STUDIA INFORMATICA 2003
Volume 24 Number 2A (53)

Leszek BORZEMSKI, Piotr ŁOPATKA
Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów

ZASTOSOWANIE SYSTEMU IBM INTELLIGENT MINER FOR TEXT DO WYSZUKIWANIA INFORMACJI W INTERNECIE

Streszczenie. Celem projektu jest zademonstrowanie możliwości budowy wyszukiwarki internetowej na bazie systemu IBM Intelligent Miner for Text. W opracowanej wyszukiwarce SearchSystem zaimplementowano zaawansowane metody analizy leksykalnej, a mianowicie ekspansję poprzez synonimy oraz analizę dźwiękową zapytań. System porównano z wyszukiwarką Google.

Słowa kluczowe: wyszukiwanie informacji w Internecie, eksploracja tekstu.

USING "IBM INTELLIGENT MINER FOR TEXT" IN SEARCHING INFORMATION FROM THE INTERNET

Summary. The aim of the project was to find out how the IBM Intelligent Miner for Text can be used to develop a "Google" - like search machine. The system called SearchSystem has been developed and evaluated based on chosen functions.

Keywords: information searching in Internet, text mining.

1. Wprowadzenie

System WWW (ang. *World Wide Web*) tworzy największą składnicę wiedzy. Przy wyszukiwaniu informacji w sieci WWW wykorzystujemy dwa podstawowe narzędzia, a mianowicie: systemy wyszukiwawcze (wyszukiwarki internetowe) do wyszukania i skatalogowania informacji oraz przeglądarki internetowe do przekazania zapytania do wyszukiwarki i do prezentacji otrzymanych dokumentów. Wyszukiwarki zbierają głównie informacje o dokumentach z sieci WWW, ale często również pracują z zasobami sieci Internet niedostępnymi przez interfejs WWW oraz z zasobami zawartymi w sieciach intranetowych. Do najpopularniejszych wyszukiwarek należą wyszukiwarki ogólnego

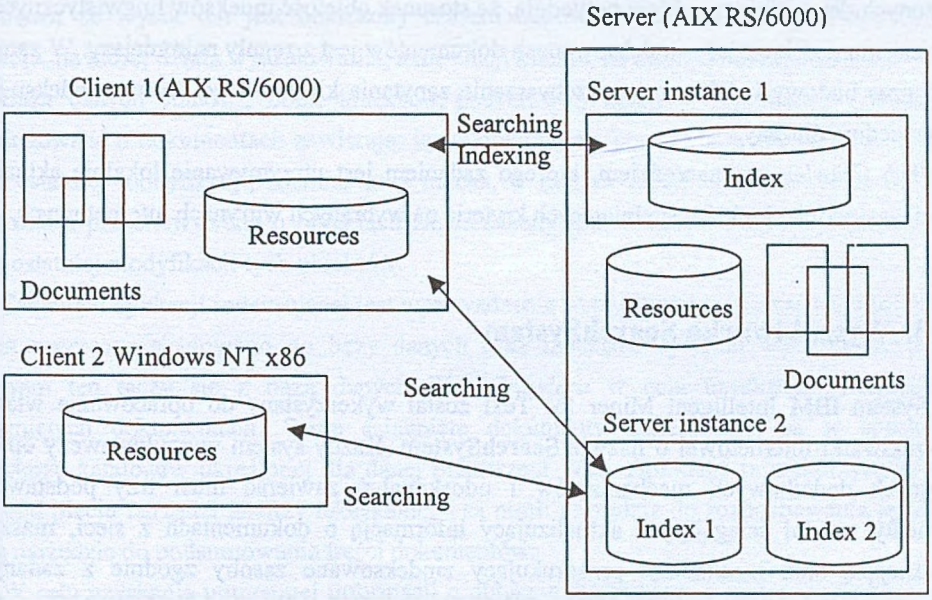
przeznaczenia Google oraz AltaVista. Istnieje też wiele wyszukiwarek mniej znanych, które bardzo często są dedykowane do współpracy ze specjalnymi zasobami bądź z określonymi grupami użytkowników. Wyszukiwarki dedykowane mogą być wyposażone w pewne specjalistyczne funkcje wyszukiwawcze, które tworzą nowe możliwości wyszukiwania informacji w Internecie. Wyszukiwarki są tworzone również pod kątem ich wykorzystania we własnych ośrodkach WWW. Tworzenie wyszukiwarek możliwe jest z wykorzystaniem różnych technologii programistycznych oraz dostępnych narzędzi. W niniejszym artykule przedstawiamy wyszukiwarkę internetową **SearchSystem** opracowaną z wykorzystaniem systemu IBM Intelligent Miner for Text (IMT) [1]. Opracowana wyszukiwarka posiada unikalne zawansowane funkcje, które pozwalają zakwalifikować ją do tzw. wyszukiwarek „inteligentnych" [2].

2. System IBM Intelligent Miner for Text

System IBM Intelligent Miner for Text ver. 2.3.1 przeznaczony jest dla programistów i projektantów systemów, których celem jest m in. szeroko pojęta analiza, przetwarzanie oraz gromadzenie dokumentów tekstowych [2]. Z punktu widzenia użytkownika pakiet ten dostarcza gotowe programy, jak również środowisko bibliotek programistycznych służących do budowy własnych aplikacji.

System zawiera następujące aplikacje: *Text Analysis Tools*, *Text Search Engine* i *Web Crawler*. System zawiera także inne narzędzie, takie jak NetQuestion Solution i Java Sample GUI, które nie są używane w niniejszym projekcie. Grupa narzędzi analizy leksykalnej (*Text Analysis Tools*) składa się z narzędzi do rozpoznawania języków, do podsumowania, do kategoryzacji tematycznej, do grupowania oraz do ekstrakcji cech. *Narzędzie do rozpoznawania języków* otrzymując na wejściu dokument potrafi skutecznie określić język, w jakim on został napisany. Działa ono na zasadzie statystycznej analizy tekstu, porównując zawartość dokumentu ze zbiorem cech uzyskanych ze wzorcowego zbioru dokumentów każdego języka. Zbiór, dla którego podobieństwo to ma najwyższą wartość, określa język dokumentu. Aktualnie narzędzie obsługuje 13 języków, niestety, bez języka polskiego. Zawiera natomiast moduł treningowy umożliwiający dołożenie do standardowo zdefiniowanej grupy nowych języków. *Narzędzie podsumowujące* potrafi odnaleźć te zdania (wyrażenia) w dokumencie, które w najlepszy sposób oddają jego zawartość, przez co mogą być użyte do podsumowania dokumentu. Pozwala to użytkownikowi wstępnie ocenić przydatność odnalezionego dokumentu, bez konieczności czytania jego zawartości. Z punktu widzenia wyszukiwarki internetowej aplikacja ta umożliwia wzbogacenie zwracanej listy linków sieciowych o krótkie ich opisy. Narzędzie to najlepiej współpracuje z dokumentami

dobrze ustrukturyzowanymi. Użytkownik może zdefiniować maksymalną liczbę zdań zawartych w podsumowaniu jako wartość absolutną lub procentową, w stosunku do długości dokumentu. Mankamentem tego narzędzia jest fakt, że współpracuje ono jedynie z dokumentami w języku angielskim. Nasz system opracowuje streszczanie tylko dla dokumentów sklasyfikowanych jako angielskie.



Rys. 1. Środowiska modułu *Text Search Engine*
Fig. 1. Operating environments of *Text Search Engine* module

Text Search Engine jest kluczowym elementem całego systemu. Stanowi go zbiór bibliotek do wykorzystania z poziomu języka C oraz gotowych programów wykonywalnych przeznaczonych do uruchomienia z linii komend. Jest to aplikacja typu klient-serwer umożliwiająca wielu jednoczesnym klientom zadawanie zapytań lub wykonywanie różnych zadań administracyjnych. Klient i serwer mogą pracować na różnych platformach sprzętowych (rys. 1). Oprogramowanie to wykorzystywane jest do analizy dokumentów tekstowych w celu późniejszej ich indeksacji. Moduł ten potrafi tworzyć i przetwarzać cztery typy indeksów: (1) indeks lingwistyczny, (2) indeks precyzyjny, (3) indeks precyzyjny - znormalizowany, (4) indeks n-gram. W projekcie wykorzystano tylko indeks lingwistyczny. Wejściowy dokument w zależności od typu wybranego indeksu poddawany jest różnym analizom lingwistycznym. Wyjątek stanowi indeks typu n-gram, dla którego żadna analiza językowa nie jest stosowana. Analiza lingwistyczna dotyczy dwóch etapów przetwarzania: indeksacji oraz przetwarzania zapytań. Indeksacja jest procesem przetwarzania dokumentu, w którym jego treść jest w pewien sposób klasyfikowana i zapamiętywana. Po zakończeniu tego

procesu system wyszukujący jest w stanie określić odpowiedniość dokumentu w stosunku do sformułowanego zapytania. Podczas indeksacji dokumenty poddawane są następującym zabiegom w celu ekstrakcji grupy terminów, które są z tym dokumentem powiązane: tokenizacja, normalizacja, rozpoznawanie zdań, sprowadzanie terminów do podstawowej formy gramatycznej, usuwanie terminów popularnych oraz dekompozycja terminów złożonych. Wszystkie te zabiegi powodują, że stosunek objętość indeksów lingwistycznych w odniesieniu do objętości zaindeksowanych dokumentów jest z reguły najmniejszy. W zamian tego czas budowy indeksu oraz przetwarzanie zapytania kierowanego do tego indeksu jest odpowiednio dłuższy.

Web Crawler jest narzędziem, którego zadaniem jest utrzymywanie lokalnie aktualnej kopii wszystkich obiektów spełniających kryteria na wybranych witrynach internetowych.

3. Wyszukiwarka SearchSystem

System IBM Intelligent Miner for Text został wykorzystany do opracowania własnej wyszukiwarki internetowej o nazwie **SearchSystem.** Każdy system wyszukiwawczy oprócz własnych dodatkowych mechanizmów i udoskonaleń zawierać musi trzy podstawowe elementy: moduł ściągający i aktualizujący informację o dokumentach z sieci, maszynę indeksującą oraz mechanizm przeszukujący zindeksowane zasoby zgodnie z zadanymi kryteriami. W celu zrealizowania zadania należało stworzyć te trzy niezbędne moduły wykorzystując dostępne narzędzia. Oprogramowanie zostało zrealizowane z użyciem kompilatora Microsoft Visual C++ 6.0., API systemu IBM Intelligent Miner for Text oraz bazy danych IBM DB2 UDB.

W projekcie zadanie robota sieciowego (szperacza) realizuje zmodyfikowana aplikacja *Web Crawler*. Wykorzystuje ona moduł maszyny wyszukującej (*Text Search Engine*), dwa spośród pięciu narzędzi analizy leksykalnej (*Text Analysis Tools*) oraz system IBM DB2 obsługujący bazę danych systemu wyszukiwawczego. System uwzględnia występowanie obiektów zagnieżdżonych (grafika, dźwięk, multimedia, applety) w szkieletach dokumentów HTML-owych. Aplikacja obsługująca zapytania użytkowników jest uruchamiana na serwerze webowym Apache i wykorzystuje mechanizmy CGI do komunikacji z użytkownikiem. Program ten korzysta z modułu wyszukującego przetwarzającego zapytania oraz z bazy danych dostarczającej dodatkowe informacje wzbogacające treść zwróconej odpowiedzi. W aplikacji tej wykorzystano część dostępnych mechanizmów maszyny wyszukującej. Są nimi: ogólne przetwarzanie zapytań, możliwość wymuszenia występowania wprowadzonych terminów w treści dokumentu, paragrafu bądź zdania, wprowadzenie ekspansji zapytania poprzez synonimy lub analiza zapytań bazująca na wymowie zamiast pisowni. Niestety,

niektóre wprowadzone funkcje pracują tylko z dokumentami w języku angielskim i z naszego punktu widzenia jest to największa wada tego pakietu. Interfejs użytkownika zaprojektowanej aplikacji pozwala wybrać rodzaj wyświetlanych informacji w odpowiedzi, jak również sposób ich porządkowania. Dodatkową funkcją jest możliwość wyznaczenia czasu potrzebnego do załadowania strony w momencie przetwarzania zapytania. Należy jednak pamiętać, że wynik ten jest obarczony czasem odpowiedzi DNS-a oraz że dotyczy on serwera, na której działa wyszukiwarka, a nie stacji klienta, na której zadano zapytanie.

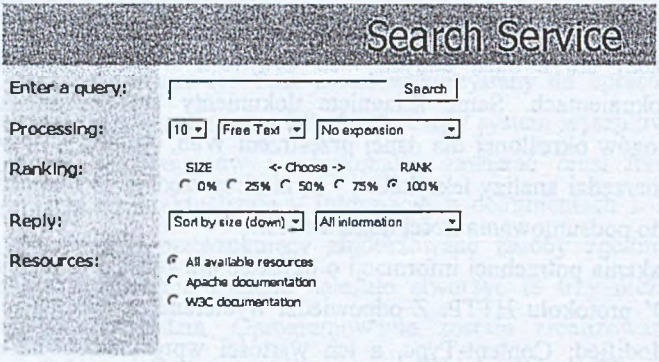
Baza danych stanowi, obok indeksów, podstawowy człon wyszukiwarki. Informacje katalogowane o dokumentach zawierają: język dokumentu, jego ewentualne podsumowanie, datę ostatniej modyfikacji, rozmiar oraz indeks, w którym został dokument umieszczony. Informacja przechowywana o obiektach zagnieżdżonych zawiera jedynie typ, rozmiar oraz datę ostatniej modyfikacji tych obiektów.

Zadaniem aplikacji indeksującej jest wprowadzenie wszystkich ściągniętych dokumentów przez szperacza sieciowego do bazy danych oraz indeksów systemu wyszukiwawczego. Program ten łączy się z bazą danych *Web Crawlera* w celu uzyskania informacji o ściągniętych dokumentach. Same ściągnięte dokumenty umieszczone są w specjalnej strukturze katalogów określonej dla danej przestrzeni Web. Aplikacja ta wykorzystuje dwa spośród pięciu narzędzi analizy leksykalnej - są nimi: narzędzie do rozpoznawania języków oraz narzędzie do podsumowania treści dokumentów.

W celu uzyskania potrzebnej informacji o obiekcie dla każdego z nich wywoływana jest metoda 'HEAD' protokołu HTTP. Z odpowiedzi wybierane są następujące pola: Content-Length, Last-Modified, Content-Type, a ich wartości wprowadzane są do bazy. Jeśli z pewnych przyczyn nie można otrzymać tych informacji (np. obiekt nie istnieje), pola ustawiane są na wartości odpowiednio (0,'unknown','unknown'). Wprowadzenie tych informacji jest niezbędne w celu uniknięcia wielokrotnego odwoływania się do nieistniejących obiektów.

Aplikacja wyszukująca jest programem działającym w środowisku serwera Web (Apache) i komunikuje się z użytkownikiem z wykorzystaniem interfejsu CGI. Aplikacja ta udostępnia użytkownikowi szereg opcji dotyczących specyfikacji zapytania, określenia zasobów, w których ma nastąpić wyszukiwanie, oraz sposobu porządkowania oraz wyświetlenia rezultatów. Program ten otrzymując zapytanie na wejściu przekazuje je do modułu zarządzającego indeksami. Moduł ten po przetworzeniu zapytania zwraca listę identyfikatorów dokumentów posortowaną zgodnie z wartością rankingową w kolejności malejącej. Lista tych identyfikatorów wzbogacona jest o szereg informacji (określonych przez użytkownika) z bazy danych systemu wyszukiwawczego. Tak przygotowany rezultat odpowiedzi przedstawiany jest użytkownikowi w postaci strony HTML.

Specyfikacja kwerendy polega na wpisaniu listy słów w polu tekstowym 'Enter a Query' oraz określeniu warunków przetwarzania. Dostępne są następujące opcje: (i) maksymalna liczba zwracanych rezultatów; (ii) określenie sposobu przetwarzania zapytania; (iii) zastosowanie mechanizmów ekspansji zapytania. Określenie sposobu przetwarzania zapytania polega na wybraniu jednej z czterech dostępnych opcji: (1) *Free Text* (najbardziej ogólne zapytanie - nie jest wymagane wystąpienie wszystkich występujących terminów kwerendy w dokumencie); (2) *Document* (wszystkie wpisane terminy muszą wystąpić w tekście dokumentu); (3) *Paragraph* (wszystkie wpisane terminy muszą wystąpić w jednym paragrafie); (4) *Sentence* (wszystkie wpisane terminy muszą wystąpić w jednym zdaniu). Opcje od 2 do 4 powodują potraktowanie zapytania jako Boolowskie, w którym wszystkie wprowadzone terminy połączone są operatorem AND. Różnica między nimi polega na innym ograniczeniu występowania terminów.

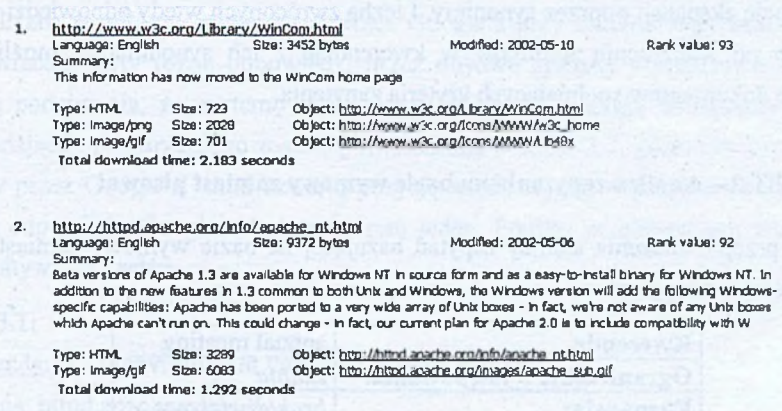


Rys. 2. Interfejs systemu **SearchSystem**
Fig. 2. The interface of **SearchSystem**

W systemie możliwe są dwa sposoby ekspansji zapytania: poprzez synonimy albo opierając się na analizie dźwiękowej. W pierwszym przypadku każdy termin kojarzony jest z listą synonimów z nim związanych. W drugim przypadku (ekspansji dźwiękowej), każdy termin kojarzony jest z grupą słów o identycznej wymowie. Wszystkie skojarzone terminy tworzą alternatywę Boolowską 'OR'. Tak rozbudowane zapytanie przekazywane jest do modułu wyszukującego. Efektem tego jest z reguły dużo obszerniejsza lista rezultatów. Aplikacja wyszukująca pozwala na określenie sposobu porządkowania rezultatów. Domyślnie ranking dokumentu decyduje o jego pozycji na liście. Możliwe jest jednak żądanie, aby rezultaty posortować zgodnie z całkowitym rozmiarem strony w kolejności rosnącej bądź malejącej. Oprócz tego wpływ na pozycję dokumentu na liście rezultatów mogą mieć ranking dokumentu oraz całkowity jego rozmiar. Wkład obu tych czynników określany jest dyskretnie z 25% wartością skokową. Ustawienie opcji "0%" powoduje, że porządkowanie zależy tylko

i wyłącznie od rozmiaru. W przypadku "100%" ranking decyduje całkowicie o pozycji dokumentu na liście. Inne wartości wymuszają wpływ częściowy.

Określenie rodzaju zwracanych informacji polega na wybraniu jednej z czterech następujących opcji: (i) *all information* (wyświetla: URL, rozmiar, język, podsumowanie, ranking, datę modyfikacji oraz informację o obiektach zagnieżdżonych); (ii) *documents' details* (wyświetla informacje z poprzedniej opcji z wyjątkiem listy obiektów zagnieżdżonych); (iii) *embedded objects* (wyświetla informacje o obiektach zagnieżdżonych); (iv) *download time* (wyświetla wszystkie informacje z pierwszej opcji oraz dodatkowo czas potrzebny do ściągnięcia strony z obiektami zagnieżdżonymi).


Results list:

1. http://www.w3c.org/Library/WinCom.html
Language: English Size: 3452 bytes Modified: 2002-05-10 Rank value: 93
Summary:
This information has now moved to the WinCom home page

Type: HTML Size: 723 Object: http://www.w3c.org/Library/WinCom.html
Type: image/png Size: 2028 Object: http://www.w3c.org/Icons/WWW/w3c_home
Type: image/gif Size: 701 Object: http://www.w3c.org/Icons/WWW/1b48x
Total download time: 2.183 seconds

2. http://httpd.apache.org/info/apache_nt.html
Language: English Size: 9372 bytes Modified: 2002-05-06 Rank value: 92
Summary:
Beta versions of Apache 1.3 are available for Windows NT in source form and as a easy-to-install binary for Windows NT. In addition to the new features in 1.3 common to both Unix and Windows, the Windows version will add the following Windows-specific capabilities: Apache has been ported to a very wide array of Unix boxes - in fact, we're not aware of any Unix boxes which Apache can't run on. This could change - in fact, our current plan for Apache 2.0 is to include compatibility with W

Type: HTML Size: 3289 Object: http://httpd.apache.org/info/apache_nt.html
Type: image/gif Size: 6083 Object: http://httpd.apache.org/images/apache_sub.gif
Total download time: 1.292 seconds

Rys. 3. Przykładowa odpowiedź
Fig. 3. Sample results

4. Testy systemu wyszukiwawczego

Przeprowadzono testy samego systemu **SearchSystem**, jak również testy porównawcze z innym systemem wyszukiwawczym. Testy samego systemu miały za zadanie pokazać oferowane zaawansowane możliwości analizy lingwistycznej. Poniżej prezentujemy dwa reprezentatywne elementy tej analizy, a mianowicie: ekspansję zapytania poprzez synonimy oraz analizę zapytania na bazie wymowy zamiast pisowni. Testy porównawcze polegały na zadaniu takich samych zapytań najpopularniejszej obecnie wyszukiwarce Google oraz naszemu systemowi.

4.1. TEST 1 – Ekspansja zapytania poprzez synonimy

Test przeprowadzono dla następującego zapytania:

Kwerenda	error remark
Ograniczenie występowania:	*zdanie*
Ekspansja:	*brak/synonimy*[1]
Zasoby:	*wszystkie zasoby*

Kwerenda "error remark" mogłaby być zadana przez osobę nie znającą dobrze języka angielskiego. W języku angielskim takiego zwrotu raczej się nie spotyka. Wynik takiego zapytania jest zbiorem pustym, czego świadectwem jest odpowiedni komunikat: "Message: No documents meet the query criteria or improper query". W celu wspomożenia kwerendy włączono opcję ekspansji poprzez synonimy. Liczba zwróconych wtedy odpowiedzi wyniosła 50. Dopiero po skojarzeniu terminów w kwerendzie z ich synonimami, możliwe było odnalezienie dokumentów spełniających kryteria zapytania.

4.2. TEST 2 – Analiza zapytania na bazie wymowy zamiast pisowni

W celu przeprowadzenia analizy zapytań bazującej na bazie wymowy zamiast pisowni zadano następujące zapytanie:

Kwerenda	annual meating
Ograniczenie występowania:	*zdanie*
Ekspansja:	*brak/dźwiękowe*
Zasoby:	*wszystkie*

W zadanej kwerendzie specjalnie zamieniono literkę w drugim słowie (zamiast meating powinnno być meeting). Nie zmienia jednak to wymowy terminu. Po zadaniu zapytania odpowiedź, podobnie jak w poprzednim przypadku, była zbiorem pustym. Następnym krokiem było włączenie analizy dźwiękowej oraz powtórzenie testu. Wynik otrzymany zawierał listę kilku odpowiedzi, które zostały przedstawione poniżej.

http://www.w3c.org/WAI/2000/03/agenda

http://www.w3c.org/WAI/IPO/Activity.html

http://www.w3c.org/People/howcome/p/pirater/

http://www.w3c.org/WAI/IPO/Activity

Pierwszy zwrócony rezultat został przeszukany pod kątem występowania terminu annual. Termin wystąpił w nim tylko jeden raz w następującym zdaniu: "Over the course of the week of March 20th, four WAI meetings will occur, before and after the annual CSUN

[1] Pierwszy test robiony był przy wyłączonej jakiejkolwiek ekspansji zapytania. W drugim teście włączono ekspansję synonimów.

conference". W zdaniu tym odnaleziono również termin meeting, co jest dowodem dźwiękowego skojarzenia terminów *meeting* i *meating* jako identycznych.

4.3. TEST 3 – Testy porównawcze

W badaniach porównawczych porównywano system **SearchSystem** z wyszukiwarką Gogle dla czterech różnych zapytań. Ponieważ nasz system wyszukiwawczy miał poindeksowane strony pochodzące tylko z dwóch witryn, dlatego też w testach Google'a zawężono możliwe zwracane rezultaty tylko do tych wybranych witryn (httpd.apache.org, www.w3.org). Pomimo tego liczba zaindeksowanych przez Google'a stron z tych dwóch witryn była kilka razy większa niż w przypadku projektowanego systemu. Efektem tego była większa liczbą rezultatów zwracanych przez Google'a przy każdym zapytaniu. Co więcej, często zwracane były różne odpowiedzi przez obydwa systemy wyszukiwawcze. Było to podstawą podejrzenia, że systemy te w różny sposób wyznaczają dokumenty relewantne (odpowiadające). Podejrzenie to zostało potwierdzone w teście 3.1, gdzie pierwszy dokument zwrócony przez Google'a miał czwartą pozycję wśród wyników projektowanego systemu, zaś piąta odpowiedź Google'a była numerem jeden. Poniżej przedstawiono wyniki dwóch reprezentatywnych testów.

Test 3.1:

kwerenda: „CGI environment variable usage"

domena: httpd.apache.org

ograniczenie występowania: dokument

URL	SearchSystem		Google
	Odpowiedź	Występowanie	Odpowiedź
docs/upgrading_to_1_3.html	4	+	1
/docs-2.0/glossary.html	-	-	2
/docs/new_features_1_3.html	2	+	3
/docs/misc/rewriteguide.html	-	+	4
/docs-2.0/mod/mod_rewrite.html	1	+	5
/docs-2.0/mod/mod_ssl.html	-	+	6
/docs/mod/mod_rewrite.html	-	+	7
/docs-2.0/mod/core.html	-	+	8
/mail/cvs/200208	-	-	9

Liczba rezultatów (Google): 9; Liczba rezultatów (SearchSystem): 8

Test 3.2:

kwerenda: "XML specification"

domena: www.w3.org

ograniczenie występowania: zdanie

URL	SearchSystem		Google
	Odpowiedź	Występowanie	Odpowiedź
/TR/REC-xml	-	+	1
/XML/	-	+	2
/TR/REC-xml-names/	-	-	3
/TR/xhtml1/	-	+	4
/TR/WD-xml-lang-970331.html	-	-	5
/TR/2001/RE-xml-c1	-	-	6
/Press/1998/XML10-REC-fact	1	+	-
/QA/TheMatrix.xml	2	+	-
/Submission/200/07	3	+	-
/Consortium.Translation/Japanese	4	+	-

Liczba rezultatów (Google): >5000; Liczba rezultatów (Search System): 50

Test 3.2 pokazuje przede wszystkim przewagę Google'a w liczbie zaindeksowanych stron. W przypadku projektowanego systemu liczba ta wynosiła ok. 1200-1300 dokumentów, bowiem taką liczbę dokumentów zaindeksowano na etapie gromadzenia informacji. Google dla tego samego zapytania „XML specification" znalazł ponad 5000 rezultatów, co oznacza, że dysponuje on co najmniej kilka razy większą bazą dokumentów. Test ten dodatkowo potwierdza tezę o różnym sposobie rankingowania dokumentów przez oba systemy. Projektowana wyszukiwarka zaindeksowała dokument pierwszy zwrócony przez Google'a, znalazła jednak 50 innych dokumentów bardziej odpowiednich dla przykładowego pytania. Przy przewadze Google'a dotyczącej liczby zaindeksowanych stron jest niemal pewne, że posiada on większość z tych 50 zwróconych dokumentów przez opracowaną wyszukiwarkę.

5. Uwagi końcowe

System IBM Intelligent Miner for Text to zaawansowane narzędzie do budowy aplikacji zorientowanych na przetwarzanie tekstów. Zawiera on wszystkie niezbędne elementy do budowy zaawansowanego systemu wyszukiwawczego w sieci Internet oraz sieciach lokalnych. Liczba oraz możliwości narzędzi analizy leksykalnej, które są oferowane przez system, są imponujące. Poza kilkoma przykładowymi aplikacjami realizującymi podstawowe funkcje, dopiero środowisko programistyczne otwiera prawdziwe możliwości systemu. W

niniejszym projekcie opracowano wyszukiwarkę internetową **SearchSystem**, z wykorzystaniem wybranych funkcji systemu IMT. Wśród oferowanych metod analizy leksykalnej zastosowano dwie z nich, których nie posiada wyszukiwarka Google, a mianowicie ekspansję poprzez synonimy oraz analizę dźwiękową zapytań. Wyniki testów porównawczych z systemem Google wypadły na korzyść tego drugiego, przede wszystkim z powodu liczby dokumentów, jakie ten system indeksuje. Dlatego analizując wyniki trzeba mieć ten fakt na uwadze. Analizując możliwości zaprojektowanej wyszukiwarki **SearchSystem** możemy stwierdzić, że nadaje się ona przede wszystkim tam, gdzie wymagana jest dogłębniejsza analiza przetwarzanych danych (np. biblioteki elektroniczne, witryny konkurencyjnych firm).

LITERATURA

1. Intelligent Miner for Text ver. 2.3.1. Dokumenty: SH12-6370, 6365, 6362, 6371.
2. Kłopotek M.: Inteligentne wyszukiwarki internetowe. Wyd. EXIT, Warszawa 2001.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 9 kwietnia 2003 r.

Abstract

The aim of the project was to find out how the IBM Intelligent Miner for Text can be used to develop a "Google"-like search machine. It is a powerful software, which is a perfect tool to create advanced search systems. It provides a collection of analysis tools and advanced linguistic processing methods (some of them were used in the project), that are missing in Google machine. In the project some of them have been used and tested. The system called **SearchSystem** has been developed. The miner supports three means of query expansion mechanisms working on linguistic indexes. In our system two of them have been included: sound and synonym expansions. Apart free text queries the engine allows very sophisticated attribute and Boolean queries. A comparison test with the Google search engine has been done. It is widely agreed, that Google is the most commonly used search system in the Internet. The reason is its working speed and most of all the number of web pages indexed.

This was the main reason why it gave more complete results, than the **SearchSystem**. The tests showed that probably both tools use completely different ranking methods.

Adresy

Leszek BORZEMSKI: Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów, ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Polska, leszek@ists.pwr.wroc.pl.

Piotr ŁOPATKA: Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów, ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Polska, piotr_lopatka@yahoo.com.