm umożliwia identyfikację treści reprezentowanych przez analizowane zestawy danych. W procesie ekstrakcji słów kluczowych wykorzystano metodę ważenia terminów, bazującą na odwrotnej częstości dokumentowej i częstości wystąpienia terminu. Dużą efektywność i dokładność wprowadzonego mechanizmu uzyskano poprzez dodatkowe zastosowanie algorytmu odległości Levenshteina i tabeli identyfikującej terminy nieistotne.

W praktyce opisana metoda może znaleźć zastosowanie np. w programach zarządzających treścią prac dyplomowych na uczelniach wyższych. Przy wyróżnieniu dwóch rodzajów użytkowników: pracownika naukowo-dydaktycznego i studenta, docelowa funkcjonalność może mieć dwojaki charakter. W pierwszym przypadku promotor ma możliwość automatycznego doboru recenzentów do zaproponowanej przez siebie pracy dyplomowej. W drugim przypadku użytkownik – student – może zaproponować temat pracy dyplomowej (lub ewentualnie zbiór słów kluczowych), a mechanizm wyłoni potencjalnych promotorów (np. dodatkowo proponując przegląd ich wcześniejszych propozycji tematów obok nazwisk).

Liczba pozycji poddanych analizie w ramach testów modułu wynosiła ok. 1000. Tematyka prac dyplomowych obejmowała zagadnienia związane z kierunkami ścisłymi: fizyką, matematyką i informatyką, w języku polskim. Uzyskane rezultaty przeprowadzonych testów wykazały dużą efektywność implementowanego mechanizmu.

BIBLIOGRAFIA

1. Manning C. D., Prabhakar R., Hinrich S.: Introduction to Information Retrieval. Cambridge University Press, 2008.
2. Beeferman D., Berger A., Lafferty J.: Statistical models for text segmentation. Mach. Learn., Vol. 34(1-3), 1999, s. 177÷210.
3. Lin D.: Automatic retrieval and clustering of similar words. COLING 1998, ACL, 1998, s. 768÷774.
4. Левенштейн В. И.: Двоичные коды с исправлением выпадений, вставок и замещений символов. Доклады Академий Наук СССР, 163 (4), 1965, s. 845÷848.

5. Piasecki M., Broda B.: Semantic similarity measure of Polish nouns based on linguistic features. Business Information Systems 10th International Conference, Poznań, Lecture Notes in Computer Science, Vol. 4439, Springer, 2007.
6. Robertson S.: Understanding Inverse Document Frequency: On theoretical arguments for IDF. Journal of Documentation, Vol. 60, No. 5, 2004, s. 503÷520.
7. Hamming R. W.: Error Detecting and Error Correcting Codes. The Bell System Technical Journal, Vol. XXIX, 1950.
8. Witten I. H., Paynter G. W., Frank E., Gutwin C., Vevill-Manning C. G.: KEA: practical automatic keyphrase extraction. DL'99 Proceedings of the fourth ACM conference on Digital libraries, 1999.
9. Lawrie D., Croft W. B., Rosenberg A.: Finding topic words for hierarchical summarization. SIGIR '01 Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001.
10. Ventura J., da Silva J. F.: Ranking and Extraction of relevant Single Words in Text. InTech, August, 2008.
11. Sarkar K., Nasipuri M., Ghose S.: A new Approach to Keyhprase Extraction Using Neural Networks. International Journal of Computer Science Issues, Vol. 7, Issue 2, No. 3, 2010.
12. Novay L. G., Novay Ch. W., Brussee R.: Thesaurus Based Term Ranking for Keyword Extraction. DEXA '10 Proceedings of the 2010 Workshops on Database and Expert Systems Applications, Computer Society, 2010.

Wpłynęło do Redakcji 10 stycznia 2013 r.

Załączniki

A. Implementacja kodu

Budowa algorytmów oparta została na języku PHP[11] oraz bazie danych MySQL[12].

A.1. Etap periodyczny

Krok 1

```
insert into Terms_matrix (ID_topics, term, tf)
select ID_topics, term, count(*) from tmpTerms group by ID_topics, term;
```

[11] PHP – obiektowy język programowania, którego kod wykonywany jest w czasie rzeczywistym po stronie serwera WWW, w celu generowania stron internetowych.
[12] MySQL – jeden z najpopularniejszych systemów bazodanowych na świecie.

Krok 2

```
set @D = (select count(*) D from Topics);
```

Krok 3

```
create temporary table tmpTM
select ID_topics, term, count(*) cnt from Terms_matrix group by term;
```

Krok 4

```
update Terms_matrix SET df = (select cnt from tmpTM where tmpTM.term =
Terms_matrix.term);

update Terms_matrix SET tf_idf = tf * log(@D/df);
```

A.2. Etap on-line

Krok 1

```
create temporary table tmp_vec_query (term varchar(100));
insert into tmp_vec_query values ('$val1'), ('$val2'), ... , ('$valn');
```
[13]

Krok 2

```
delete from tmp_vec_query where
lower(term) in (select lower(term) from Terms_iness);
```

Krok 3

```
/1/    create temporary table tmp_results as
/2/    select subq_2.*, sum(s_tf_idf) sum_s_tf_idf from (
/3/    select ID_user, name, surname, ID_topics, s_tf_idf from
/4/    (select tmp_vec_query.term, sum(tf_idf) s_tf_idf, ID_topics
/5/    from tmp_vec_query
/6/    left join Terms_matrix on fSim(Terms_matrix.term, tmp_vec_query.term)
   >=0.75 group by ID_topics order by s_tf_idf desc) subq_1
/7/    left join Topics using(ID_topics)
/8/    left join Users using(ID_user)
/9/    left join Employee using(ID_employee)
/10/   ) subq_2
/11/   group by ID_user
/12/   order by sum_s_tf_idf desc, surname, name;
```

Omówienie zapytania:

a) pozycje w kodzie: /4/-/6/ – podzapytanie *subq_1*, zwracające sumę wag poszczególnych terminów zawartych w zapytaniu użytkownika, pogrupowanych względem propozycji tematu (wartość *s_tf_idf*), oraz unikalną wartość *ID_topics*. Połączenie tabeli *Terms_matrix* z tabelą *tmp_vec_query* następuje poprzez spełnienie warunku większej lub równej 0,75 wartości miary podobieństwa terminów[14];

[13] Znak $ oznacza zmienną w języku PHP.

[14] Podana wartość wynika z wcześniejszych badań nad zastosowaniem odległości Levenshteina w analizie danych tekstowych.

b) pozycje w kodzie: /7/-/9/ – do podzapytania *subq_1* dołączone są tabele umożliwiające zwrócenie na bazie podzapytania *sub_1* informacji o wartościach: *ID_user*, *name* i *surname*. Pozycje w kodzie: /3/-/10/ wchodzą w skład podzapytania *subq_2*;
c) pozycje w kodzie: /2/ i /11/ – zsumowanie wartości wag poszczególnych terminów dla zapytania użytkownika względem pracowników naukowo-dydaktycznych (wartość *sum_s_tf_idf*);
d) pozycja w kodzie: /12/ – wartość *sum_s_tf_idf* jest wyznacznikiem rankingu. Pracownik o największej wartości zmiennej rozpoczyna ranking i jest osobą najbardziej prawdopodobną jako promotor lub recenzent.

A.3. Dalsza obróbka danych

Poniższe kody SQL przedstawiają przykład dalszej analizy danych, z uwzględnieniem połowy maksymalnej wartości *sum_s_tf_idf*, jako dolnego pułapu listy pracowników.

Kod SQL, umożliwiający ustawienie dolnej granicy dla rankingu:

```
set @boundary = (select * from (select (max(sum_s_tf_idf) / 2) t from
tmp_results union select '0' t) p where t is not null limit 1);
```

Kod SQL, wyświetlający wynik z uwzględnieniem dolnej granicy dla rankingu:

```
select * from tmp_results where sum_s_tf_idf >= @boundary;
```

B. Uzupełnienie kodu

Kod SQL, przedstawiający funkcję *fSim*, zwracającą miarę podobieństwa terminów:

```
CREATE FUNCTION `fSim`(s1 varchar(255), s2 varchar(255)) RETURNS float
begin
return 1 - ((levenshtein(s1,s2)) / (fmax(length(s1),length(s2))));
end
```

Kod SQL, przedstawiający funkcję *fmax*, zwracającą wartość maksymalną z podanych:

```
CREATE FUNCTION `fmax`(n1 int, n2 int) RETURNS int(11)
begin
if n1 >=n2 then return n1; else return n2; end if;
end
```

Abstract

This paper presents the mechanism of identification and classification of content, based on terms weighted method with inversed document frequency analysis and Levenstein distance algorithm. The mechanism is based on extracting keywords. Methods implemented in mechanism are based on Cartesian product of inversed document frequency and term fre-

quency (formula 1) in combination with Levenshtein distance (formula 2). Cartesian product is using to calculate weights of terms. Originally, Levenshtein distance is used to calculate the number of changes needed to replace one word into second one. The proposed mechanism uses algorithm to calculate the similarity between terms (including one, two or more words) (formula 3).

The proposed mechanism is applied in the analysis of topics and descriptions of selected written works (diploma thesis) to automatic selection of supervisors and reviewers. The mechanism may find application on the universities to automatic selections of supervisors (for students) or reviewers (for employees). Proposed solution was tested at the Faculty of Physics, Mathematics and Computer Science of the Cracow University of Technology.

Adresy

Artur NIEWIAROWSKI: Politechnika Krakowska, Wydział Fizyki, Matematyki i Informatyki, ul. Podchorążych 1, 30-084 Kraków, Polska, aniewiarowski@pk.edu.pl.
Marek STANUSZEK: Politechnika Krakowska, Wydział Fizyki, Matematyki i Informatyki, ul. Warszawska 24, 31-155 Kraków, Polska, marek.stanuszek@pk.edu.pl.