NAUKI PODSTAWOWE
CZASOPISMO TECHNICZNE
TECHNICAL TRANSACTIONS
FUNDAMENTAL SCIENCES
WYDAWNICTWO POLITECHNIKI KRAKOWSKIEJ
1-NP/2012
ZESZYT 18
ROK 109
ISSUE 18
YEAR 109

KRZYSZTOF ULMAN, KRZYSZTOF RZECKI*

ALGORYTM WYKRYWANIA TREŚCI NA STRONACH PORTALI INTERNETOWYCH

DETECTION ALGORITHM FOR CONTENT ON INTERNET WEB PORTALS

Streszczenie

W artykule przedstawiono podejście wykorzystane podczas projektowania i implementowania algorytmu automatycznego wykrywania treści na stronach portali internetowych oparte o analizę struktury kodu HTML strony WWW. Za treść strony uznano tekst artykułów wraz z jego nagłówkiem, z pominięciem innych tekstów występujących na stronie (menu, reklamy, komentarze, podpisy pod zdjęciami, itp.).

Słowa kluczowe: wykrywanie treści, eksploracja danych, ekstrakcja danych, gromadzenie danych, analiza budowy stron WWW, HTML

Abstract

The paper shows steps, made during designing and implementing automatic web pages contents recognition algorithm, based on HTML structure analysis. A web page contents is the article text with its headline, without any other text like menu, advertisements, user's comments, image captions, etc.

Keywords: web pages contents recognition, data mining, web scraping, data collection, web pages structure analysis, HTML

* Inż. Krzysztof Ulman, dr inż. Krzysztof Rzecki, Instytut Teleinformatyki, Wydział Fizyki, Matematyki i Informatyki, Politechnika Krakowska.

1. Wstęp

Każda nowoczesna strona WWW oprócz głównej treści (np. tekstu publikacji dziennikarskiej, czy naukowej) składa się z wielu innych elementów, takich jak: nagłówek strony, stopka, menu, hiperłącza do innych podstron, elementy multimedialne, podpisy pod zdjęciami, reklamy, komentarze, ankiety itp. W przypadku portali internetowych (zwłaszcza informacyjnych) strony WWW są generowane dynamicznie przez oprogramowanie do zarządzania treścią CMS (*Content Management System*). Źródłowy zredagowany artykuł jest czystym tekstem (z ewentualnymi referencjami) umieszczonym w bazie artykułów. Przygotowanie danej strony przez CMS polega na zastosowaniu szablonu, który jest wypełniany elementami pobranymi z różnych baz danych danego portalu. Szablon, który sam może pochodzić z bazy szablonów, zapewnia zestandaryzowaną dla danego portalu (lub działu portalu) postać strony, a bazy danych dostarczają zmienne elementy (w tym tekst artykułu, reklamy, konfigurację formatowania, itp.). Odczytanie istoty pierwotnej informacji (czyli tekstu artykułu) z tak skonstruowanych witryn często staje się trudne bądź niewygodne dla użytkownika. W przypadku, gdy chcemy taką stronę zapisać, gromadzimy na dysku twardym kilkukrotnie więcej danych niż zajmuje interesująca nas treść. Ponadto, jeśli chcemy pobierać dane ze stron WWW i gromadzić do korpusu tekstów, by później je analizować (np. badania związane z przetwarzaniem języka naturalnego, wymagające dużej ilości próbek tekstowych), może się okazać, że nadmiar dodatkowych informacji nam to uniemożliwi lub wypaczy wyniki. Jak temu zaradzić, kiedy nie mamy bezpośredniego dostępu do bazy artykułów danego dostawcy treści? Z pomocą może przyjść algorytm wykrywania treści na stronach portali internetowych.

Za właściwą treść strony WWW uznajemy tekst artykułu wraz z jego nagłówkiem. Projektowany algorytm wykrywania treści w założeniu ma automatycznie wykryć treść artykułu z pominięciem innych znajdujących się na stronie napisów, takich jak: reklamy, komentarze, podpisy pod obrazkami itp. Chcielibyśmy, aby projektowany algorytm był uniwersalny, czyli samoczynnie dopasowywał się do wskazanego portalu i był odporny na zmiany struktury przez autorów czy administratorów odpowiedzialnych za formatowanie stron mających zawierać artykuły.

Docelowo zaprojektowany algorytm został wykorzystany w aplikacji tworzącej korpusy językowe na bazie artykułów umieszczanych na stronach portali internetowych, która cały czas periodycznie pobiera artykuły z portali.

2. Aktualnie stosowane algorytmy i systemy wykrywania treści

Natrafiono na kilka komercyjnych przykładów zastosowania algorytmu wykrywania treści. Jednym z nich jest moduł Reader w przeglądarce Apple® Safari od wersji 5 (opublikowanej w czerwcu 2010 roku). Reader pozwala wyświetlić treść artykułu w wygodnej do czytania formie: na białym tle, bez banerów reklamowych (często animowanych, co dodatkowo rozprasza czytelnika), bez zbędnego tekstu menu, komentarzy, itp. W niedługim czasie pojawiła się wtyczka (*plugin*) iReader o podobnej funkcjonalności dla przeglądarek Mozilla Firefox oraz Google Chrome. Kod źródłowy Safari Reader jest zamknięty, a iReader obfuskowany (celowo zaciemniony, w taki sposób, aby jego

zrozumienie przez człowieka było znacznie utrudnione), więc nie wiadomo do końca, na jakiej zasadzie pracują. Jednak po analizie działania można wnioskować, że użyte w nich algorytmy bazują na podobnych założeniach co prezentowany algorytm. Takie wnioski można wysunąć na podstawie zbliżonych problemów z rozpoznawaniem treści, na tych samych stronach.

3. Algorytm wykrywania treści

Tradycyjne podejście do wyszukania treści pozwala pracować tylko z jednym, wybranym portalem (lub jego częścią), pod warunkiem dokonania wcześniejszej analizy budowy i wyróżnieniu poszczególnych części witryny [4]. Przykładowo dla portalu RMF24: nagłówek to znacznik *<h1>* wewnątrz bloku * *, pierwszy akapit znajduje się w znaczniku *<p>* wewnątrz bloku *<div class=”lead textContent”>*, a następne akapity również w znaczniku *<p>*, jednak wewnątrz bloku *<div class=”text textContent”>*. Metoda ta jest szczególnie wrażliwa na zmianę struktury strony przez autora i posiada wszystkie wady, od których miał być wolny projektowany algorytm, a które zostały wymienione we wstępie.

Podejście zastosowane podczas projektowania omawianego algorytmu opiera się na analizie struktury kodu HTML, ale z pominięciem szczegółowych znaczników i atrybutów. Dzięki sprawdzaniu stosunkowo niewielkiej ilości elementów języka HTML oraz wyszukiwaniu jak najbardziej ogólnych warunków brzegowych udało się uniezależnić rozpoznawanie treści od konkretnego portalu [1]. Nie jest natomiast analizowany kod JavaScript ani style CSS (*Cascading Style Sheets* – kaskadowe arkusze styli). Algorytm otrzymał roboczą nazwę ***PortalCrawler***.

Implementacja prototypowa została wykonana w języku Perl. Do analizowania stron wykorzystano pakiety HTML::TreeBuilder oraz HTML::Element, które na podstawie kodu HTML tworzą drzewo DOM (ang. *Document Object Model* - obiektowy model dokumentu) i pozwalają je wygodnie przeszukiwać.

3.1. Metoda rozróżniania typów podstron

Kluczowym elementem algorytmu jest kryterium rozróżniania typów podstron, czyli stron zawierających artykuły, od tych, które ich nie zawierają. Na podstawie wcześniejszej analizy budowy stron WWW oraz standardu języka HTML [2] zdecydowano, że takim kryterium będzie znalezienie nagłówka i jest to warunek konieczny, ale niewystarczający do stwierdzenia, że na stronie znajduje się artykuł.

Podczas analizy wybranych portali zauważono, że nagłówek zwykle zawarty jest w tagu *<h1>* lub *<h2>*, a ponadto występuje również w meta znaczniku *<title>*. Z powyższą sytuacją mamy do czynienia w popularnych systemach CMS, używanych także przez duże firmy i redakcje dzienników na całym świecie. Sprawdzono: Drupal, Joomla!, Mambo, MediaWiki, PHP-Fusion, PHP-Nuke, TYPO3, WordPress, XOOPS. Warto zaznaczyć, że odnaleziona prawidłowość nie ma nic wspólnego z samym standardem języka HTML, a jest jedynie przykładem na spójne jego wykorzystanie na wielu badanych portalach.

Zdarza się, że na stronie występuje znacznik *<h1>*, który nie zawiera nagłówka artykułu, a jego zawartość występuje w treści meta znacznika *<title>* – dotyczy to najczęściej samej nazwy portalu. Stąd też wprowadzono minimalną długość nagłówka (domyślnie 18 znaków). Dodatkowo, jeśli znaleziono więcej potencjalnych nagłówków, jako właściwy uznawany jest ten, który występuje wcześniej wewnątrz meta znacznika *<title>*, gdyż taką prawidłowość zaobserwowano podczas analizowania struktury stron wybranych portali.

Jeżeli na stronie nie znaleziono nagłówka, to można poszukać odnośników do innych stron, które mogą zawierać artykuły. Zaobserwowano, że takie odnośniki znajdują się zazwyczaj wewnątrz list zwykłych (znacznik *<ul>*).

3.2. Wykrywanie treści artykułu

Jak wcześniej wspomniano, warunkiem koniecznym do uznania, że strona zawiera artykuł jest znalezienie nagłówka, natomiast warunkiem wystarczającym jest jednoczesne znalezienie bloku tekstu o długości większej niż pewna ustalona liczba znaków (domyślnie 280) poniżej nagłówka. Nagłówek okazał się znakomitym elementem brzegowym dla początku szukanej treści artykułu, natomiast znalezienie drugiego uniwersalnego elementu brzegowego, kończącego tekst, okazało się niezwykle trudne.

Po odnalezieniu nagłówka kolejnym krokiem jest obróbka wszystkich bloków * * oraz **. Modyfikacja polega na usunięciu całego bloku lub bloków w nim zagnieżdżonych przez usunięcie fragmentu kodu HTML od początku znacznika otwierającego do końca odpowiadającego mu znacznika zamykającego, a więc łącznie z zawartym w nim tekstem oraz podrzędnymi znacznikami.

Z bloków usuwane są wszystkie zagnieżdżone znaczniki, z wyjątkiem tych odpowiedzialnych za formatowanie tekstu (*<p>, , <blockquote>, <a>, <b>, <i>, <u>, , <small>, <h2> ... <h7>)* oraz innych wykorzystywanych do późniejszej analizy (*
, <img>*). Należy tutaj zwrócić uwagę na podwójną rolę znacznika **, który jest wykorzystywany w dwojaki sposób, tj. jako element grupujący oraz formatujący. Zdarza się, że rodzi to pewne problemy podczas rozpoznawania, tzn. jego pominięcie może skutkować wycięciem fragmentu sformatowanego tekstu, a pozostawienie może skutkować pojawieniem się np. podpisów spod obrazków w wynikowym tekście. W każdym bloku z początku i końca zostają usunięte fragmenty niebędące zdaniami, gdzie jako zdanie uważany jest tekst, który zaczyna się od dużej litery i kończy kropką, wykrzyknikiem lub pytajnikiem. W tak przetworzonych blokach obliczana jest długość pozostałego tekstu, przy czym należy uwzględnić tekst leżący bezpośrednio w bloku typu * * oraz tekst leżący wewnątrz ewentualnie występujących znaczników *<p>*. Bloki, w których długość ta jest mniejsza niż ustalono (domyślnie 4 znaki), są pomijane.

Następnie wybierany jest blok z tekstem o największej długości – jest to główna treść artykułu. Aby wykluczyć, że jest to strona z samymi streszczeniami artykułów lub komentarzami, konieczne jest sprawdzenie, czy nie ma więcej niż ustalono (domyślnie 6) bloków z tekstem o podobnej długości (dłuższym niż połowa długości obecnego bloku). Do odszukania pozostaje pierwszy akapit tekstu, który leży zawsze pomiędzy znalezionym nagłówkiem a główną treścią, czasami w więcej niż jednym bloku, a jego minimalna długość wynosi domyślnie 90 znaków.

Dzięki usuwaniu z bloków wszystkich wymienionych wcześniej znaczników podrzędnych, z tekstu artykułu pomijane są wszelkie niechciane elementy (np. reklamy, bloki „Zobacz także", podpisy pod obrazkami, itd.). Dodatkowo kasowane są bloki <*span*> zawierające co najmniej jeden znacznik <*img*> oraz krótki tekst (domyślnie mniej niż 90 znaków). Dzięki temu pozbyć się można wszystkich podpisów pod zdjęciami. Elementów <*span*> nie można usunąć analogicznie do pozostałych znaczników, gdyż pełnią one również rolę formatującą tekst i takie działanie prowadziłoby do usunięcia części treści (często nawet pojedynczych wyrazów).

3.3. Wady algorytmu PortalCrawler

Zaprojektowany algorytm nie jest wolny od wad. W trakcie realizacji badań wykryto następujące aspekty wymagające dopracowania:

1. Jeżeli pierwszy paragraf tekstu nie jest umieszczony w bloku razem z pozostałą treścią, ani nie leży wewnątrz jednego ze znaczników grupujących (<*div*>, <*span*>, <*p*>, <*h2 ... h7*>) to nie zostanie on wykryty;
2. Jeżeli nie wykryto bloku komentarzy (element zawierający wykrywane słowo „Komentarze" lub „Opinie" nie istnieje, albo jest ładowany za pomocą JavaScript – przykład dla portalu Wirtualna Polska) i jednocześnie treść jednego z komentarzy jest dłuższa od tekstu artykułu, to komentarz ten zostaje rozpoznany, jako treść na stronie;
3. W przypadku gdy pierwszy wydzielony akapit tekstu jest dłuższy niż pozostała część artykułu, ta pozostała część zostaje pominięta;
4. W przypadku gdy pierwszy wydzielony akapit tekstu jest krótszy niż ustalono, (domyślnie 90 znaków) to zostanie on pominięty;
5. Jeżeli fragment tekstu wyświetlany jest w liście zwykłej (znacznik <*ul*>), to zostanie on pominięty;
6. Domyślnie nie są wspierane strony, których budowa oparta jest o tabele, a których treść nie jest dodatkowo ujęta w bloki grupujące <*div*> lub <*span*>;
7. Jeżeli na danej stronie nagłówek artykułu nie jest umieszczony wewnątrz znacznika <*h1*> bądź <*h2*>, to automatyczne rozpoznanie treści staje się niemożliwe. Z tego względu algorytm wyposażono w możliwość jawnego wskazania bloku zawierającego nagłówek, jednak przeczy to założonej uniwersalności;
8. Strona z fragmentami innych artykułów zostaje rozpoznana jako artykuł, jeżeli występuje duża dysproporcja pomiędzy długością fragmentów.
9. Podpisy pod obrazkami wchodzą w skład tekstu, jeżeli nie są wydzielone do podrzędnego bloku <*div*> lub są w bloku <*span*>, a ich długość jest większa niż ustalono (domyślnie 90 znaków);
10. W przypadku artykułów, których treść występuje więcej niż w dwóch wydzielonych blokach grupujących, zostanie rozpoznana tylko część treści (dwa lub trzy najdłuższe fragmenty);
11. W przypadku ładowania tekstu z następnych stron za pomocą JavaScriptu (z sytuacją taką można się spotkać w niektórych działach portalu Gazeta.pl) zostanie rozpoznana tylko część z pierwszej strony.

Zanim prezentowany algorytm uzyskał obecny kształt, próbowano innych podejść do wykrycia treści, w tym np. odszukanie najmniejszego bloku zawierającego nagłówek i duży fragment tekstu. Podejścia te od początku wykazywały małą dokładność i tendencję do zbierania nadmiarowej ilości danych ze strony.

Wymienione aspekty mają swoje odzwierciedlenie w skuteczności algorytmu, która zostanie omówiona w części dotyczącej oceny poprawności działania. Każdy z wymienionych punktów można jeszcze dopracować, przy czym należy zwrócić uwagę na fakt, że większość z nich może być o tyle trudna do dopracowania, że ich wdrożenie zachwieje stabilnością działania algorytmu. Może nawet skutkować powstaniem nowych sytuacji, w których wykrywanie będzie niepoprawnie wykonywane. W przyszłości wyniki działania tego algorytmu można poprawić np. przez dodatkową analizę kodu stylów CSS albo wykorzystanie elementów sztucznej inteligencji, które na podstawie podanych przykładów poprawnego i błędnego rozpoznania treści na danym portalu potrafiłyby radzić sobie lepiej z wykrywaniem interesujących nas elementów na stronach internetowych.

4. Ocena poprawności działania

Jedynym możliwym sposobem na ocenę poprawności rozpoznania treści artykułów jest porównanie przez człowieka treści całej oryginalnej strony internetowej z tekstem rozpoznanym. Automatyczna analiza semantyki tekstów byłaby bardzo trudna, jeśli w ogóle możliwa do zrealizowania.

4.1. Kryterium oceny i metodologia badania

Ustalono, że zebrane zostaną artykuły z polskojęzycznych portali informacyjnych o niejednorodnej tematyce, dla których przeprowadzana była analiza budowy oraz z co najmniej takiej samej liczby innych portali, niebędących w zestawieniu. Artykuły powinny być zbierane i rozpoznawane na przestrzeni kilku dni, aby wykluczyć wrażliwość na drobne zmiany struktury strony bądź tymczasowo prezentowane elementy (np. blok z życzeniami świątecznymi). Następnie zostanie sprawdzona poprawność rozpoznania treści minimum 100 artykułów wybranych losowo dla każdego portalu biorącego udział w badaniu.

Przyjęto, że treść artykułu została poprawnie rozpoznana, gdy jednocześnie:

1. Poprawnie rozpoznano nagłówek artykułu;
2. Poprawnie rozpoznano pierwszy akapit artykułu (zazwyczaj prezentowany pogrubioną czcionką);
3. Poprawnie rozpoznano pozostałą treść artykułu;
4. Pominięto informacje niebędące bezpośrednio częścią artykułu (data publikacji, autor, podpisy pod zdjęciami, bloki „Zobacz również” itp.);
5. Pominięto treść będącą częścią strony, a nie samego artykułu (tekst menu, reklamy, komentarze itp.);
6. Wynikowy tekst tworzy spójną, logiczną całość (zakładając, że tekst źródłowy również taki był).

Nie wszystkie przypadki sprawdzania poprawności są jednak proste do rozstrzygnięcia i zdarza się, że kilka osób może mieć odmienne zdanie na temat tego, czy dany fragment strony WWW zaliczyć do artykułu, czy też nie. Najwięcej wątpliwości budzą nagłówki umieszczone nad poszczególnymi partiami akapitów. W przypadku niejednoznaczności w ocenie kierowano się, poza wiernością oddania tekstu oryginalnego, możliwym niekorzystnym wpływem danego fragmentu na tworzony korpus językowy.

Wskaźnikiem, który będzie badany jest dokładność, określony analogicznie do definicji używanej w przypadku oceniania poprawności wyników wyszukiwarek internetowych [3]:

$$dokładność = \frac{liczba\ artykułów\ rozpoznanych\ poprawnie}{liczba\ wszystkich\ artykułów} \times 100\%$$

Wskaźnik ten zostanie obliczony dla reprezentatywnego zbioru artykułów wybranych do oceny poprawności działania algorytmu.

4.2. Wyniki badania

Poprawność zbadano dla 14 portali o różnej tematyce (Tabela 1). Okazało się, że algorytm dla portali, dla których przeprowadzono wstępną analizę, niekoniecznie uzyskał najlepsze wyniki. Poprawność wykrywania treści zależy głównie od:

1. W jakich i w ilu blokach umieszony jest szukany tekst?
2. Czy i w jaki sposób pomiędzy tekstem artykułu umieszczane są dodatkowe bloki?
3. Jaka jest długości tekstu artykułu (krótkie notatki przysparzają więcej problemów)?

Finalnie średnia dokładność algorytmu wykrywającego treść na portalach internetowych dla zbadanej próbki artykułów (sprawdzono łącznie 2098 stron z 14 portali) wyniosła **91,3%**, a odchylenie standardowe **8,4%**. Uzyskany wynik uznaje się za zadawalający, zwłaszcza że można wskazać sześć portali, dla których dokładność jest większa niż 95% oraz jeden, dla którego osiągnięto rezultat 100%.

Dokonano również obserwacji, z której wynika, że znaczna większość błędów rozpoznawania powiela się w obrębie danego portalu. Stąd też bardzo słaby wynik działania algorytmu dla portalu Interia, gdzie głównym problemem były działy, w których podpisów pod zdjęciami nie wydzielono osobnym blokiem, a na końcu artykułów przez pewien okres czasu znajdował się tekstowy materiał reklamowy, również niewydzielony do osobnego bloku.

Rys. 1. Dokładność wykrywania treści dla poszczególnych portali (opracowanie własne)

Fig. 1. Accuracy of content detection for individual portals (own)

Tabela 1

Wyniki sprawdzania poprawności rozpoznania treści

Nazwaportalu	Liczba zebranych artykułów	Liczba oznaczonych artykułów	Liczba oznaczonych, jako poprawne	Liczba oznaczonych, jako błędne[1]	Dokładność
Dziennik Internautów www.di.com.pl	119	119	108	0/0/4/7	90,8 %
F1 Portal www.f1portal.pl	152	152	152	0/0/0/0	100 %
Gazeta * www.gazeta.pl	207	131	125	4/0/2/0	95,4 %
Interia * www.interia.pl	129	129	88	0/20/6/15	68,2 %
International Data Group www.idg.pl	108	108	104	0/0/0/4	96,3 %
Money * www.money.pl	269	205	168	0/35/1/1	82,0 %
Nasze miasto Kraków www.krakow.naszemiasto.pl	123	123	105	2/2/6/8	85,4 %
Onet * www.onet.pl	192	165	149	6/3/4/3	90,3 %
RMF 24 * www.rmf24.pl	856	227	224	2/0/1/0	98,7 %
Ściąga www.sciaga.pl	231	172	167	0/0/4/1	97,1 %
Sport www.sport.pl	205	130	124	0/2/4/0	95,4 %
TVN 24 www.tvn24.pl	269	138	128	1/8/1/0	92,8 %
Wirtualna Polska * www.wp.pl	179	114	102	4/0/6/2	89,5 %
Wyborcza Biznes www.wyborcza.biz	189	185	178	1/1/5/0	96,2 %
Łącznie	3228	2098	1922	20/71/44/41	Średnia 91,3 %

***** Portale, na których dokonano analizy budowy.

Opracowanie własne autora

[1] Z podziałem na kategorie: inne/brak pierwszego akapitu/brak części tekstu/reklamy lub podpisy

Podczas sprawdzania i oznaczania artykułów przy pomocy przygotowanej do tego celu aplikacji osoba sprawdzająca musiała wskazać jedną z kategorii błędów, jeżeli treść została rozpoznana nieprawidłowo. W przypadku wystąpienia większej liczby nieprawidłowości wybierano przypadek poważniejszy, który w większym stopniu dyskwalifikował pobraną treść:

1. Brak pierwszego akapitu – pierwsza część tekstu prezentowana pogrubioną czcionką zawarta jest najczęściej w innym bloku niż reszta tekstu, stąd możliwe błędy podczas rozpoznawania;
2. Brak części tekstu – nie wykryto treści w ogóle, treść jest niekompletna, obcięty jest fragment tekstu itp.;
3. Reklamy lub podpisy – w zapisanej treści znalazły się reklamy, fragmenty bloków typu „Zobacz także", podpisy zdjęć itp.;
4. Inne – pozostałe błędy, w tym przypadki uznania za artykuł strony, która go nie zawierała.

Najczęstszym błędem było nierozpoznanie pierwszego akapitu tekstu, kolejnym brak części artykułu. Znaleziono również jeden przypadek nieprawidłowego wykrycia nagłówka, ale rozpoznany tekst był jego fragmentem.

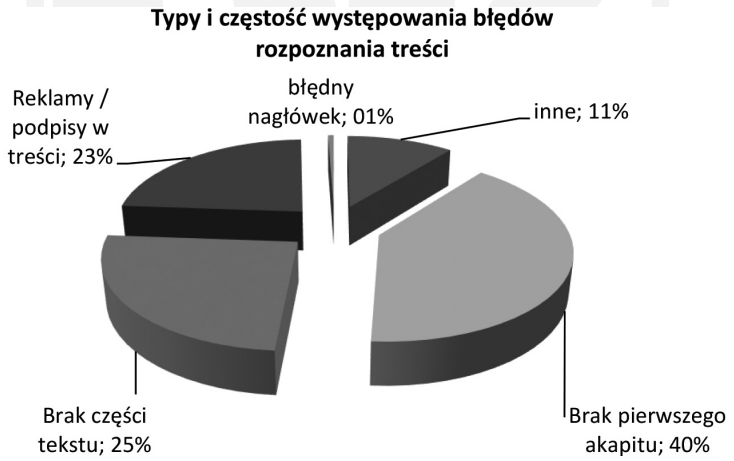
Dokonano również porównania rozpoznawania treści przez prezentowany algorytm ze wspomnianymi narzędziami komercyjnymi. Sprawdzono po 10 losowo wybranych stron (w tym 5 rozpoznanych przez PortalCrawler nieprawidłowo, jeśli tyle znaleziono) dla każdego z 14 badanych portali. Safari Reader oraz iReader największe problemy sprawiało wykrycie pierwszego akapitu tekstu i znalezienie treści krótkich artykułów.

Wyniki porównania PortalCrawler z Apple® Safari Reader kształtują się następująco:

– W 31,4% przypadków otrzymano te same wyniki;
– W 15,7% przypadków Safari Reader zwrócił poprawne wyniki, a PortalCrawler nie;
– W 52,9% przypadków Safari Reader zwrócił niepoprawne wyniki a PortalCrawler tak.

Wyniki porównania PortalCrawler z pluginem iReader kształtują się następująco:

– W 36,4% przypadków otrzymano te same wyniki;
– W 14,3% przypadków iReader zwrócił poprawne wyniki, a PortalCrawler nie;
– W 49,3% przypadków iReader zwrócił niepoprawne wyniki, a PortalCrawler tak.



Rys. 2. Typy i powszechność występowania błędów rozpoznawania (opracowanie własne)

Fig. 2. Types and prevalence of recognition errors (own)

Artykuł zgłoszony do piblikacji w roku 2010.

Literatura

[1] Hemenway K., Calishain T., *100 sposobów na tworzenie robotów sieciowych*, Helion, Warszawa 2004.
[2] Kłopotek M., *Inteligentne wyszukiwarki internetowe*, Akademicka Oficyna Wydawnicza Exit, Warszawa 2001.
[3] Markov Z., Larose D., *Eksploracja zasobów internetowych. Analiza struktury, zawartości i użytkowania sieci WWW*, PWN, Warszawa 2009.
[4] Hłybin M., *Web scraping for fun and profit – Ekstrakcja danych ze stron WWW*, (http://marcinhlybin.com/slides/scraping_article.pdf – odczyt z dnia 10.06. 2011).