OCENA PODOBIEŃSTWA DOKUMENTÓW XML

Bartłomiej Jabłoński[1], Maciej Zakrzewicz[2]

[1]Uniwersytet Łódzki, Wydział Matematyki
bartek@math.uni.lodz.pl
[2]Politechnika Poznańska, Wydział Informatyki i Zarządzania
mzakrz@cs.put.poznan.pl

Abstrakt. Jednym z fundamentalnych zagadnień w dziedzinie semistrukturalnych baz danych jest realizacja zapytań przybliżonych, w których użytkownik poszukuje obiektów najbardziej zbliżonych do zadanego wzorca. Zapytania przybliżone opierają się na wartościowaniu miary podobieństwa, której definicja może subiektywna dla użytkownika. W artykule proponujemy nową metodę oceny podobieństwa dokumentów XML opartą na znajdowaniu najkrótszej ścieżki w grafie wyznaczonym przez przekształcenia XSLT definiowane przez użytkownika. Omawiana metoda umożliwia porównywanie zarówno treści, jak i struktury dokumentów XML.

1. Wprowadzenie

Rosnąca popularność języka XML [Con][Har99] w dziedzinie konstrukcji semistrukturalnych baz danych stanowi inspirację dla szeregu prac badawczych związanych z przetwarzaniem danych, których struktura nie jest jednorodna i nie jest znana użytkownikowi. Koncepcja zautomatyzowanych transformacji XSLT [Conb] pozwala na definiowanie reguł, zgodnie z którymi mogą być realizowane złożone rekurencyjne przekształcenia dokumentów składowanych w bazie danych. Dotychczas zaproponowane rozwiązania języków zapytań dla baz danych dokumentów XML, m.in. XPath [CD99] i XQuery [CFR+01], umożliwiają realizację zapytań opartych o powtarzalne fragmenty struktury, które są odwzorowywane na dane źródłowe w sposób dokładny. Niestety, rozwiązania te nie implementują koncepcji *zapytań przybliżonych* (ang. approximated queries, similarity queries), w których użytkownik poszukuje obiektów/dokumentów niekoniecznie identycznych, lecz tylko najbardziej zbliżonych/podobnych do

przedstawionego wzorca. Brak takiego elementu funkcjonalnego istotnie ogranicza stosowalność semistrukturalnych baz danych.

Zapytania przybliżone odgrywają kluczową rolę nie tylko w systemach wyszukiwania informacji (ang. information retrieval), ale także w coraz popularniejszych środowiskach eksploracji danych (ang. data mining), które realizują zadania grupowania (ang. clustering) i klasyfikowania (ang. classification) obiektów. Podstawą wielu metod eksploracji danych jest ocena podobieństwa dokumentów w celu agregacji dokumentów najbardziej do siebie zbliżonych lub w celu przydzielenia nowego dokumentu do jednej z wcześniej zdefiniowanych klas, której elementy są do niego najbardziej zbliżone. Zapytania przybliżone są także wykorzystywane przez algorytmy poszukiwania najbliższego sąsiada (ang. nearest neighbour search), w których konstruowane są rankingi *n* obiektów najbardziej podobnych do wzorca.

Podstawowym problemem występującym w przetwarzaniu zapytań przybliżonych jest ewaluacja podobieństwa dwóch obiektów/dokumentów, rozumiana jako wyliczenie pewnej miary, której wartość reprezentuje „stopień" lub „siłę" tego podobieństwa. Miara ta jest niestety subiektywna, zależna od punktu widzenia użytkownika-obserwatora oraz od specyfiki branży, którą opisują dokumenty [LN91][ZR94]. Przykładowo, dla jednego użytkownika dwa obrazy olejne są podobne, ponieważ przedstawiają tę samą martwą naturę, natomiast dla innego użytkownika mogą być one zupełnie odmienne gdyż zostały wykonane inną techniką. Warto zauważyć, że użytkownik, który w sensie potocznym uzasadnia tezę o podobieństwie dwóch konkretnych obiektów, najczęściej wymienia pewne elementy prostsze, które są identyczne lub podobne (rekurencja) w obu rozpatrywanych obiektach. Oznacza to, że dokonuje on na obiektach przekształceń polegających na wyabstrahowaniu poszczególnych elementów składowych i przeprowadzaniu na nich operacji porównań[1]. Ponieważ dokumenty XML mogą być wykorzystywane do reprezentowania dowolnych struktur danych, to z punktu widzenia naszych badań ten aspekt subiektywności oceny podobieństwa jest niezwykle istotny: inne będą kryteria podobieństwa dla dokumentów reprezentujących nutowy zapis muzyczny (MusicML), inne – dla dokumentów opisujących formuły matematyczne (MathML), a jeszcze inne – dla dokumentów zawierających obrazy grafiki wektorowej (VML). Ponadto, w przeciwieństwie do metod przetwarzania danych o typach prostych, w ocenie podobieństwa dokumentów XML powinny uczestniczyć nie tylko wartości, ale także ich struktura.

Miarę podobieństwa dokumentów XML można rozumieć jako pewną funkcję *w*, która każdej parze dokumentów ze zbioru *X* przypisuje nieujemną liczbę rzeczywistą:

$$w : X \times X \rightarrow R^{+} \tag{1}$$

Najprostszym przykładem będzie funkcja $w(x_1,x_2)$, gdzie $x_1,x_2 \in X$, która dla identycznych dokumentów przyjmuje wartość 0, zaś w przeciwnym przypadku wartość 1:

[1] Transformaty mogą również opisywać te elementy, których w porównywanych dokumentach nie ma! Elementy te mogą być wytworem naszej wyobraźni, która łatwo ulega złudzeniom [LN91], [ZR94]

$$w(x_1, x_2) = \begin{cases} 0 & \text{gdy } x_1 = x_2 \\ 1 & \text{gdy } x_1 \neq x_2 \end{cases} \quad (2)$$

Tak zdefiniowana funkcja podobieństwa odpowiada w ogólnym zarysie implementacjom podobieństwa realizowanym w językach XPath i XQuery (w porównaniach występuje cała rodzina operatorów, których rezultatem jest prawda lub fałsz logiczny, czyli odpowiednik zera lub jedynki).

W niniejszej pracy proponujemy wykorzystanie pojęcia transformat XML do oceny podobieństwa dwóch dokumentów. Transformatę XML traktujemy jako dowolną funkcję przekształcającą dokument XML w inny dokument XML. Funkcja taka może być wyrażona w języku transformacji XSLT. Podobieństwo definiujemy jako najkrótszą ścieżkę w grafie wyznaczonym przez kolejno wykonywane transformacje umożliwiające przekształcenie jednego porównywanego dokumentu XML w drugi. W celu umożliwienia implementacji „subiektywności" pojęcia podobieństwa, wprowadzamy pojęcie profilu użytkownika, stanowiącego zbiór transformat z przypisanymi wagami, gdzie każda transformata opisuje sposób, którym wg użytkownika dwa dokumenty mogą się różnić. Waga takiego przekształcenia odzwierciedla miarę podobieństwa elementarnej operacji, natomiast wagę ścieżki oblicza się jako sumę wag poszczególnych przekształceń. W procesie oceny podobieństwa można wyróżnić cztery scenariusze: (1) badane dokumenty są identyczne – wtedy zgodnie z definicją przypisujemy im zerową wartość miary podobieństwa, (2) dokumenty są różne i udało się znaleźć ścieżkę w grafie, lecz suma wag przekształceń wynosi zero (czyli wszystkie zastosowane przekształcenia mają zerowe wagi), wtedy dokumenty traktujemy jako tożsame, czyli nie różniące się w sposób istotny dla użytkownika, (3) dokumenty są różne i waga ścieżki w grafie jest większa od zera – waga ścieżki w tym przypadku jest wartością miary podobieństwa, (4) ścieżka w grafie nie istnieje – w tym przypadku traktujemy dokumenty XML jako niepodobne i przypisujemy im nieskończoną wartość miary podobieństwa.

Zaproponowana metoda spełnia cztery przyjęte przez nas założenia: (Z1) zgodność z potocznie rozumianym pojęciem podobieństwa, (Z2) uwzględnianie preferencji branżowych w sposób praktyczny dla użytkownika, (Z3) możliwość zastosowania w przetwarzaniu zapytań przybliżonych, (Z4) łatwość implementacji w środowisku systemu zarządzania bazą danych. Stosowana przez nas miara podobieństwa bazuje na funkcjach, które wykorzystują całe spektrum nieujemnych liczb rzeczywistych zgodnie z zasadą: im większa wartość funkcji, tym większa „odległość" między dokumentami, czyli tym mniej dokumenty te są podobne do siebie. Od takich funkcji oczekuje się tylko, że będą spełniały powszechnie uznane postulaty podobieństwa: (P1) dla dokumentów tożsamych (i tylko takich) wartością funkcji jest zero (warunek tożsamości), (P2) kolejność argumentów funkcji nie ma znaczenia (warunek symetrii). Nie wymagamy natomiast spełnienia warunku trójkąta, który pozwoliłby zbudować metryczną przestrzeń dokumentów XML, gdzie pojęcie odległości byłoby realizowane w oparciu o miarę podobieństwa. Brak takiego wymagania nie kłóci się jednak z doświadczeniem i oczekiwaniami użytkowników.

1.1. Przegląd dotychczasowych badań

Semistrukturalne bazy danych są od wielu lat obszarem intensywnych badań naukowych [Abi97][Bun97]. XML jest popularnym formatem zapisu danych w takich bazach [Con][Har99]. Wiele prac było związanych z językami zapytań do baz danych gromadzących informacje w formacie XML. Najpopularniejsze rozwiązania obejmują: XML-QL [DFF+99a], XQL [Rob99], Lorel [AQM+97], StruQL [FFK+98], UnQL [BDHD96], XPath [CD99], XQuery [CFR+01]. Opracowano również szereg podejść uogólnionych, opartych na wyszukiwaniu drzew [ACLS01][ACS02][FGGP01] [GT87][JLS+99][NS00] i grafów [AB99][AQM+97] [Bun97][CM90][Gut94][GPG90] [MMM97][MW95]

XSLT [Bos98][Conb][Cla99b] jest rekurencyjnym językiem konstrukcji arkuszy reguł formatujących rekomendowanym przez W3C [Con]. Pierwotnie, zastosowania XSLT skupiały się wokół implementacji warstwy prezentacji dokonującej konwersji dokumentów XML do HTML. Dziś coraz powszechniej wykorzystuje się XSLT jako podstawowe narzędzie transformacji dokumentów XML w inne dokumenty XML. W pracy [BMN00] zawarto interesujący opis modelu przetwarzania wykorzystywanego przez XSLT.

Język XML został również wykorzystany do konstrukcji wielu branżowych języków opisu dokumentów semistrukturalnych. Język MathML umożliwia zapis formuł matematycznych [Conc], SVG – wektorowych obrazów graficznych [Cond], MusicML – utworów muzycznych [TCF].

W dziedzinie przetwarzania dokumentów tekstowych opracowano wiele mechanizmów zapytań przybliżonych, w większości opartych na pojęciu Edit Distance [Lev65]. Edit Distance to miara podobieństwa dwóch ciągów znakowych wyznaczana jako liczba operacji wstawienia i usunięcia pojedynczych znaków w celu przekształcenia jednego porównywanego ciągu w drugi.

2. Ocena podobieństwa dokumentów XML

Proponowana metoda oceny podobieństwa dwóch dokumentów XML, x_A i x_B, polega na wykorzystaniu predefiniowanego zbioru ważonych transformat XSLT do znalezienia takiego skończonego ciągu transformacji $t_1, t_2, \ldots, t_n$ że $x_B = t_n(\ldots t_2(t_1(x_A)))$ lub $x_A = t_n(\ldots t_2(t_1(x_B)))$. Suma wag transformat użytych w ciągu jest traktowana jako miara podobieństwa dokumentów x_A i x_B. W przypadku, gdy istnieje wiele alternatywnych ciągów transformacji, wówczas wybierany jest ten, dla którego suma wag transformat jest najmniejsza. Łatwo zauważyć, że tak przyjęta miara podobieństwa spełnia nasze postulaty (P1) i (P2). Jeżeli przyjąć pewne ograniczenia co do możliwości definiowania transformat, to również będą spełnione założenia (Z3) i (Z4) (zagadnienie to będzie przedmiotem dalszej dyskusji).

2.1. Definicje formalne

Niech X oznacza zbiór wszystkich dokumentów XML reprezentowanych w postaci kanonicznej C14N [Con], a T: $X \to X$ oznacza zbiór wszystkich przekształceń dokumentów XML. Niech $T' \subset T$ będzie zbiorem przekształceń, które opisują sens pojęcia podobieństwa dla obserwatora. Niech I oznacza zbiór przekształceń tożsamościowych, tzn.:

$$I = \{i \in T' : \forall \mathrm{x} \in \mathrm{X} \;\; i(\mathrm{x}) = x\} \tag{3}$$

Niech W będzie funkcją wagi przekształceń:

$$W : T' \to R^{+}$$

Aby dokumenty tożsame były traktowane jako identyczne, funkcja W musi spełniać warunek:

$$\forall i \in I \;\; w(i) = 0 \tag{4}$$

Niech *profil* oznacza zbiór ważonych przekształceń, reprezentujący preferencje obserwatora:

$$P = \{(t, w(t)) : t \in T', w \in W\} \tag{5}$$

W dalszej części pracy, przez p^t i p^w będziemy rozumieli odpowiednio pierwszy i drugi człon pary elementu $p \in P$. W ogólnym zarysie, funkcja W powinna być tak dobrana, aby dla przekształceń mocno różnicujących zwracać wartości duże, a dla przekształceń mało różnicujących - małe.

Proponowana przez nas metoda oceny podobieństwa dokumentów XML wymaga znalezienia ciągów transformacji przekształcających jeden porównywany dokument w druki. W pierwszym kroku definiujemy zbiór ciągów, wśród których będziemy szukać nas rozwiązania:

$$Z_P = \left\{ \{p_i\}_{i=0}^{n} : p_i \in P, n \in N, n < +\infty \right\} \tag{6}$$

Elementami tego ciągu są ważone transformacje, będące składnikami zdefiniowanego profilu.

2.2. Ocena podobieństwa metodą ścieżki bezpośredniej

Gdy do dokumentu $x \in X$ zastosujemy pewien ciąg transformacji ze zbioru Z_P, wówczas otrzymamy odpowiadający mu ciąg S_P dokumentów tworzących ścieżkę rozpoczynającą się od x:

$$S_P(x) = \left\{ \{x_i\}_{i=0}^{n} : x_0 = x, x_i = p_{i-1}(x_{i-1}), \text{ gdzie } x_i \in X, p_i \in Z_P \right\} \quad (7)$$

Ponieważ nas interesują tylko takie ścieżki, które łączą porównywane dokumenty, to zdefiniujemy również zbiór ścieżek rozpoczynających się od dokumentu x_1 a kończących na dokumencie x_2:

$$S_P(x_1, x_2) = \{ s \in S_P(x_1) : s_{last} = x_2 \} \quad (8)$$

Niech $x_1, x_2 \in X$ oraz $s \in S_P(x_1, x_2) = \{p_i\}$. Waga ścieżki to suma wag wszystkich przekształceń użytych do zbudowania tej ścieżki:

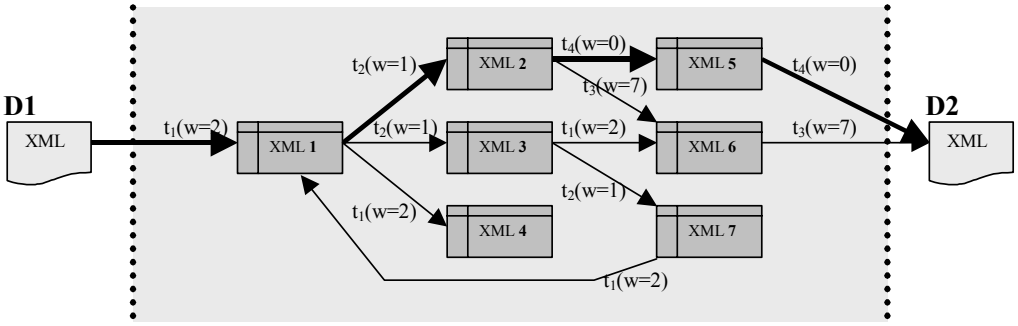
$$w(x_1, x_2, s) = \sum_i p_i^w \quad (9)$$

Podobieństwo dwóch dokumentów x_1 i x_2 definiujemy jako wagę najlżejszej ścieżki w grafie łączącej te dokumenty:

$$w(x_1, x_2) = \begin{cases} 0 & x_1 = x_2 \\ \min\left\{ \min_{s \in S_P(x_1, x_2)} \{w(x_1, x_2, s)\}, \min_{s \in S_P(x_2, x_1)} \{w(x_1, x_2, s)\} \right\} & \text{dla pozostalych przypadków} \\ +\infty & S_P(x_1, x_2) = \varnothing \wedge S_P(x_2, x_1) = \varnothing \end{cases} \quad (10)$$

W celu zilustrowania podstaw techniki oceny podobieństwa metodą ze ścieżką bezpośrednią, na rys. 1 przedstawiono graf zawierający ścieżki służące porównaniu dokumentów D1 i D2. Do budowy ścieżek użyto czterech transformat: t1, t2, t3 i t4, o wagach odpowiednio 2, 1, 7, 0. Aby poprawić czytelność, na rysunku pominięto transformacje-niezmienniki. W grafie można znaleźć 3 ścieżki: D1→XML1→XML2→ →XML5→D2, D1→XML1→XML2→XML6→D2, D1→XML1→XML3→XML6→D2), z których pierwsza posiada najmniejszą wagę w=2+1+0+0=3. Podobieństwo dokumentów D1 i D2 wynosi zatem 3.

Na szczególną uwagę zasługuje fakt, iż istnieje przekształcenie (pomiędzy dokumentem 7 i 1), które tworzy cykl i generuje nieskończoną liczbę możliwych przejść od dokumentu D1 do D2. Ponieważ jednak, każda taka pętla wpływa na powiększanie wagi ścieżki, to może zostać pominięta.



Rys. 1. Przykładowy graf przekształceń dokumentów XML – metoda ścieżki bezpośredniej

2.3. Ocena podobieństwa metodą ścieżki bezpośredniej - przykład

Wyznaczmy wartość miary podobieństwa dokumentów MathML opisujących ułamki zwykłe. Rozważmy 3 następujące dokumenty XML oraz 3 transformacje XML przedstawione w tabeli:

3/12

```
<m:math>
  <m:apply>
    <m:divide/>
    <m:cn>3</m:cn>
    <m:cn>12</m:cn>
  </m:apply>
</m:math>
```

1/4

```
<m:math>
  <m:apply>
    <m:divide/>
    <m:cn>1</m:cn>
    <m:cn>4</m:cn>
  </m:apply>
</m:math>
```

4/12

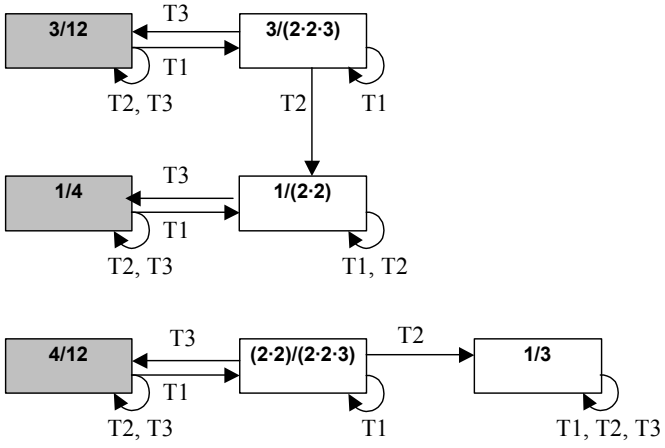
```
<m:math>
  <m:apply>
    <m:divide/>
    <m:cn>4</m:cn>
    <m:cn>12</m:cn>
  </m:apply>
</m:math>
```

Nazwa	Waga	Opis	Przykład	
			We	WY

T1	3	Znajduje liczby złożone i rozkłada je na czynniki pierwsze.	`<m:math>`
			`  <m:apply>`
			`  <m:divide/>`
			`  <m:cn>3</m:cn>`
			`  <m:cn>12</m:cn>`
			`  </m:apply>`
			`</m:math>`
				`<m:math>`
				`  <m:apply>`
				`  <m:divide/>`
				`  <m:cn>3</m:cn>`
				`  <m:apply>`
				`  <m:times/>`
				`  <m:cn>2</m:cn>`
				`  <m:cn>2</m:cn>`
				`  <m:cn>3</m:cn>`
				`  </m:apply>`
				`  </m:apply>`
				`</m:math>`

T2	1	Znajduje wszystkie liczby występujące zarówno w liczniku, jak i w mianowniku i usuwa je skracając tym samym ułamek.	<m:math> <m:apply> <m:divide/> <m:apply> <m:times/> <m:cn>2</m:cn> <m:cn>3</m:cn> <m:cn>7</m:cn> </m:apply> <m:apply> <m:times/> <m:cn>2</m:cn> <m:cn>2</m:cn> <m:cn>3</m:cn> </m:apply> </m:apply> </m:math>	<m:math> <m:apply> <m:divide/> <m:cn>7</m:cn> <m:cn>2</m:cn> </m:apply> </m:math>
T3	0	Znajduje liczby zapisane w postaci iloczynu i zapisuje je w postaci wymnożonej.	<m:math> <m:apply> <m:divide/> <m:cn>1</m:cn> <m:apply> <m:times/> <m:cn>2</m:cn> <m:cn>2</m:cn> <m:cn>3</m:cn> </m:apply> </m:apply> </m:math>	<m:math> <m:apply> <m:divide/> <m:cn>1</m:cn> <m:cn>12</m:cn> </m:apply> </m:math>

Stosując do porównywanych dokumentów wszystkie transformacje uzyskujemy dokumenty, do których możemy ponownie zastosować transformacje itd. Proces ten kontynuujemy do chwili gdy przestanie przybywać nowych dokumentów. Dla omawianego przypadku doprowadzi to do powstania następującego grafu:



Na podstawie powyższego grafu odczytujemy wartość miary podobieństwa dokumentów 3/12 i 1/4– wynosi ona $w\left(3/12, 1/4\right) = 3+1+0 = 4$. Zauważmy, że z powodu braku ścieżki pomiędzy dokumentem 4/12 a pozostałymi badanymi dokumentami, orzekamy o nieskończonej wartości miary podobieństwa dokumentów 4/12 a 3/12 i 4/12 a 1/4.

Zauważmy jednocześnie, że: dla dużej liczby przekształceń graf może być bardzo rozbudowany, przekształcenia mogą być nieodwracalne (np. T2), mogą występować cykle, graf może nie być spójny

2.4. Ocena podobieństwa metodą ścieżki uogólnionej

Ścieżką uogólnioną S^* nazywamy dowolną konkatenację ścieżek prostych – konkatenacja może zachodzić tylko na końcach ścieżek:

$$S_P^*(x) = \left\{ \begin{array}{l} \{s_i\}_{i=0}^n : s_i \in S_P,\ \ s_{0,0} = x,\ s_{i,last} = s_{i+1,0} \vee \\ \vee\ s_{i,last} = s_{i+1,last} \vee s_{i,0} = s_{i+1,0} \vee s_{i,0} = s_{i+1,last} \end{array} \right\} \tag{11}$$

Najbardziej interesujący w tym wzorze jest warunek $s_{i,last} = s_{i+1,last}$. Opisuje on bowiem te konkatenacje dla których spotykają się zwroty kierunków przekształceń.. Zdefiniujmy zatem ścieżki, dla których liczba takich złączeń jest nie większa niż k:

Niech $K : S_P^* \to N$ będzie funkcją określająca liczbę konkatenacji:

$$K(s) = \overline{\overline{\left\{(s_i, s_{i+1}) : s_i, s_{i+1} \in S_P \wedge s_{i,last} = s_{i+1,last}\right\}}} \tag{12}$$

Zbiór ścieżek uogólnionych rozpoczynających się od dokumentu x oraz zbiór ścieżek łączących dwa dokumenty x_1 i x_2 przedstawiamy następująco:

$$S_P^k(x) = \left\{ s \in S_P^*(x) : K(s) \le k \right\} \tag{13}$$

$$S_P^k(x_1, x_2) = \left\{ s \in S_P^k(x_1) : s_{last,last} = x_2 \right\} \tag{14}$$

Na potrzeby dalszych rozważań przyjęto założenie, że $k \in \{0, 1\}$. Dla $k = 0$ otrzymujemy ścieżki bezpośrednie.

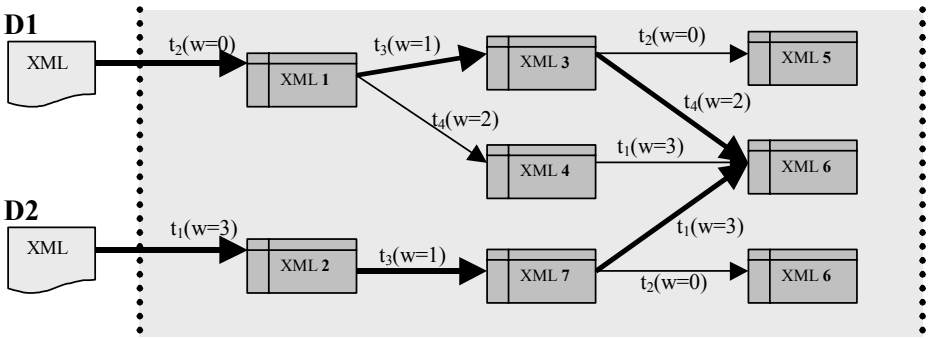
Niech $x_1, x_2 \in X$ oraz $s \in S_P^k(x_1, x_2) = \{p_i\}$. Waga ścieżki uogólnionej jest sumą wag ścieżek składowych:

$$w^*(x_1, x_2, s) = \sum_i w(x_1, x_2, s_i) \tag{15}$$

Podobieństwo dwóch dokumentów x_1 i x_2 definiujemy jako wagę najlżejszej ścieżki uogólnionej w grafie łączącej te dokumenty:

$$w^*(x_1, x_2) = \begin{cases} 0 & x_1 = x_2 \\ \min\left\{ \min\limits_{s \in S_P^K(x_1, x_2)} \left\{ w^*(x_1, x_2, s) \right\}, \min\limits_{s \in S_P^K(x_2, x_1)} \left\{ w^*(x_2, x_1, s) \right\} \right\} & \text{dla pozostalych przypadków} \\ +\infty & S_P^k(x_1, x_2) = \varnothing \wedge S_P^k(x_2, x_1) = \varnothing \end{cases}$$

W celu zilustrowania podstaw techniki oceny podobieństwa metodą ze ścieżką uogólnioną, na rys. 2 przedstawiono graf zawierający ścieżki służące porównaniu dokumentów D1 i D2. Do budowy ścieżek użyto czterech transformat: t1, t2, t3 i t4, o wagach odpowiednio 2, 1, 7, 0. Zauważmy, że mimo iż nie istnieje taki zestaw transformacji, który w sposób bezpośredni wyznaczy ścieżkę pomiędzy badanymi dokumentami, to istnieje dokument – wspólna baza – do której prowadzą ścieżki z obu badanych dokumentów. Jeżeli użytkownik dopuszcza możliwość konkatenowania ścieżek ($k = 1$), to przedstawiona na rysunku linia pogrubiona wyznacza ścieżkę o najmniejszej wadze. Podobieństwo dokumentów D1 i D2 wynosi zatem 10.



Rys.2. Przykładowy graf przekształceń dokumentów XML – metoda ścieżki uogólnionej

2.5. Ocena podobieństwa metodą ścieżki uogólnionej - przykład

Wyznaczmy wartość miary podobieństwa dokumentów MathML opisujących ułamki zwykłe. Rozważmy 4 następujące dokumenty XML oraz 3 transformacje XML z przykładu 2.3:

3/12

```
<m:math>
  <m:apply>
    <m:divide/>
    <m:cn>3</m:cn>
    <m:cn>12</m:cn>
  </m:apply>
</m:math>
```

1/4

```
<m:math>
  <m:apply>
    <m:divide/>
    <m:cn>1</m:cn>
    <m:cn>4</m:cn>
  </m:apply>
</m:math>
```

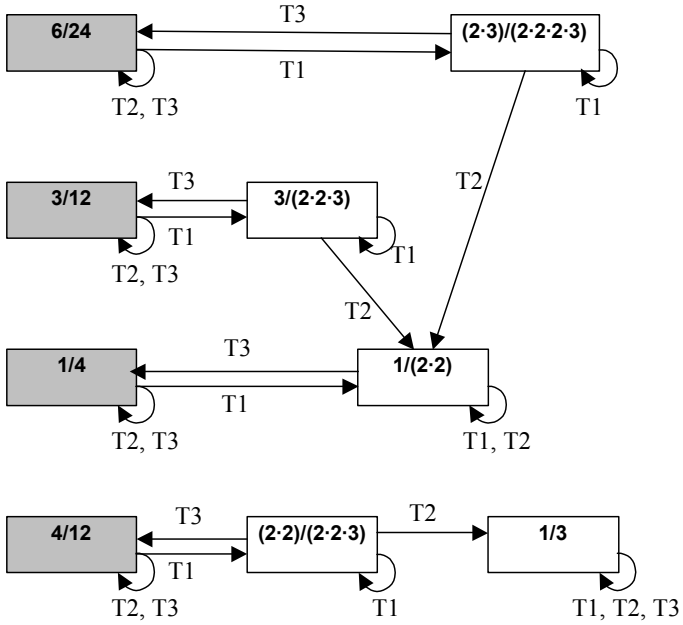
4/12

```
<m:math>
  <m:apply>
    <m:divide/>
    <m:cn>4</m:cn>
    <m:cn>12</m:cn>
  </m:apply>
</m:math>
```

6/24

```
<m:math>
  <m:apply>
    <m:divide/>
    <m:cn>6</m:cn>
    <m:cn>24</m:cn>
  </m:apply>
</m:math>
```

Stosując do porównywanych dokumentów wszystkie transformacje uzyskujemy dokumenty, do których możemy ponownie zastosować transformacje itd. Proces ten kontynuujemy do chwili gdy przestanie przybywać nowych dokumentów. Dla omawianego przypadku doprowadzi to do powstania następującego grafu:



Na podstawie powyższego grafu możemy odczytać wartość miary podobieństwa dokumentów 6/24 i 1/4 oraz 3/12 i 1/4. Jednak w kontekście omawianej metody ze ścieżką uogólnioną, najbardziej interesującym przypadkiem jest podobieństwo dokumentów 6/24 i 3/12, którego miara wynosi $w\left(\frac{6}{24},\frac{3}{12}\right)=3+1+1+3=8$. Fakt, że jej wartość jest większa, niż wartość miary podobieństwa dla $w\left(\frac{6}{24},\frac{1}{4}\right)=w\left(\frac{3}{12},\frac{1}{4}\right)=4$, prowadzi do konkluzji, że dokumenty 6/24 i 1/4 są do siebie „bardziej" podobne, aniżeli dokumenty 6/24 i 3/12.

3. Podsumowanie

W artykule przedstawiliśmy nową metodę oceny podobieństwa dokumentów XML, opartą na znajdowaniu najkrótszej ścieżki w grafie wyznaczonym przez transformaty XML definiowane przez użytkownika branżowego. Omawiana metoda umożliwia porównywanie zarówno treści, jak i struktury dokumentów XML. Jej głównymi zastosowaniami praktycznymi mogą być: realizacja zapytań przybliżonych w semistrukturalnych bazach danych, eksploracja danych semistrukturalnych metodą grupowania, eksploracja dokumentów semistrukturalnych metodą klasyfikacji.

Komentarza wymaga sposób praktycznej implementacji przedstawionej metody. Aby wyznaczyć wartość miary podobieństwa pomiędzy dwoma (lub więcej) dokumentami, w pierwszej kolejności należy zbudować graf przekształceń. Niestety, liczba węzłów i krawędzi w takim grafie rośnie wykładniczo wraz ze wzrostem liczby transformacji zdefiniowanych w profilu. Zważywszy na fakt, że transformacje XSLT są zwykle bardzo kosztowne czasowo, budowanie takiego grafu na potrzeby realizacji tylko jednego zapytania jest nieopłacalne. Warto rozpatrywać model, w którym dokumenty pośrednie są trwale przechowywane w postaci zmaterializowanej. Dla przyspieszenia procesu przeszukiwania, dla materializowanych dokumentów pośrednich zastosować można znane techniki indeksowania. Innym, nie mniej istotnym, problemem jest przeszukiwanie grafu przekształceń w celu znalezienia najkrótszej ścieżki. Graf przekształceń może nie być skończony – może składać się z nieskończenie wielu dokumentów pośrednich. W takim przypadku konieczne jest wprowadzenie warunku stopu.

Bibliografia

[AB99] S. Abiteboul, P. Buneman, D. Suciu, Data on the web: from relations to semistructured data and XML, Morgan Kaufman, 1999

[Abi97] S. Abiteboul. Querying semi-structured data. In Proceedings of the International Conference on Database Theory, Delphi, Greece, January 1997

[ACLS01] S. Amer-Ahia, S. Cho, L.V.S. Lakshmanan, D. Srivastava, Minimization of tree pattern queries, Proc. of ACM SIGMOD 2001

[ACS02] S. Amer-Ahia, S. Cho, D. Srivastava, Tree pattern relaxation, Proc. Int'l Conference on Extending Database Technology, 2002

[AQM+97] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. L. Wiener. The lorel query language for semistructured data. International Journal on Digital Libraries, 1(1):68-88, 1997.

[BDHD96] G.J. Bex, S. Maneth, F. Neven, A formal model for an expressive fragment of XSLT, Lecture Notes in Computer Science 1861, 2000.

[Bos98] A. Bosworth, A proposal for an XSL query language, http://www.w3.org/TandS/QL/QL98/pp/microsoft-extensions.html

[Bun97] P. Buneman. Semi-structured data. In Proceedings of the Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Tuscon, AZ, May 1997

[CD99] J. Clark and S. DeRose. XML path language (XPath), version 1.0. http://www.w3.org/TR/xpath, November 1999. W3C Recommendation

[CFR+01] D. Chamberlin, D. Florescu, J. Robie, J. Simon, and M. Stefanescu. XQuery: A Query Language for XML W3C working draft. Technical Report WD-xquery-20010215, World Wide Web Consortium, 2001

[Cla99b] James Clark. XSL Transformations (XSLT) Version 1.0. W3C Recommendation. http://www.w3.org/TR/xslt, November 1999

[CM90] M.P. Consens, A.O. Mendelzon, GraphLog: a visual formalism for real life recursion, Proc. ACM SIGMOD – ACM SIGART Symposium on Principles of Database Systems, 1990

[Con] World Wide Web Consortium. Extensible Markup Language (XML), http://www.w3.org/XML

[Conb] W3C, Extensible Stylesheet Language1.0, Recommendation, http://www.w3.org/TR/1998/REC-xml-19980210, technical report, The World Wide Consortium, Jan. 2000

[Conc] Mathematical Markup Language (mathml) 1.01 specification, http://www.w3.org/TR/REC-MathML, 1999

[Cond] Scalable Vector Graphics, http://www.w3.org/TR/SVG

[Cone] Extensible Markup Language XML, http://w3c.org/xml

[CWI] GraphXML, http://www.cwi.nl/InfoVisu/GraphXML

[DFF+99a] A. Deutsch, M. fernandez, D. Florescu, A. Levy, D. Maier, and D. Suciu, Querying XML data. Data Engineering Bulletin, 22(3):10{18, 1999

[FGGP01] A. Ferro, D. Gallo, R. Giugno, A. Pulvirenti, Best-match retrieval for structured images, IEEE Trans. on Pattern Analysis and Machine Intelligence, 23 (7) 707-718, 2001

[FKK+98] M. F. Fernandez, D. Florescu, J. Kang, A. Y. Levy, and D. Suciu. Catching the boat with strudel: Experiences with a web-site management system. In L. M. Haas and A. Tiwary, editors, SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, pages 414-425, ACM Press, 1998.

[GPG90] M. Gyssens, J. Paredaens, D.V. Gucht, A graph-oriented object database model, Proc. ACM SIGMOD – ACM SIGART Symposium on Principles of Database Systems, 1990

[GT87] G.H. Gonnet, F.W. Tompa, Mind your grammar: a new approach to modeling text, Proc. VLDB 87, 1987

[Gut94] R.H. Guting, GraphDB: modeling and querying graphs in databases, Proc. VLDB 1994, 1994

[Har99] E. R. Harold: XML Bible, IDG Books Worldwide, 1999.

[JLS+99] H.V. Jagadish, L.V.S. Lakshmanan, T. Milo, D. Srivastava, D. Vista, Querying network directories, Proc. ACM SIGMOD 1999, 1999

[LN91] P.H. Lindsay, D.A. Norman Procesy przetwarzania informacji u człowieka., PWN 1991

[MMM97] A.O. Mendelzon, G.A. Mihaila, T. Milo, Querying the world wide web, International Journal on Digital Libraries 1(1), 54-67, 1997

[MW95] A.O. Mendelzon, P.T. Wood, Finding regular simple paths in graph databases, SIAM J. Comput., 24(6), 1235-1258, 1995

[NS00] F. Neven, T. Schwentick, Expressive and efficient pattern languages for tree-structured data, Proc. ACM SIGMOD – ACM SIGART Symposium on Principles of Database Systems, 2000

[Rob99] J. Robie. The design of XQL. http://www.texcel.no/whitepapers/xql-design.html, 1999.

[TCF] MusicML, http://www.tcf.nl/3.0/musicml

[ZR94] P.G. Zimbardo, F.L. Ruch, Psychologia i życie., PWN 1994