Jacek Małyszko[1], Witold Abramowicz[2], Tomasz Wagner[3]

System wyszukiwania aktów prawnych LSIP

1. Wstęp

Akty prawne są wydawane na różnych szczeblach władzy centralnej i samorządowej. Przykładami organów wydających takie akty na szczeblu samorządowym są sejmik województwa i zarząd województwa. Ponieważ uchwały tych organów mogą mieć znaczny wpływ na życie obywateli czy funkcjonowanie instytucji działających na danym obszarze, konieczne jest to, aby każdy mógł łatwo z takimi uchwałami się zapoznać.

Często akty prawne są publikowane w witrynach internetowych wydających je organów (przykładowo w biuletynach informacji publicznej). W wielu przypadkach aktów prawnych wydawanych przez dany organ, a tym samym dostępnych w danej witrynie internetowej, mogą być tysiące. W takich sytuacjach przy braku dodatkowych narzędzi odnalezienie uchwał dotyczących konkretnej sprawy może być zadaniem trudnym i czasochłonnym. Proces ten można znacząco ułatwić poprzez udostępnienie usług automatycznego wyszukiwania aktów w takich witrynach, dzięki któremu użytkownicy mogliby, przykładowo, łatwo zidentyfikować uchwały dotyczące interesującego ich zagadnienia poprzez zgłoszenie do systemu odpowiedniego zapytania, na które mogą składać się opisujące to zagadnienie słowa kluczowe.

Udostępnienie usługi wyszukiwawczej dla aktów prawnych wymaga poniesienia pewnych kosztów przez organ udostępniający uchwały. Usługa wyszukiwawcza musi również zostać odpowiednio przygotowana tak, aby pozwalała na wyszukiwanie z zadowalającą skutecznością i wydajnością. W prezentowanym artykule zostały opisane doświadczenia, jakie we wspomnianych zakresach uzyskano w trakcie prac nad Lokalnym Systemem Informacji Prawnej (LSIP),

[1] Uniwersytet Ekonomiczny w Poznaniu, Wydział Informatyki i Gospodarki Elektronicznej.
[2] Uniwersytet Ekonomiczny w Poznaniu, Wydział Informatyki i Gospodarki Elektronicznej.
[3] Uniwersytet Ekonomiczny w Poznaniu, Wydział Informatyki i Gospodarki Elektronicznej.

będącym z założenia wyszukiwarką aktów prawnych Sejmiku Województwa Wielkopolskiego i Zarządu Województwa Wielkopolskiego.

Prezentowany artykuł ma następującą strukturę. Najpierw pokrótce zaprezentowano przegląd dostępnych rozwiązań możliwych do wykorzystania w celu udostępnienia usługi wyszukiwania aktów prawnych: od gotowych rozwiązań komercyjnych do bibliotek programistycznych o otwartym źródle, ułatwiających przygotowanie własnego rozwiązania. Część trzecia zawiera opis architektury i wybranych zagadnień implementacyjnych opracowanego systemu LSIP, opartego na rozwiązaniu *Elasticsearch*. Natomiast części czwarta oraz piąta zawierają ewaluację wypracowanego systemu dokonaną odpowiednio w kontekście skuteczności wyszukiwania uchwał za jego pomocą i wydajności systemu. Artykuł kończy się krótkim podsumowaniem.

2. Istniejące rozwiązania

Wdrażając usługę wyszukiwawczą, należy dokonać wyboru jej dostawcy. Przede wszystkim trzeba się zdecydować bądź na wdrożenie systemu już istniejącego, bądź też na przygotowanie dedykowanego rozwiązania.

Istnieją gotowe, komercyjne rozwiązania przygotowane z myślą o aktach prawnych, udostępniające usługę ich wyszukiwania. Przykładem jest choćby Zbiór Aktów Prawa Miejscowego firmy ABC PRO[4]. Firma ta w momencie przygotowywania niniejszego artykułu oferowała usługę digitalizacji aktów prawnych, a dla otrzymanych w trakcie digitalizacji plików udostępniała dziedzinową wyszukiwarkę, umożliwiającą przykładowo przeprowadzanie wyszukiwania pełnotekstowego wśród dostępnych aktów.

Jak wspomniano, drugą możliwością przy wdrażaniu usługi wyszukiwawczej jest przygotowanie dedykowanej wyszukiwarki odpowiadającej potrzebom danego podmiotu. Pomimo konieczności poniesienia pewnego wysiłku w trakcie przygotowywania takiego systemu zadanie to nie jest jednak aż tak złożone, jak mogłoby się wydawać. Dzieje się tak dzięki istnieniu wyspecjalizowanych bibliotek programistycznych czy gotowych silników wyszukiwawczych na wolnych licencjach, które można wykorzystać w tym celu. Przykładami takich

[4] http://www.abcpro.pl/Posts/Index/services (data odczytu: 20.11.2015).

rozwiązań są *Apache Lucene*[5] i *Xapian*[6], będące bibliotekami programistycznymi, oraz *Sphinx*[7], *Solr*[8] czy *Elasticsearch*[9], będące bardziej kompleksowymi rozwiązaniami. Przykładowo, *Solr* i *Elasticsearch* korzystają z *Apache Lucene*, ułatwiając korzystanie z możliwości tej biblioteki i dostarczając wiele dodatkowych funkcji, takich jak łatwe tworzenie klastrów wielu komputerów współpracujących ze sobą na potrzeby wyszukiwania czy przechowywanie dokumentów w dedykowanej bazie danych.

Z tego względu przygotowanie wyszukiwarki odpowiadającej określonym potrzebom nie wiąże się z dużym ryzykiem i kompletna usługa wyszukiwawcza, działająca z wysoką skutecznością, może z powodzeniem zostać przygotowana w ciągu kilku miesięcy (w przypadku zaangażowania kilkuosobowego zespołu doświadczonych programistów). Jednak ostateczny wybór tego, czy przygotowywany ma być dedykowany system, czy też warto zakupić już istniejące komercyjne rozwiązanie, musi zostać dokonany z uwzględnieniem specyfiki wdrażającego je podmiotu i wymagań scenariusza wykorzystania takiego systemu.

3. Lokalny System Informacji Prawnej – architektura rozwiązania

Jak wspomniano, Lokalny System Informacji Prawnej (LSIP) został przygotowany jako system wyszukiwawczy dla uchwał Sejmiku Województwa Wielkopolskiego oraz Zarządu Województwa Wielkopolskiego. Pozwala on na przeprowadzanie wyszukiwania z wykorzystaniem kilku kryteriów. Składa się z kilku komunikujących się ze sobą modułów, tak jak zaprezentowano na rysunku 1. Centralnym elementem wypracowanego systemu jest wspomniane w poprzedniej części niniejszego opracowania rozwiązanie *Elasticsearch*, przechowujące dane dotyczące uchwał i odpowiednio je indeksujące. Spośród dostępnych rozwiązań zdecydowano się na wybór właśnie tego systemu ze względu na fakt, że wykorzystuje bibliotekę *Apache Lucene*, która jest *de facto* standardem, jeśli chodzi o biblioteki służące do wyszukiwania informacji, oraz z powodu bogatej dokumentacji rozwiązania oraz łatwości nauki systemu (komunikacja z serwerem wyszukiwawczym odbywa się tu z wykorzystaniem intuicyjnej usługi sieciowej).

[5] https://lucene.apache.org/core (data odczytu: 20.11.2015).
[6] http://xapian.org (data odczytu: 20.11.2015).
[7] http://sphinxsearch.com (data odczytu: 20.11.2015).
[8] http://lucene.apache.org/solr (data odczytu: 20.11.2015).
[9] https://www.elastic.co/products/elasticsearch (data odczytu: 20.11.2015).

Elasticsearch, oprócz samego wyszukiwania uchwał, przechowuje odpowiednio przetworzone dane w wewnętrznym indeksie, który można tu rozumieć jako nierelacyjną bazę danych (NoSQL). W indeksie takim są przechowywane wszystkie metadane uchwał wraz ze ścieżkami do plików PDF zawierających zdigitalizowane uchwały.

Podstawowym wymaganiem w stosunku do *Elasticsearch* była odpowiednia obsługa języka polskiego. System musiał być w stanie przeprowadzać analizę dokumentów i zapytań tak, aby możliwe było odnajdywanie dokumentów również wówczas, gdy wyrazy użyte w dokumencie i w zapytaniu miały inną formę gramatyczną i tym samym nie były jednakowymi łańcuchami znaków. Aby to umożliwić, wykorzystano wtyczkę[10] do *Elasticsearch*, pozwalającą na korzystanie z pakietu *Lucene Morfologik*. Po instalacji wtyczki i odpowiedniej konfiguracji indeksu *Elasticsearch* taka analiza odbywa się w pełni automatycznie przy indeksowaniu dokumentów oraz wykonywaniu zapytań.

Ze względów bezpieczeństwa w przygotowanym systemie baza *Elasticsearch* jest dostępna jedynie dla programów działających na tym samym serwerze. Dopiero te programy, mające odpowiednie mechanizmy kontroli dostępu, pozwalają na uzyskanie dostępu do danych zawartych w indeksie.

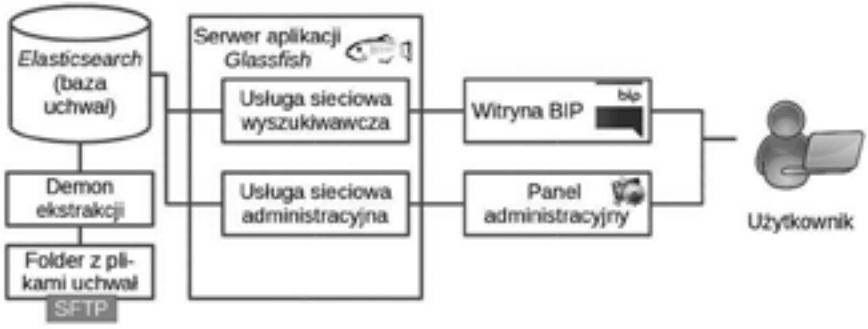
Głównym modułem ładującym dane do indeksu *Elasticsearch* jest tzw. demon ekstrakcji. Jest to program napisany w języku Java, jego działanie polega na monitorowaniu zawartości określonych katalogów w poszukiwaniu pojawiających się tam nowych plików z uchwałami. W przypadku wykrycia takich plików, są one automatycznie przetwarzane (bądź przez odczytania danych z dodatkowego pliku z metadanymi, bądź też – w przypadku braku takiego pliku – przez analizę samego pliku PDF z uchwałą) i dodawane do indeksu. Demon ekstrakcji jest uruchamiany okresowo (raz na dobę, w nocy o ustalonej godzinie) i działa na tym samym serwerze, na którym znajduje się *Elasticsearch*, mając tym samym do niego bezpośredni dostęp, dzięki czemu może modyfikować istniejący indeks. Same pliki z uchwałami mogą być umieszczane w monitorowanym katalogu za pomocą protokołu SFTP czy to przez użytkowników, czy też przez inne systemy.

Dalszy dostęp do indeksu jest możliwy z wykorzystaniem dwóch przygotowanych usług sieciowych, działających zgodnie z modelem REST. Usługi te pełnią role pośredników pomiędzy aplikacjami, z których ostatecznie korzystają użytkownicy, a modułem wyszukiwawczym. Pierwszą z tych usług jest usługa wyszukiwawcza, która jest publicznie dostępna za pośrednictwem protokołu HTTP.

[10] Wtyczkę rozumiemy tu jako dodatkowy moduł oprogramowania przygotowany zgodnie ze zdefiniowanymi dla danego systemu regułami i rozszerzający jego funkcjonalność.

Umożliwia ona jedynie wyszukiwanie uchwał (nie można za jej pomocą dokonywać w indeksie *Elasticsearch* żadnych zmian). Druga usługa, zwana usługą administracyjną, pozwala z kolei na modyfikację danych zawartych w indeksie. Podobnie jak *Elasticsearch*, może być wykorzystywana wyłącznie przez inne programy działające na tym samym komputerze.

Ostatnimi elementami przygotowanego systemu są witryny internetowe, komunikujące się z usługami sieciowymi i pozwalające tym samym bądź to na pobieranie, bądź też na modyfikację danych przechowywanych w indeksie. Użytkownicy mogą przeprowadzać wyszukiwanie z poziomu witryny Biuletynu Informacji Publicznej Urzędu Marszałkowskiego Województwa Wielkopolskiego. Druga witryna, która jest wykorzystywana w systemie, to panel administracyjny, pozwalający na dokonywanie zmian w bazie *Elasticsearch* poprzez komunikację z usługą sieciową administracyjną. Działa on na tym samym komputerze co usługa wyszukiwawcza (i tym samym co baza *Elasticsearch*) i dzięki temu ma do nich dostęp. Aby móc skorzystać z panelu administracyjnego, konieczne jest połączenie z serwerem za pomocą protokołu HTTPS i pomyślne uwierzytelnienie użytkownika. Tak przygotowany system w końcowej fazie prac poddany został audytowi, aby zapewnić bezpieczeństwo przechowywanych w nim danych.



Rysunek 1. Schemat architektury Lokalnego Systemu Informacji Prawnej
Źródło: opracowanie własne.

LSIP umożliwia wyszukiwanie uchwał z wykorzystaniem następujących kryteriów:

- wyszukiwanie pełnotekstowe po sekcji „w sprawie" uchwały (w skrócie opisującym zagadnienie, którego uchwała dotyczy);
- wyszukiwanie pełnotekstowe po całej treści uchwały;

- wyszukiwanie uchwał po dacie uchwalenia (możliwe zdefiniowanie zakresu dat, z którego uchwały interesują użytkownika);
- wyszukiwanie po numerze uchwały (kryterium zwraca maksymalnie jeden wynik, jeśli odnaleziono uchwałę z pasującym numerem);
- ograniczanie wyświetlanych wyników do wybranego organu wydającego (zarząd lub sejmik).

Jak wspomniano, wyszukiwanie pełnotekstowe uwzględnia różne odmiany wyrazów w języku polskim. Dodatkowo, w trakcie wyszukiwania system szuka również dokumentów zawierających podobne (zgodnie z przyjętą miarą odległości łańcuchów znaków) wyrazy tak, aby był w stanie zwrócić wyniki wyszukiwania również w przypadku błędów w pisowni wyrazów (np. literówek) w zapytaniu.

4. Ewaluacja – precyzja wyników wyszukiwania

Dla opisanego w poprzedniej części systemu przeprowadzono ewaluację w celu wykazania skuteczności oraz wydajności, z jakimi pozwala on na wyszukiwanie uchwał. W tej części zostanie opisany pierwszy z eksperymentów, oceniający skuteczność wyszukiwania, rozumianą tutaj za pomocą standardowej miary precyzji działania systemu. Drugi eksperyment, oceniający szybkość jego działania, zostanie opisany w kolejnej części artykułu. Otrzymane rezultaty testów w dużej mierze mogą służyć jako wskazówki na temat potencjalnej wydajności i skuteczności wyszukiwania samego rozwiązania *Elasticsearch*.

Jak wspomniano, pierwszy z przeprowadzonych eksperymentów miał pozwolić na ocenę precyzji działania wyszukiwarki. Precyzja opisuje to, jaka część z rezultatów zwróconych przez system jest prawidłowa, tj. relewantna dla zapytania użytkownika:

$$\text{precyzja} = \frac{\text{poprawne}}{\text{poprawne} + \text{nadmiarowe}}.$$

W powyższym wzorze poprawne to te rezultaty wyszukiwania, które zostały zwrócone w odpowiedzi na zapytanie i są relewantne dla użytkownika, natomiast nadmiarowe to te, które użytkownik uznaje za nierelewantne. W systemach wyszukiwawczych, w których są zwracane wyniki rankingowane ze względu na relewancję dokumentów w kontekście zapytania (a tak właśnie działa oceniany system), stosuje się wariant miary precyzji nazywany precyzją na pozycji

N, oznaczany często jako *P@N*. Dla przyjętej wartości *N* (np. *N* = 10) mówi on, ile z wyników wyszukiwania zwracanych na pozycjach od pierwszej do *N* jest istotnych w kontekście zadanego zapytania. Przykładowo, dla *N* = 10 analizuje się rezultaty na pozycjach od pierwszej do dziesiątej i określa się, ile z tych rezultatów pasuje do sformułowanego zapytania. Jeśli liczbę pasujących rezultatów określimy jako *r*, to precyzję na pozycji *N* określa się za pomocą wzoru *P@N* = *r*/*N*.

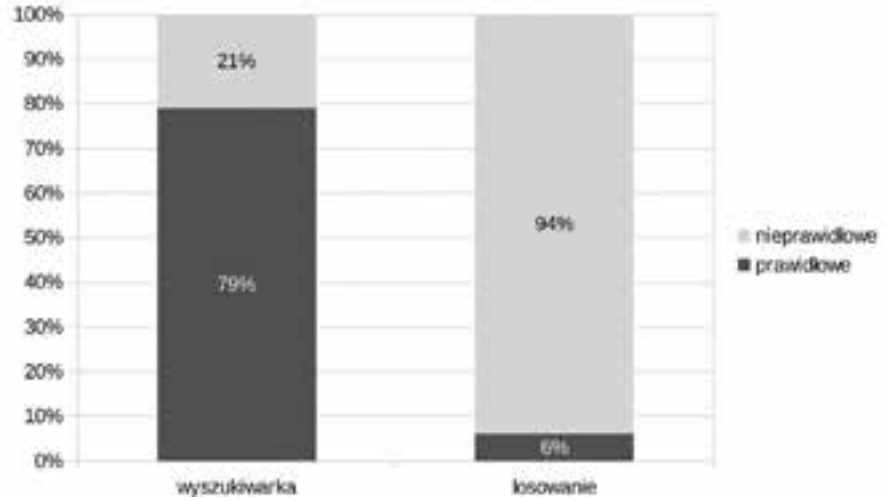
Eksperyment przeprowadzono z wykorzystaniem zapytań złożonych z trzech sformułowań kluczowych, gdzie na każde sformułowanie mogło składać się kilka słów opisujących określone zagadnienie poruszane w akcie. Sformułowania kluczowe zostały przypisane przez eksperta do losowo wybranych uchwał (eksperyment przeprowadzono na korpusie liczącym 5107 uchwał, spośród których losowo wybrano 141). Zapytania były zadawane w wyszukiwarce za pomocą kryterium służącego do wyszukiwania w całym akcie prawnym. Dla tak zadawanych zapytań spośród wszystkich zwróconych wyników losowo wybrano 100 uchwał będących w otrzymanych rankingach na pozycjach od pierwszej do dziesiątej i zapisano, w odpowiedzi na jakie zapytanie zostały one zwrócone. Należy zauważyć, że takie rezultaty w przeważającej większości nie były tymi samymi uchwałami, którym ekspert przypisał sformułowania kluczowe, ale ciągle mogły dotyczyć tej samej tematyki, tzn. zadawane do systemu zapytania mogły trafnie opisywać również te dokumenty. Następnie wylosowano kolejną próbkę 100 uchwał z całej puli 5107 dostępnych dokumentów, które przypisano do losowych zapytań z posiadanej puli.

Posiadano więc 200 par zapytanie–akt prawny, gdzie 100 z nich odpowiadało wynikom działania systemu, a 100 było losowo połączonymi parami. Dla posiadanej kolekcji 200 par dokonano losowego pomieszania ich kolejności oraz zakryto informację o tym, czy dana para pochodzi z wyszukiwarki, czy z losowania. Tak przygotowaną listę 200 par przedstawiono ekspertowi, który w przypadku każdej pary miał zadecydować, czy dana uchwała jest trafnie opisywana przez sformułowania kluczowe, czy też nie. Decyzję ekspert podejmował na podstawie analizy sekcji „w sprawie" poszczególnych uchwał.

Opisana procedura eksperymentu miała zminimalizować ryzyko powstania przekłamań w procesie adnotacji oraz pozwoliła na porównanie precyzji działania wyszukiwarki z sytuacją, gdyby odpowiedzi na zapytania były dobierane przez system w sposób losowy.

Uzyskane rezultaty zaprezentowano na rysunku 2. Uzyskana precyzja na pozycji dziesiątej sięga niemal 80%. Oznacza to, że w odpowiedzi na zapytanie wśród dziesięciu pierwszych wyników zwracanych przez system zazwyczaj ok. ośmiu

z nich będzie istotnych z punktu widzenia zapytania zadanego przez użytkownika. Dla porównania, losowe zwracanie dokumentów pozwala na osiągnięcie *P*@10 na poziomie jedynie 6%.



Rysunek 2. Porównanie precyzji na pozycji dziesiątej dla rezultatów zwracanych przez przygotowany mechanizm wyszukiwawczy z precyzją uzyskaną w przypadku losowego zwracania aktów prawnych

Źródło: opracowanie własne.

Należy zwrócić uwagę na fakt, że precyzja wypracowanego rozwiązania może być zaniżana z przyczyn obiektywnych. Przykładowo, w wielu przypadkach wśród posiadanych aktów prawnych liczba dokumentów dotyczących tematyki opisywanej przez sformułowania kluczowe w zapytaniu może być mniejsza niż dziesięć. Może się nawet zdarzyć, że istnieje tylko jeden dokument, który dotyczy tematu istotnego z punktu widzenia zapytania. W takim wypadku wartość precyzji jest pomniejszona, pomimo że system działa prawidłowo. W związku z tym to, że 21% wyników zwróconych przez system na pierwszych dziesięciu pozycjach nie jest prawidłowe, może wynikać częściowo z faktu, że dla danego zapytania w bazie było mniej niż dziesięć istotnych dokumentów i system zwrócił wśród dziesięciu pierwszych rezultatów w rankingu dokumenty mniej istotne. Jest to problem napotykany podczas ewaluacji wszystkich systemów wyszukiwawczych.

5. Ewaluacja – wydajność systemu

Testy opisane w tej sekcji miały przeanalizować przygotowany system z punktu widzenia szybkości udzielania odpowiedzi na zadawane zapytania oraz sposobu działania systemu pod znacznym obciążeniem, tj. w przypadku wysyłania do niego dużej liczby żądań w krótkim czasie. System wyszukiwawczy musi być odpowiednio przygotowany pod tym kątem ze względu na fakt, że może z niego korzystać jednocześnie wielu użytkowników, a wyszukiwanie pełnotekstowe w dużej kolekcji dokumentów może być zadaniem wymagającym pod względem obliczeniowym.

Do testów wykorzystano program *JMeter*[11]. Za pomocą tego programu do usługi wyszukiwawczej była kierowana duża liczba automatycznie generowanych zapytań. Dla każdego zapytania usługa musiała następnie skomunikować się z *Elasticsearch* w celu przeprowadzenia wyszukiwania, odebrać od niego wyniki i odesłać do nadawcy, tj. do programu *JMeter*, który zbierał statystyki dotyczące czasów oczekiwania na odpowiedzi oraz ewentualnych błędów zwracanych przez usługę.

W eksperymencie wykorzystano dwa serwery o różnych parametrach technicznych dla umożliwienia zaobserwowania różnic w szybkości działania systemu wraz ze wzrostem dostępnych zasobów. Wykorzystano serwery działające w środowisku wirtualizacji KVM z następującymi konfiguracjami:

- serwer 1 (konfiguracja słabsza):
  - przydzielony jeden procesor Intel Xeon E5645 @ 2.4 GHz,
  - pamięć RAM: 8 GB DDR3 1333MHz;
- serwer 2 (konfiguracja mocniejsza):
  - przydzielonych osiem procesorów Intel Xeon E5645 @ 2.4 GHz,
  - pamięć RAM: 16 GB DDR3 1333MHz.

Wejściem do eksperymentu były zapytania kierowane do usługi wyszukiwawczej. Zapytania do serwera były generowane automatycznie, gdzie konkretne wartości zapytań były pobierane losowo z zadanej puli, na którą składały się sformułowania kluczowe przygotowane przez ekspertów, opisane w poprzedniej sekcji. Do usługi wyszukiwawczej trafiały różne zapytania w losowej kolejności, dzięki czemu nie było ryzyka ewentualnych przekłamań wyniku testów z powodu przechowywania przez *Elasticsearch* wyników poprzednich wyszukiwań i wykorzystywania ich w odpowiedzi na kolejne zapytania (ang. *caching*).

[11] http://jmeter.apache.org (data odczytu: 20.11.2015).

W trakcie eksperymentów testowano różne warianty zapytań i ich częstotliwości w celu szerokiej analizy szybkości działania przygotowanego systemu i jego odporności na dużą liczbę zadawanych zapytań. Czynniki, jakie brano tu pod uwagę, to:

- liczba użytkowników jednocześnie korzystających z systemu – czynnik ten analizowano za pomocą odpowiedniego konfigurowania programu *JMeter*; możliwe było ustalanie następujących cech eksperymentu:
  - **liczba wątków**, odpowiadająca liczbie użytkowników jednocześnie korzystających z systemu,
  - **liczba powtórzeń** – reprezentuje liczbę zapytań zadawanych przez jednego użytkownika, gdzie każdy użytkownik wysyła kolejne zapytanie natychmiast po otrzymaniu odpowiedzi na zapytanie poprzednie; w przeprowadzonych testach wartość tę ustalono na 120; warto zauważyć, że w rzeczywistości użytkownicy nie wysyłają kolejnych zapytań, jak tylko otrzymają rezultaty poprzedniego wyszukiwania, tak więc w testach analizowano najbardziej wymagający scenariusz przy danej liczbie użytkowników;
- **średnia liczba słów w zapytaniu** – naturalną sytuacją w systemach wyszukiwawczych jest to, że w przypadku zadawania dłuższych zapytań (składających się z większej liczby słów) czas oczekiwania na odpowiedź może być dłuższy niż wówczas, gdy są zadawane zapytania krótkie, np. jednowyrazowe; osobno zadawano więc zapytania bardzo proste i dłuższe oraz analizowano, jak system zachowuje się w takich sytuacjach;
- **odpytywanie kryterium** – analizowano szybkość odpowiedzi przygotowanego rozwiązania na zapytania kierowane do różnych kryteriów:
  - wyszukiwanie w całej treści uchwały,
  - wyszukiwanie w sekcji „w sprawie”,
  - wyszukiwanie po zakresie dat;
- **limit liczby zwróconych wyników** – system miał zwracać 15 najtrafniejszych wyników wyszukiwania (15 pierwszych pozycji w rankingu).

Wyniki testów zaprezentowano w tabelach 1 oraz 2 – pierwsza z nich prezentuje wyniki otrzymane dla słabszej, a druga dla mocniejszej konfiguracji sprzętowej. Wyniki te pokazują bardzo duży wzrost wydajności działania systemu w przypadku udostępnienia większej mocy obliczeniowej. Należy zwrócić uwagę na fakt, że przy słabszej konfiguracji sprzętowej już w przypadku 15 użytkowników zadających nieustannie długie zapytania (średnio cztery i pół słowa w zapytaniu) do kryterium wyszukującego w całym akcie prawnym długość czasu oczekiwania na odpowiedź systemu staje się nieakceptowalna i wynosi ponad 13 sekund (choć przy krótkich zapytaniach złożonych z pojedynczych

słów szybkość jest wciąż akceptowalna). Natomiast dla mocniejszej konfiguracji sprzętowej czas oczekiwania również dla dłuższych zapytań znacznie spada i wynosi już średnio niewiele ponad 2 sekundy, co już jest w pełni akceptowalnym czasem, analogicznym do czekania na odpowiedź w przypadku choćby popularnych wyszukiwarek internetowych. Warto również zauważyć, że wyszukiwanie w całej treści uchwały jest znacznie bardziej wymagające obliczeniowo niż w przypadku korzystania z innych kryteriów. Już w przypadku wyszukiwania w sekcji „w sprawie" uchwał, pomimo że z technicznego punktu widzenia przetwarzanie jest przeprowadzane dokładnie tak samo jak przy szukaniu w całej treści uchwały, czas oczekiwania na odpowiedź jest znacznie krótszy. Dzieje się tak oczywiście ze względu na fakt, że sekcje te mają znacznie mniej tekstu niż całe uchwały (sekcja „w sprawie" jest tylko niewielkim fragmentem całej uchwały) i indeks utworzony dla tego kryterium jest znacznie mniejszy. Jeszcze szybciej działa wyszukiwanie uchwał po zakresie dat oraz po numerach uchwał (choć wyników dla tego ostatniego nie zawarto w tabelach).

Tabela 1. Wydajność opracowanego systemu dla różnych wariantów zadawanych zapytań w przypadku słabszej konfiguracji sprzętowej serwera

Liczba wątków	Średnia liczba słów w zapytaniu	Kryterium	Średni czas odpowiedzi (ms)	Mediana czasu odpowiedzi (ms)	Odchylenie standardowe czasu odp. (ms)
1	4,5	cała treść uchwały	941	931	476
15	4,5	cała treść uchwały	13 532	13 199	3 128
20	4,5	w sekcji „w sprawie"	1 608	1 517	394
40	nie dotyczy	data od – data do	1 046	1 047	106
10	1	cała treść uchwały	2 429	2 386	799
15	1	cała treść uchwały	3 677	3 565	1 100

Źródło: opracowanie własne.

Przeprowadzony eksperyment wykazał, że prezentowany system, wykorzystujący serwer wyszukiwawczy *Elasticsearch*, jest w stanie działać z akceptowalną wydajnością przy obciążeniu 15 użytkowników zadających ciągle przez dłuższy czas wielowyrazowe zapytania za pomocą kryterium wyszukiwania w całym akcie prawnym. W przypadku korzystania choć przez część użytkowników

z innych kryteriów liczba użytkowników korzystających z systemu mogłaby być dużo większa, a i tak pozwoliłaby na uzyskanie podobnego czasu oczekiwania na odpowiedź. Przeprowadzony eksperyment wykazał również, że w przypadku udostępniania kolejnych zasobów sprzętowych system może działać szybciej. Dodatkowo warto zaznaczyć, że *Elasticsearch* ma możliwość rozproszonego działania na klastrze wielu komputerów, co może jeszcze bardziej zwiększyć jego wydajność.

Tabela 2. Wydajność opracowanego systemu dla różnych wariantów zadawanych zapytań w przypadku mocniejszej konfiguracji sprzętowej serwera

Liczba wątków	Średnia liczba słów w zapytaniu	Odpytywane kryterium	Średni czas odpowiedzi (ms)	Mediana czasu odpowiedzi (ms)	Odchylenie standardowe czasu odp. (ms)
1	4,5	cała treść uchwały	331	317	157
15	4,5	cała treść uchwały	2397	2379	588
20	4,5	w sekcji „w sprawie”	412	389	105
40	nie dotyczy	data od–data do	416	422	48

Źródło: opracowanie własne.

6. Podsumowanie

W artykule zaprezentowano Lokalny System Informacji Prawnej, będący przykładem systemu przygotowanego do wyszukiwania aktów prawnych. Oprócz opisania wypracowanej architektury zaprezentowano również ewaluację systemu pod względem precyzji zwracanych wyników i wydajności działania systemu. Rezultaty opisane w artykule mogą stanowić wskazówki dla instytucji wdrażających podobne systemy w przyszłości i pomóc podjąć decyzję dotyczącą tego, w jaki sposób system wyszukiwawczy w przypadku podobnych potrzeb powinien zostać pozyskany i wdrożony.

Źródła sieciowe

http://jmeter.apache.org (data odczytu: 20.11.2015).
http://lucene.apache.org/solr (data odczytu: 20.11.2015).
http://sphinxsearch.com (data odczytu: 20.11.2015).
http://www.abcpro.pl/Posts/Index/services (data odczytu: 20.11.2015).
http://xapian.org (data odczytu: 20.11.2015).
https://lucene.apache.org/core (data odczytu: 20.11.2015).
https://www.elastic.co/products/elasticsearch (data odczytu: 20.11.2015).

* * *

Legal Information Retrieval System, LSIP

Summary

This article describes a legal information retrieval system developed for the needs of the Marshal Office of the Wielkopolska Voivodship. The architecture of the system is described and its evaluation is presented.

Keywords: information retrieval, search engines, Elasticsearch