Biuletyn WAT
Vol. LXI, Nr 3, 2012

Wykorzystanie formalnej analizy pojęć do analizy dziedzinowych danych tekstowych

MARCIN MICHAŁ MIROŃCZUK

Instytut Podstaw Informatyki PAN, Zakład Sztucznej Inteligencji,
01-248 Warszawa, ul. Jana Kazimierza 5, m.marcinmichal@gmail.com

Streszczenie. W artykule opisano proces projektowania systemu ekstrakcji informacji SEI. Projektowanie tego systemu bazuje na regułach oraz zastosowaniu formalnej analizy pojęć do ich odpowiedniego ułożenia w bazie wiedzy opisywanego systemu.
Słowa kluczowe: formalna analiza pojęć, FCA, ekstrakcja informacji, analiza danych tekstowych, projektowanie ekstraktorów informacji

1. Wstęp

Autor w jednej ze swoich prac zaproponował metodę projektowania systemu informacyjnego SI w oparciu o eksploracyjną analizę danych tekstowych (ang. *text mining*) [1, 2]. Metoda ta została wykorzystana do opracowania przykładowego projektu oraz implementacji SI w postaci operacyjnej bazy danych (rejestru) na temat *punktów czerpania wody — Hydrantów* w skrócie *Hydrantów* dla służb ratowniczych Państwowej Straży Pożarnej PSP [3, 4]. Opracowany rejestr *Hydrantów* powstał w celu dostarczenia informacji m.in. o ich lokalizacji i stanie (sprawny, niesprawny, przyczyny niesprawności) dla Kierujących Działaniami Ratowniczymi KDR podczas akcji ratowniczo-gaśniczych czy też ogólniej dla podmiotów uczestniczących w interwencji [4]. Model rejestru został zaprojektowany oraz uzupełniony potrzebną informacją w oparciu o wspomnianą autorską propozycję projektową SI wykorzystującą dostępną dla autora dokumentację z systemu ewidencji zdarzeń EWID-99 [5-8]. Analizie poddane zostały dane tekstowe z rekordu ww. systemu o nazwie *Dane opisowe do informacji ze zdarzenia.* W rekordzie tym po zakończonej akcji

ratowniczo-gaśniczej KDR umieszcza się opisy przebiegu działań, które wyrażone są za pomocą języka naturalnego. KDR dokonują opisu m.in. tego, jak przebiegały działania ratownicze (zagrożenia i utrudnienia, zużyty i uszkodzony sprzęt), jakie jednostki przybyły na miejsce zdarzenia etc. [4]. Ze względu na to, że znajdujące się tam informacje są nieustrukturalizowane, tj. wyrażone zostały za pomocą języka naturalnego (tekstu), ich przetwarzanie komputerowe jest utrudnione. Przetwarzanie opisywanego rekordu za pomocą algorytmów do wyszukiwania informacji (ang. *information retrieval*) może dawać nieoczekiwane efekty w przypadku próby wyszukania informacji np. na temat lokalizacji i stanu hydrantów, z których można zatankować wodę [3].

Z powyższych względów autor zaproponował proces strukturalizacji informacji zawartych w danych tekstowych rekordu *Dane opisowe do informacji ze zdarzenia* i zaprojektował bardziej odpowiednie rozwiązanie w postaci SI uzupełnionego potrzebną informacją w procesie jej ekstrakcji z ww. rekordu. W artykule tym autor opisał badania oraz ich wyniki związane z realizacją procesu ekstrakcji informacji z pola *Dane opisowe do informacji ze zdarzenia*. W punkcie 2 autor przedstawił podstawy projektowania systemu ekstrakcji informacji SEI. Opisał w nim możliwości wykorzystania reguł oraz Formalnej Analizy Pojęć i kraty pojęć do reprezentacji wiedzy na temat ekstrakcji informacji. Ogólny problem reprezentacji wiedzy i reguł stanowi zarówno autorską alternatywę dla propozycji wysuniętej w pracy omawiającej zastosowania sieci Petriego do tego celu [9]. W punkcie 3 przytoczono rezultaty eksperymentów związanych z wytworzonym oprogramowaniem do ekstrakcji informacji na temat *punktów czerpania wody — Hydrantów* z dostępnej dokumentacji w postaci *Danych opisowych do informacji ze zdarzenia*. W ostatnim 4 punkcie omówiono wnioski, jakie płyną z aktualnie przeprowadzonych badań, oraz przedstawiono dalsze kierunki rozwoju projektu.

2. Projektowanie systemu ekstrakcji informacji

Ekstrakcja informacji (ang. *information extraction*) jest to identyfikacja, polegająca na odnajdywaniu właściwej informacji w nieustrukturalizowanych danych tekstowych wyrażonych za pomocą języka naturalnego. Proces ten jest zgodny z klasyfikacją polegającą na strukturyzowaniu poprzez nadawanie klas semantycznych dla wybranych elementów tekstu. Proces ten czyni informację zawartą w tekście bardziej właściwą i przydatną w realizowanych zdaniach [10]. Ekstrakcja informacji nazywana jest także ekstrakcją (rozpoznawaniem) encji i modelowaniem ich relacji (ang. *concept/entity extraction, named entity recognition*) [11], jednak jest to ograniczenie definicji ekstrakcji informacji tylko do jednego z podstawowych jej zadań. Wymienione zadanie polega na pozyskiwaniu z dokumentów tekstowych nazw obiektów, np. osób, oraz na wyznaczaniu związków i relacji pomiędzy

wydobytymi obiektami. W ogólnym przypadku można pozyskiwać w ten sposób z tekstu nazwy miast, imiona i nazwiska osób, kody pocztowe, numery PESEL itp. W przypadku szczególnym, który stanowią analizy raportów z akcji ratowniczo-gaśniczych, można pozyskać informacje na temat: liczby akcji, w których brała udział dana osoba, liczby ofiar śmiertelnych zarejestrowanych w akcji ratunkowej. Przy pomocy tak wydobytych cech można sprawdzać, czy analizowany obiekt, np. osoba, nie zmieniła rangi (nie awansował na wyższy stopień), czy nie zaszły jakieś kluczowe zmiany na obiekcie, np. niedziałające hydranty, czy też w mediach nie pojawiły się informacje o zdarzeniach określonego typu (katastrofy, wypadki, akty terrorystyczne). Do pozostałych podstawowych zadań z zakresu ekstrakcji informacji należą: rozróżnianie wyrażeń rzeczownikowych z relacją gramatyczną (ang. *noun phrase coreference resolution*), rozpoznawanie ról semantycznych (ang. *semantic role recognition*), rozpoznawanie relacji między encjami (ang. *entity relation recognition*) czy też rozpoznanie czasu oraz określanie linii czasu zachodzenia zdarzeń (ang. *timex and time line recognition*) [10].

Do typowych problemów, które muszą być rozwiązane przez system ekstrakcji informacji, należą następujące zagadnienia [10, 12]:

a) rozpoznanie i utworzenie skryptów (scenariuszy) będących kompleksowym opisem zdarzeń,
b) utworzenie modeli (wzorców) wynikających z tekstu,
c) podział tekstu na ciągi zdań,
d) podział zdań na wyrażenia z przypisanymi wartościami cech gramatycznych,
e) rozpoznawanie skrótów, fraz rzeczownikowych, nazw bez wnikania w ich strukturę wewnętrzną i ich funkcje w zdaniu,
f) budowanie przybliżonej struktury zdania (np. drzewa rozbioru) ze słów i wcześniej rozpoznanych elementów,
g) wypełnienie przygotowanych modeli informacjami z tekstu.

Pierwsze cztery ww. zadania mają charakter ogólny i ich rozwiązania mogą być stosowane w wielu różnych systemach. Ostatnie zadanie natomiast jest ściśle związane z konkretnym zastosowaniem. Wzorce i reguły ich wypełniania zależą od tego, jakie informacje są poszukiwane.

Przytoczone wyżej pojęcia ekstrakcji informacji wiążą się najczęściej z normalizacją i identyfikacją w tekście wybranych typów danych oraz ich powiązań. Autor w swych dotychczasowych badaniach zrealizował zmodyfikowany przez siebie SEI. Do podstawowych modyfikacji należało usunięcie elementów związanych z zagadnieniami realizowanymi przez system ekstrakcji opisanymi w punktach d) oraz częściowo e) i f). Punkt e) był realizowany w oparciu o skonstruowane słowniki komputerowe zawierające nazwy ulic czy też możliwych uszkodzeń hydrantów. Natomiast punkt f) był implementowany bez drzew rozbioru. Budowa przybliżonej struktury zdania opierała się o wyrażenia regularne utworzone na podstawie

wykrytych reguł, które zostały opisane dalej w artykule. Całościowy zmodyfikowany tor ekstrakcji informacji zaprezentowano na przykładzie. Założono, że dostępny jest raport w następującej postaci:

Po przybyciu na miejsce zdarzenia stwierdzono, iż na balkonie 3. kondygnacji otwartym ogniem palą się szafki, koszyki wiklinowe, szmaty oraz okna i przylegająca elewacja. Działania polegały na podaniu dwóch prądów wody w natarciu: 1 z ziemi na balkon, 2 — prowadzony klatką schodową do mieszkania. Zniszczeniu uległy drzwi wejściowe podczas wyważania. Pomieszczenie oddymiono, miejsce zdarzenia przekazano właścicielowi -----. Samochód zatankowano przy ul. Łabiszyńskiej nr 1673 — sprawny.

Na podstawie analizy raportów autor ustalił, że można w nich wyróżnić pięć typów klas, do których mogą należeć znajdujące się w nich segmenty (zdania). Tymi klasami były klasa: operacje, sprzęt, szkody, meteo i ogólna [3, 13]. Po procesie segmentacji raportu oraz zaklasyfikowaniu jego poszczególnych segmentów do ww. klas otrzymywany jest półustrukturalizowany użyteczny przypadek zdarzenia [1]. Przykład takiego przypadku przedstawiono w tabeli 1.

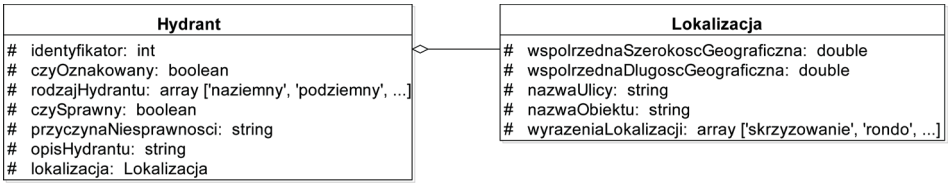
Tabela 1

Przykładowy półustrukturalizowany raport z zaklasyfikowanymi segmentami do odpowiednich klas. Źródło: opracowanie własne

Segment	Klasa semantyczna
Po przybyciu na miejsce zdarzenia stwierdzono, iż na balkonie 3. kondygnacji otwartym ogniem palą się szafki, koszyki wiklinowe, szmaty oraz okna i przylegająca elewacja	opis
Działania polegały na podaniu dwóch prądów wody w natarciu: 1 z ziemi na balkon, 2 — prowadzony klatką schodową do mieszkania	operacje
Zniszczeniu uległy drzwi wejściowe podczas wyważania. Pomieszczenie oddymiono, miejsce zdarzenia przekazano właścicielowi -----	zniszczenia
Pomieszczenie oddymiono, miejsce zdarzenia przekazano właścicielowi -----	operacje
Samochód zatankowano przy ul. Łabiszyńskiej nr 1673 — sprawny	sprzęt

Po sklasyfikowaniu segmentów do poszczególnych klas autor wybrał do modelowania, strukturalizowania klasę, sprzęt i opisy dotyczące hydrantów. Na podstawie dalszych analiz autor ustalił dla tych opisów ustrukturalizowany model, do którego w dalszej kolejności ekstrahowana była informacja na temat *Hydrantów*. Model ten wyrażony w notacji obiektowej w postaci klas przedstawia rysunek 1.

Rysunek 1 przedstawia finalną wersję modelu reprezentującego *Punkt czerpania wody–Hydrant*. Model ten składa się z dwóch klas *Lokalizacja* oraz *Hydrant*. Klasa *Lokalizacja* składa się z pięciu atrybutów przechowujących informacje



Rys. 1. Finalna wersja utworzonego modelu *Punkt czerpania wody — Hydrant*. Źródło: opracowanie własne

o m.in. szerokości i długości geograficznej czy też nazwie ulicy, na której znajduje się hydrant itd. Należy zauważyć, że atrybuty *wspolrzednaSzerokoscGeograficzna* oraz *wspolrzednaDlugoscGeograficzna* nie pochodzą bezpośrednio z analizowanych tekstów. Pośrednio jednak można je uzyskać z procesu translacji nazwy ulicy, atrybutu *nazwaUlicy* na współrzędne geograficzne. Proces taki umożliwiają interfejsy programowania aplikacji (ang. *application programming interface — API*) dostarczane przez niektóre firmy zajmujące się mapami cyfrowymi [14]. Należy także zwrócić uwagę, że położenie hydrantu pomimo translacji dalej będzie względne, tj. będzie odnosiło się do jakiegoś obiektu w przestrzeni, np. numeru bloku, przy którym stoi dany hydrant, a nie do położenia bezpośredniego samego hydrantu. Niemniej sytuacja taka nie powinna pogarszać zlokalizowania hydrantu na miejscu lub w pobliżu interwencji. Utworzona podczas modelowania klasa *Hydrant* zawiera siedem atrybutów przechowujących informacje o m.in.: numerycznym identyfikatorze hydrantu, o tym, czy jest oznakowany etc.

Utworzony i wyżej opisany model może być utrwalany w dowolnej bazie danych. Preferowanym rozwiązaniem jest katalogowa baza danych w celu utrzymania spójności z wcześniejszymi propozycjami i badaniami innych autorów nad ich wykorzystaniem w PSP [15-20]. Niemniej dla rozważań autora sposób utrwalania, tj. w jaki sposób dane będą utrwalane i gdzie, nie odgrywa znaczącej roli. Ważny jest natomiast sposób przejścia z informacji nieustrukturalizowanej, opisanej językiem naturalnym, do ustrukturalizowanej w postaci ww. modelu opisu hydrantów. Z segmentu należącego do klasy *sprzęt* i opisującego m.in. hydrant, który przedstawia tabela 1, w postaci *samochód zatankowano przy ul. Łabiszyńskiej nr 1673 — sprawny*, można wyekstrahować do rozpatrywanego modelu następujące dane (atrybut, wartość, opis):

— atrybut używalność przyjmuje wartość prawdy logicznej (ang. *true*), wyrażenie *zatankowano* w danym zdaniu sugeruje, że hydrant działa,
— atrybut numer identyfikacyjny przyjmuje wartość *1673*, który występuje w danym zdaniu,
— atrybut położenie względne może przyjąć wartość strukturalną złożoną z nazwy ulicy o wartości *Łabiszyńskiej*.

Jak zademonstrowano na powyższym przykładzie, ekstrakcja encji polega na rozpoznawaniu i klasyfikowaniu wykrytych wyrażeń z tekstu, takich jak: nazwy ulic,

identyfikatory, stan obiektów etc. do utworzonego modelu. Sposób projektowania ekstraktora, reguły ekstrakcji oraz przykładowe działanie ekstraktora zaprezentowano w kolejnych podpunktach artykułu.

2.1. Reguły wykrywania wzorca i ekstrakcji informacji

W celu pozyskiwania wybranych informacji z tekstu (segmentu, zdania,) został zaprojektowany i zaimplementowany program, który używał bazy reguł wykrywania wzorca BRWW i bazy reguł ekstrakcji BRE. Składał się on więc z dwóch warstw przetwarzania segmentu *s*. Pierwsza warstwa wykorzystująca BRWW służyła do wykrywania zdefiniowanego wzorca w segmencie. Druga natomiast używała BRE i z segmentu *s* wyodrębniała informację. BRWW zawiera reguły wykrywania w następującej postaci:

$$r_{wykrywania_i} : zawieraWzorzec(s,l_1) \wedge \ldots \wedge zawieraWzorzec(s,l_i) \rightarrow c_i. \quad (1)$$

Regułę (1) można odczytać w następujący sposób: jeśli segment *s* zawiera wzorzec l_1 (literał opisujący wzorzec wykrywania) i segment *s* zawiera wzorzec l_i, to wykryto schemat. Wykrycie wszystkich wzorców, a więc opisanego przez nich schematu, tj. spełnienie takiej reguły (1), prowadzi do konkluzji c_i w postaci prawdy lub fałszu logicznego i zwrócenia identyfikatora reguły. W przypadku prawdy logicznej uruchamiany jest drugi składnik SEI w postaci BRE. Na podstawie identyfikatora rozpoznawana jest reguła ekstrakcji i uruchomiona zostaje reguła służąca do procesu ekstrakcji informacji z wykrytego schematu. Fakt ten można zapisać w następujący sposób:

$$r_{ekstrakcji_i} : c_i \rightarrow wyodrebnijInformacje(s,l_{e1}) \wedge \ldots \wedge wyodrebnijInformacje(s,l_{ei}). \quad (2)$$

Regułę (2) można odczytać w następujący sposób, z segmentu *s* wyodrębnij informację zgodnie z wzorcem ekstrakcji l_{e1} (literał opisujący wzorzec ekstrakcji), operację ekstrakcji przeprowadź kolejno dla segmentu *s* i wzorców ekstrakcji l_{ei}.

W celu zademonstrowania działania programu posłużono się przykładem. Założono, że BRWW składa się z jednej reguły wykrywania $r_{wykrywania1}$, której spełnienie wywołuje odpowiednią regułę ekstrakcji informacji $r_{ekstrakcji1}$. Reguła wykrywania wzorca składa się z następujących elementów:

$$\begin{aligned} &r_{wykrywania_1} : zawieraWzorzec(s, wyrazenie\_hydrant\_numer\_wzorzec\_numer) \wedge \\ &zawieraWzorzec(s, wyrazenie\_ulica\_wzorzec\_ulicy) \rightarrow c_1 \\ &r_{ekstrakcji_1} : c_1 \rightarrow wyodrebnijInformacje(s, numer\_hydrantu) \wedge \\ &wyodrebnijInformacje(s, nazwa\_numer\_ulicy). \end{aligned} \quad (3)$$

Następnie rozpatrzono segment *s* w postaci *Sprawdzono hydrant o numerze 192838 przy ulicy Mickiewicza 2.* W segmencie *s* po przejściu przez pierwszą warstwę oprogramowania, która używa BRWW, zostanie zwrócona konkluzja c_1. W segmencie tym za pomocą pierwszej części rozpoznawania schematu, tj. za pomocą funkcji *zawieraWzorzec(s, wyrazenie_hydrant_numer_wzorzec_numer)*, zostanie wykryty fakt, że segment *s* zawiera wyrażenie *hydrant* oraz *numer*, po którym następuje kombinacja cyfr w postaci *192838*. Druga część reguły w postaci funkcji *zawieraWzorzec(s, wyrazenie_ulica_wzorzec_ulicy)* wykryje fakt, że segment *s* zawiera wyrażenie *ulica*, po którym występuje właściwa nazwa ulicy *Mickiewicza* wraz z numerem *2*. Wszystko to sprawia, że reguła przyjmuje wartość *true* wskazującą na to, że wykryto odpowiedni schemat. Po jego wykryciu uruchamiana jest druga warstwa programu używająca BRE. Wywołana reguła ekstrakcji $r_{ekstrakcji1}$ wydobędzie z segmentu *s* za pomocą funkcji *wyodrebnijInformacje(s, numer_hydrantu)* numer hydrantu *192838*, a następnie za pomocą funkcji *wyodrebnijInformacje(s, nazwa_numer_ulicy)* jego lokalizację — *Mickiewicza 2*.

W przypadku mało skomplikowanej dziedziny, która produkuje małą liczbę reguł oraz literałów zarówno opisujących wykrywanie wzorca jak i ekstrakcję informacji, nie ma problemu z manualnym układaniem w stos i ich implementacją w oprogramowaniu. Najpierw należy wykryć segmenty zawierające jak najwięcej informacji, a więc największą liczbę odpowiednich literałów. W przykładzie jeśli byłby dostępny segment *s* w postaci np. *Sprawdzono hydrant o numerze 192838*, to należałoby dodać odpowiednio po jednej regule do BRWW i BRE:

$$\begin{aligned} &r_{wykrywania_2} : zawieraWzorzec(s, wyrazenie\_hydrant\_numer\_wzorzec\_numer) \rightarrow c_2 \\ &r_{ekstrakcji_2} : c_2 \rightarrow wyodrebnijInformacje(s, numer\_hydrantu). \end{aligned} \tag{4}$$

Dzięki tym regułom wykryto by fakt, że segment *s* zawiera tylko informację o identyfikatorze hydrantu *192838*, który by następnie wyekstrahowano.

Termin mało skomplikowana dziedzina dotyczy założenia, że opisy np. położenia hydrantów byłyby w raportach odnotowywane zawsze tak samo, tj. poprzez użycie standardowego szyku wyrażeń budujących segment opisujący hydrant. W rzeczywistości szyk wyrażeń jest różny. Badania autora dotyczące analizy dokumentacji i opisów m.in. lokalizacji, stanu czy też typu *punktów czerpania wody — Hydrantów* wykazały, że opisy te zawierają 14 charakterystycznych wzorców (literałów), za pomocą których można odnaleźć i wydobyć z segmentu potrzebną informację. Autor zbadał 1416 segmentów opisujących różne fakty dotyczące *punktów czerpania wody — Hydrantów*. Częstotliwość występowania literałów i procentowego odwzorowania za ich pomocą zbioru segmentów wraz z wartościami skumulowanymi zaprezentowano w tabeli 2.

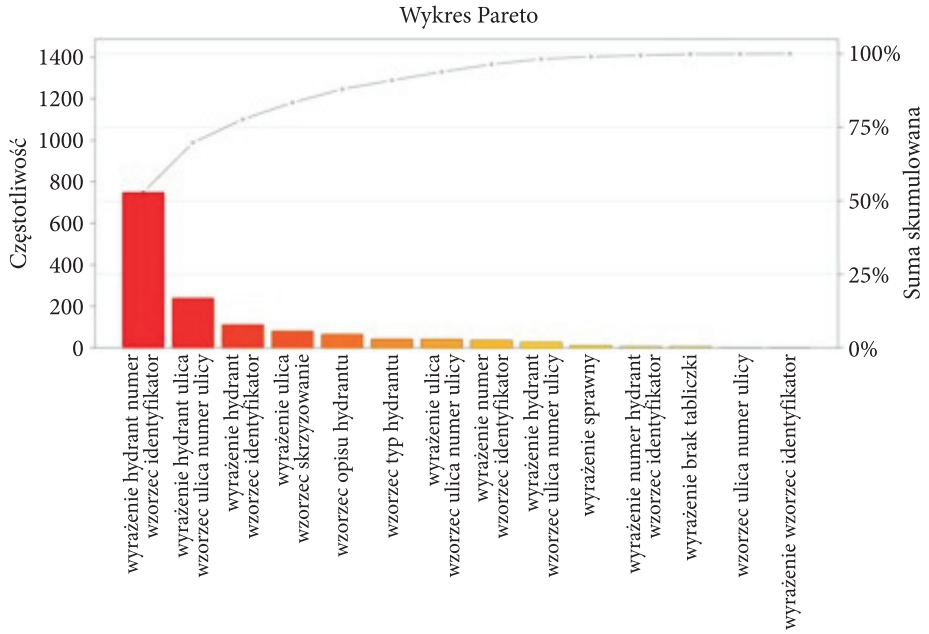
Tabela 2

Dane na temat odwzorowania dostępnych segmentów opisujących *Hydranty* przez literały. Źródło: opracowanie własne

Literał	Częstotliwość	Skumulowana częstotliwość	Procent [%]	Skumulowany procent [%]
Wyrażenie hydrant numer wzorzec identyfikator	749	749	52,89548	52,89548
Wyrażenie hydrant ulica wzorzec ulica numer ulicy	239	988	16,87853	69,77401
Wyrażenie hydrant wzorzec identyfikator	112	1100	7,909605	77,68362
Wyrażenie ulica wzorzec skrzyżowanie	81	1181	5,720339	83,40395
Wzorzec opisu hydrantu	65	1246	4,590395	87,99435
Wzorzec typ hydrantu	42	1288	2,966102	90,96045
Wyrażenie ulica wzorzec ulica numer ulicy	41	1329	2,89548	93,85593
Wyrażenie numer wzorzec identyfikator	36	1365	2,542373	96,39831
Wyrażenie hydrant wzorzec ulica numer ulicy	26	1391	1,836158	98,23446
Wyrażenie sprawny	11	1402	0,776836	99,0113
Wyrażenie numer hydrant wzorzec identyfikator	6	1408	0,423729	99,43503
Wyrażenie brak tabliczki	6	1414	0,423729	99,85876
Wzorzec ulica numer ulicy	1	1415	0,070621	99,92938
Wzorzec identyfikator	1	1416	0,070621	100

Na podstawie danych, które zaprezentowano w tabeli 2, został wykonany wykres Pareto. Wykres ten przedstawiono na rysunku 2.

Na wykresie Pareto zaprezentowano procentowy udział literałów reguł w odwzorowywaniu segmentów opisujących *Hydranty*. Widać, że najwięcej segmentów zawiera wzorzec zdefiniowany za pomocą atrybutu *wyrażenie hydrant numer wzorzec identyfikator* z pozostałymi kombinacjami atrybutów. Ponad 50% segmentów pasuje do tego schematu, a więc 50% segmentów zawiera takie wyrażenia jak *hydrant*, *numer* i *identyfikator*. Na drugim miejscu znajduje się wzorzec w postaci *wyrażenie hydrant ulica wzorzec ulica numer ulicy*. Ponad 15% segmentów można dopasować do takiego wzorca wraz z pozostałymi kombinacjami atrybutów. Pozostałe wzorce i kombinacje atrybutów opisują mniej niż 10% segmentów. Ponadto warto zauważyć, że nie istnieją segmenty złożone tylko z samego wyrażenia *niesprawny*



Rys. 2. Wykres Pareto danych na temat odwzorowania dostępnych segmentów opisujących *Hydranty* przez literały. Źródło: opracowanie własne, wykonane przy wykorzystaniu pakietu [21]

czy zawierające tylko *wzorzec skrzyżowania*. Atrybuty te są najczęściej elementami bardziej rozbudowanych schematów opisu segmentów poprzez odpowiednią kombinację atrybutów.

Zwiększanie się ilości informacji, które należy wyodrębnić z segmentów dotyczących np. typu, lokalizacji czy stanu technicznego hydrantu, czy też różne kombinacje literałów, powoduje, że manualna próba konstrukcji oprogramowania, tj. stosu reguł wykrywających jak i wydobywających informacje, może stać się kłopotliwa w realizacji. Z tego względu autor proponuje półautomatyczne rozwiązanie projektowania i implementacji takiego stosu za pomocą niżej opisanej formalnej analizy pojęć. Półautomatyczny oznacza, że manualnie należy określić i przyporządkować literały l_i do badanych segmentów s. Następnie na podstawie tak utworzonej relacji należy zbudować automatycznie model w postaci kraty pojęć, którą w dalszej kolejności implementuje się w SEI. Szczegóły dotyczące zarówno FCA jak i projektowania za jej pomocą ww. baz reguł wykorzystywanych w SEI opisano w podpunkcie 2.2.

2.2. Formalna analiza pojęć w ekstrakcji informacji

W niniejszym punkcie opisano podstawy teoretyczne formalnej analizy pojęć oraz przedstawiono jej powiązanie z regułami opisanymi w podpunkcie 2.1 służącymi do wykrywania wzorców oraz ekstrakcji informacji. W dalszej części

opracowania przedstawiono także przykładowe jej zastosowanie w projektowaniu i w implementacji na jej podstawie SEI.

2.2.1. Opis formalnej analizy pojęć

Formalna analiza pojęć wprowadzona została przez Rudolfa Wille'a w 1984 roku. Jej koncepcja zbudowana została na teorii sieci i częściowego porządku, które zostały rozwinięte przez Birkhoffa i innych w latach 30. XX wieku [22-25]. FCA służy m.in. do matematyzacji pojęcia (*Konceptu*) dostarcza także formalne narzędzie do analizy danych i reprezentacji wiedzy. Do wizualizacji zachodzących relacji pomiędzy wykrytymi pojęciami służy w FCA krata pojęć (ang. *concept lattice*). Krata pojęć graficznie może być zaprezentowana za pomocą diagramu liniowego nazywanego także diagramem Hassego (ang. *Hasse diagram*) [26, 27]. Diagram ten służy do konstruowania hierarchii pojęć. Składa się z węzłów (wierzchołków) oraz krawędzi. Każdy wierzchołek reprezentuje pojęcie, natomiast krawędzie służą do połączenia wierzchołków w określony sposób [26]. Formalna analiza pojęć jest jedną z wielu metod wykorzystywanych w inżynierii wiedzy do odkrywania i budowania ontologii specyficznej dla rozważanej dziedziny z m.in. danych tekstowych [174-176]. Aktualnie FCA stosowana jest ponadto w dziedzinach z zakresu m.in. [22]: psychologii, socjologii, antropologii, medycyny, biologii, lingwistyki, matematyki czy też informatyki. Autorowi najbliższe są zastosowania z zakresu technik informacyjnych i informatyki, w których formalna analiza pojęć wykorzystywana jest w szczególności do realizacji zadań z zakresu:

- wydobywania hierarchii pojęć (ang. *concept hierarchies*) z tekstu dla systemów bazujących na wiedzy [28], tj. systemów komputerowych stosujących wiedzę z danej dziedziny zapisanej w bazie wiedzy [29]. Wydobyta hierarchia pojęć stanowi taksonomię polegającą na klasyfikacji (uporządkowaniu) jednostek systematycznych w kategorie,
- odnajdywania grupy dokumentów dzielących te same atrybuty, zadanie to jest ważnym elementem m.in. w: eksploracyjnej analizie tekstów, przetwarzaniu informacji czy też wyszukiwaniu informacji w zbiorze dokumentów tekstowych. W ostatnim przykładzie FCA pełni najczęściej rolę silnika wspierającego systemy wyszukiwania informacji w tekście [27]. Natomiast diagramy liniowe służą do tworzenia i wizualizacji hierarchii dokumentów oraz ich powiązań,
- analizy kodu źródłowego [30], a w szczególności pozyskiwania i grupowania wzorców projektowych [31, 32] jak i analizy, projektowania, tworzenia oraz refaktoryzacji hierarchii klas z zakresu paradygmatu projektowania obiektowego [23, 26, 33-39]. FCA w tym przypadku służy więc do zarządzania i rozwoju oprogramowania w inżynierii programowania [40] jak i modelowania całych systemów informatyczno-informacyjnych [41, 42],

— wspierania projektowania systemów CBR [43] oraz ich udoskonalania [44] poprzez np. implementację w silnikach wspierających grupowanie i selekcję przypadków zdarzeń [45, 46],
— wykrywania zależności funkcyjnych (ang. *functional dependencies*) w relacyjnych bazach danych [47],
— tworzenia metod półautomatycznych do konstruowania wybranych ontologii [48-50].

Dokładny przegląd zastosowań formalnej analizy pojęć z zakresu odkrywania wiedzy można znaleźć w literaturze [51]. W przeglądzie tym autorzy zebrali artykuły z lat 2003-2009 i określili procentowy udział formalnej analizy pojęć w badaniach związanych z odkrywaniem wiedzy.

Autorska propozycja projektowania bazy wiedzy SEI poprzez analizę sekcji *Dane opisowe do informacji ze zdarzenia* bazuje na formalnej analizie pojęć oraz diagramach liniowych do wizualizacji wykrytych relacji między obiektami (regułami). Ogólnie formalna analiza pojęć zawiera trzy podstawowe kroki, na które składają się następujące elementy:

— zdefiniowanie obiektów *O*, atrybutów *C* oraz relacji incydencji,
— zdefiniowanie kontekstu formalnego *K* w terminach obiektu, atrybutu i relacji incydencji,
— zdefiniowanie pojęcia formalnego dla danego kontekstu formalnego.

Kontekstem formalnym *K* jest nazywana następująca trójka [48]:

$$K(O, C, R), \tag{5}$$

gdzie: *O* — niepusty zbiór obiektów;
C — niepusty zbiór atrybutów;
R — binarna relacja między obiektami a atrybutami;
olc — relacja *l* opisująca fakt, że obiekt *o* posiada atrybut *m*, atrybut *m* można przypisać do obiektu *o*.

Informację o zależności pomiędzy wykrytymi obiektami oraz określającymi je atrybutami można wyrazić za pomocą tabeli. Przykładowo taką formę prezentacji relacji przedstawiono w tabeli 3.

W tabeli 3 zamieszczono informację o zależnościach pomiędzy wykrytymi obiektami oraz atrybutami. W przypadku gdy do obiektu *o* pasuje przynajmniej jeden atrybut c_k, odnotowywane jest to w tablicy poprzez wstawienie do odpowiedniej jej komórki wartości 1, w przeciwnym razie komórka tablicy pozostaje pusta. W ten sposób tworzone są relacje między obiektami i opisującymi je atrybutami. Z kontekstu formalnego *K* można wywnioskować następujące zależności:

— dowolny podzbiór obiektów A, $A \subseteq O$ generuje zbiór atrybutów *A*', które można przypisać wszystkim obiektom z A, np. $A = \{o2, o3\} \rightarrow A' = \{c2, c3\}$,

— dowolny podzbiór atrybutów B, $B \subseteq C$ generuje zbiór obiektów B' posiadających wszystkie atrybuty z B np. $B = \{c2\} \rightarrow B' = \{o2, o3\}$.

Tabela 3

Tabela dla dowolnego formalnego kontekstu *K*. Źródło: opracowanie własne

Obiekty	Atrybuty				
	c_1	c_2	c_3	…	c_k
o_1	1				
o_2		1	1		
o_3		1			
…					
o_n					

Formalne pojęcie (ang. *formal concept*) kontekstu $K(O, C, R)$ stanowi para uporządkowana (A, B), gdy [48]:

— $A = B' = \{ o \in O : \forall c \in B \; olc \}$ — ekstensja (A, B),

— $B = A' = \{ c \in C : \forall o \in A \; olc \}$ — intensja (A, B).

Z każdym pojęciem związane są jego: ekstensja i intensja. Ekstensja to klasa przedmiotów (obiektów) opisywanych przez pojęcie. Natomiast intensja to klasa cech (własności, atrybutów) wspólnych dla wszystkich przedmiotów z ekstensji.

Pojęcia (A_1, B_1) oraz (A_2, B_2) kontekstu $K(O, C, R)$ są uporządkowane względem relacji, którą można zdefiniować w następujący sposób [48]:

$$(A_1, B_1) \leq (A_2, B_2) \overset{def}{\Longleftrightarrow} A_1 \subseteq A_2 \Leftrightarrow B_2 \subseteq B_1 . \tag{6}$$

Zbiór wszystkich pojęć S kontekstu K wraz z relacją $\leq (S(K), \leq)$ tworzą kratę, która w analizie FCA nazywana jest kratą pojęć formalnego kontekstu $K(O, C, R)$ [48].

Dokonując pewnego spostrzeżenia oraz odpowiednich modyfikacji oznaczeń, reguły ekstrakcji można powiązać z kratą pojęć i tym samym uzyskać algorytm konstruowania ekstraktora informacji. Najpierw należy utożsamić literały (wzorce) *l* występujące w regułach z atrybutami *c* z formalnej analizy pojęć. Następnie trzeba zbudować kratę pojęć poprzez wyznaczenie relacji binarnej między segmentami *s* stanowiącymi obiekty *o* a atrybutami *c*. Węzły–pojęcia utworzonej w ten sposób kraty pojęć są tożsame w tym przypadku z regułami ekstrakcji. Tak więc w pojęciach zakodowana zostaje informacja o regułach wykrywania schematów w segmentach. Pojęcie zawiera informacje o poszczególnych cechach (literałach) wykrywania jak i ekstrakcji informacji przypisanych do grupy obiektów w postaci segmentów. Ponadto dzięki utworzonej kracie, która opisuje relacje między pojęciami, otrzymany zostaje naturalny porządek ułożenia reguł w SEI. Poczynając od korzenia

i przechodząc ku górze kraty, najpierw dopasowywane są wszystkie wzorce opisane atrybutami *c* (literałami, *l*) do segmentu *s*.

2.2.2. Przykład zastosowania

W poniżej zaprezentowanym przykładzie rozważono siedem przykładowych segmentów (s_1,…, s_7) opisanych pięcioma atrybutami *c* w postaci: wyrażenie hydrant wzorzec ulica numer ulicy, wyrażenie sprawny, wzorzec typ hydrantu, wyrażenie numer wzorzec identyfikator oraz wyrażenie sprawny. Formalny kontekst *K* stanowi — *reguły ekstrakcji informacji na temat punktów czerpania wody — Hydrantów*. Związek pomiędzy poszczególnymi ekstensjami i intensjami, tj. relacje między segmentami *s* a opisującymi je atrybutami *c*, zaprezentowano w tabeli 4.

Tabela 4

Tabela opisująca dane składające się na formalny kontekst *reguły ekstrakcji informacji na temat punktów czerpania wody — Hydrantów.* Źródło: opracowanie własne

Segment	Wyrażenie hydrant wzorzec ulica numer ulicy	Wyrażenie sprawny	Wzorzec typ hydrantu	Wyrażenie numer wzorzec identyfikator	Wyrażenie sprawny
s_1	1		1	1	
s_2	1		1	1	1
s_3	1	1			
s_4	1	1		1	
s_5	1		1		1
s_6	1		1		
s_7	1		1	1	

W tabeli 4 opisano, jakich reguł, zapisanych za pomocą atrybutów *c*, należy użyć, aby z segmentu *s* wyekstrahować potrzebną informację. Jeśli segment *s* zawiera którąś z reguł, wówczas ten fakt odnotowywany jest za pomocą wartości 1, w przeciwnym razie pole pozostaje puste. Na podstawie tak zdefiniowanych relacji można utworzyć tabelę ekstencji i intensji oraz kratę pojęć. Prowadzoną przez autora analizę dokumentacji przedstawiono w tabeli 5. Utworzoną kratę dla danych pokazano na rysunku 3.

W tabeli 5 zamieszczono wszelkie możliwe kombinacje, które występują w badanym zbiorze segmentów atrybutu *wyrażenie hydrant wzorzec ulica numer ulicy* z pozostałymi atrybutami. Widać, że istnieje 13 różnych połączeń atrybutu *wyrażenie hydrant wzorzec ulica numer ulicy* z pozostałymi atrybutami i dla każdego połączenia istnieje przynajmniej jeden segment opisany takim połączeniem atrybutów. Dysponując tabelą pojęć, można utworzyć kratę pojęć.

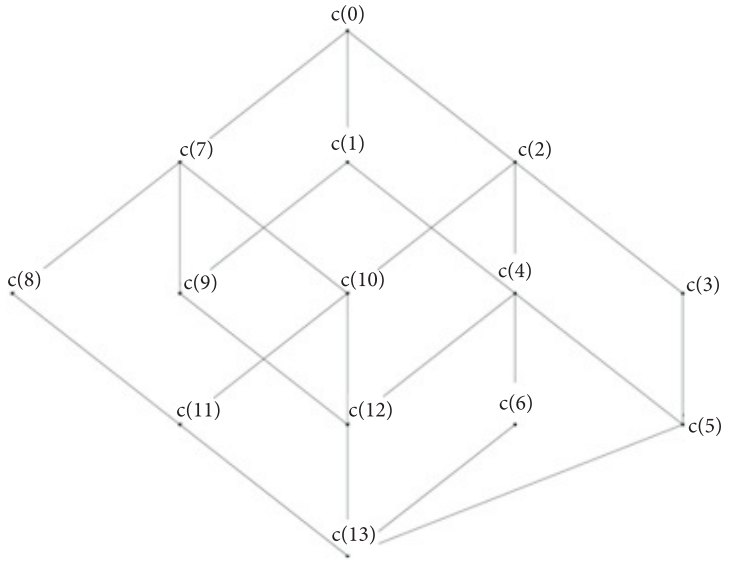
Diagram utworzonej kraty pojęcia *wyrażenie hydrant wzorzec ulica numer ulicy* prezentuje rysunek 3.

Tabela 5

Przykładowe pojęcia formalnego kontekstu *reguły ekstrakcji informacji na temat punktów czerpania wody — Hydrantów*. Źródło: opracowanie własne, przy wykorzystaniu [52]

Pojęcie	Intensje
c(0)	{wyrażenie hydrant wzorzec ulica numer ulicy}
c(1)	{wyrażenie hydrant wzorzec ulica numer ulicy; wyrażenie sprawny}
c(2)	{wyrażenie hydrant wzorzec ulica numer ulicy; wzorzec typ hydrantu}
c(3)	{wyrażenie hydrant wzorzec ulica numer ulicy; wzorzec typ hydrantu; wzorzec identyfikator}
c(4)	{wyrażenie hydrant wzorzec ulica numer ulicy; wzorzec typ hydrantu; wyrażenie sprawny}
c(5)	{wyrażenie hydrant wzorzec ulica numer ulicy; wzorzec typ hydrantu; wyrażenie sprawny; wzorzec identyfikator}
c(6)	{wyrażenie hydrant wzorzec ulica numer ulicy; wzorzec typ hydrantu; wyrażenie brak tabliczki; wyrażenie sprawny; wzorzec ulica numer ulicy}
c(7)	{wyrażenie hydrant wzorzec ulica numer ulicy; wyrażenie numer wzorzec identyfikator}
c(8)	{wyrażenie hydrant wzorzec ulica numer ulicy; wyrażenie numer wzorzec identyfikator; wzorzec opisu hydrantu}
c(9)	{wyrażenie hydrant wzorzec ulica numer ulicy; wyrażenie numer wzorzec identyfikator; wyrażenie sprawny}
c(10)	{wyrażenie hydrant wzorzec ulica numer ulicy; wyrażenie numer wzorzec identyfikator; wzorzec typ hydrantu}
c(11)	{wyrażenie hydrant wzorzec ulica numer ulicy; wyrażenie numer wzorzec identyfikator; wzorzec typ hydrantu; wzorzec opisu hydrantu}
c(12)	{wyrażenie hydrant wzorzec ulica numer ulicy; wyrażenie numer wzorzec identyfikator; wzorzec typ hydrantu; wyrażenie sprawny}

Po uzyskaniu tabeli opisującej pojęcia oraz diagramu w postaci kraty pojęć wyrażającej zachodzące relacje między obiektami i atrybutami, kolejnym krokiem w analizie i projektowaniu SEI jest odpowiednie złożenie reguł wyrażonych przez atrybuty do wykrywania odpowiednich schematów. Składanie atrybutów w schematy nie odbywa się *ad hoc*. Należy zauważyć, że wytworzona tabela pojęć oraz diagram pojęć daje opis tego, w jaki sposób tworzyć oprogramowanie i jak składać reguły ekstrakcji. Dysponując segmentami oraz kratą pojęć, którą pokazano na rysunku 3, można w łatwy sposób skonstruować oprogramowanie w postaci SEI, wykrywające odpowiednie schematy złożone z atrybutów dla analizowanych segmentów. Składanie



Rys. 3. Krata pojęcia *wyrażenie hydrant wzorzec ulica numer ulicy* formalnego kontekstu *reguły ekstrakcji informacji na temat punktów czerpania wody — Hydrantów*. Źródło: opracowanie własne, przy wykorzystaniu [52]

bazy wiedzy programu odbywa się tak, aby nie popełnić żadnego błędu związanego z pominięciem reguły i uzyskać pełne odwzorowanie polegające na wychwyceniu dla segmentu jak największej ilości atrybutów, które go opisują. Jeśli dany segment będzie analizowany według utworzonej kraty, to okaże się, że należy wędrować od dołu do góry kraty. Na dole kraty znajdują się obiekty opisane jak największą ilością pasujących do nich atrybutów. Podążając w górę kraty poprzez odpowiednie łuki między pojęciami, liczba tych atrybutów maleje. Tak więc oprogramowanie do ekstrakcji, wynikające z analizy diagramu, ma charakter stosu. Na początku stosu znajdują się najbardziej rozbudowane wzorce wychwytujące jak największą liczbę atrybutów, natomiast im bliżej końca stosu, wzorce te stają się coraz bardziej ubogie, tj. liczba dopasowywanych atrybutów maleje.

W celu zademonstrowania działania, przykładowo rozważono segment w postaci *hydrant hoża 30 numer 140 głęboki zasypany.* Segment ten zostanie dopasowany do pojęcia *c*(11), które zawiera oprócz swoich atrybutów atrybuty bezpośrednio dziedziczone z pojęć *c*(8) i *c*(10). Zaimplementowane oprogramowanie wykryje i stwierdzi, że badany segment zawiera *wyrażenie hydrant wzorzec ulica numer ulicy* i *wyrażenie numer wzorzec identyfikator* oraz *wzorzec typ hydrantu* i *wzorzec opisu hydrantu*. Po rozpoznaniu schematu zostaną następnie dla niego uruchomione reguły ekstrakcji i narzędzia mapowania wyekstrahowanych wartości do atrybutów wcześniej utworzonego modelu. W rozpatrywanym przypadku atrybut *nazwaUlicy* modelu *Lokalizacja* przyjmie wartość *hoża 30*, a atrybuty *identyfikator*,

rodzajHydrantu i *opisHydrantu* modelu *Hydrant* przyjmą kolejno wartości *140*, *głęboki* i *zasypany*. Drugi przykładowy segment w postaci *hydrant warszawska 15 podziemny 40382* zostanie dopasowany do pojęcia *c(3)*. Program wykryje schemat złożony z atrybutów *wyrażenie hydrant wzorzec ulica numer ulicy* i *wzorzec typ hydrantu* i *wzorzec identyfikator*, które znajdują się w tym pojęciu. Następnie, jak uprzednio, zostaną uruchomione reguły ekstrakcji dla tak wykrytego schematu. Zostaną wydobyte i zmapowane do odpowiednich pól modelu *Lokalizacja* oraz *Hydrant* wartości atrybutów *nazwaUlicy*, *rodzajHydrantu* i *identyfikator*.

3. Wyniki ekstrakcji informacji

Poniżej zaprezentowano i opisano podstawowe przykładowe statystyki, jakie mogą zostać otrzymane na podstawie danych pozyskiwanych z zaprojektowanego, zaimplementowanego i uzupełnionego rejestru przechowującego informacje o *punktach czerpania wody — Hydrantach*. Rejestr ten powstał z implementacji autorskiego procesu do budowy wybranego modelu oraz ekstrakcji do niego informacji z raportów opisujących interwencje PSP [1] za pomocą SEI, którego elementy zostały omówione w powyższych punktach artykułu.

Do skonstruowanego rejestru wyekstrahowano informacje na temat 1416 hydrantów. Dane na temat sprawności hydrantów zaprezentowano w tabeli 6.

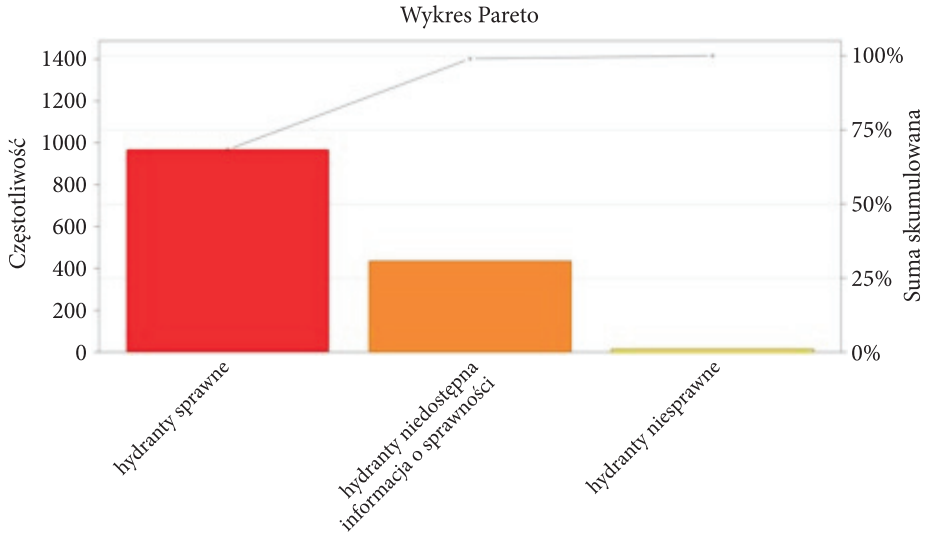
Tabela 6

Dane na temat sprawności hydrantów uzyskanych z badanych raportów tekstowych. Źródło: opracowanie własne

Hydranty	Częstotliwość	Skumulowana częstotliwość	Procent [%]	Skumulowany procent [%]
Sprawne	967	967	68,29	68,29
Niedostępna informacja o sprawności	435	1402	30,72	99,01
Niesprawne	14	1416	0,99	100

Na podstawie danych, które przedstawiono w tabeli 6, został wykonany wykres Pareto. Wykres ten przedstawiono na rysunku 4.

Na wykresie zaprezentowano informację o sprawności hydrantów, uzyskaną z procesu ekstrakcji raportów tekstowych, a dokładniej z ekstrakcji sekcji sprzęt. Widać, że w około 68% badanych przypadków KDR odnotowali fakt, że hydrant był sprawny. Liczba niesprawnych hydrantów jest bliska 1%. W przypadku 31% przeanalizowanych raportów KDR nie umieścili żadnej informacji o stanie danego hydrantu, tj. nie wiadomo, czy dany obiekt był lub nie był sprawny. Dodatkowo



Rys. 4. Wykres Pareto danych na temat sprawności hydrantów uzyskanych z badanych raportów tekstowych. Źródło: opracowanie własne, wykonane przy wykorzystaniu pakietu [21]

należy wspomnieć, że w badanym zbiorze najczęściej pojawiającymi się wyrażeniami opisującymi stan hydrantu, a także przyczyny jego niesprawności, były takie słowa, jak: *zasypany*, *zastawiony*, *zamarznięta pokrywa*, *urwany czop*, *uszkodzona pokrywa*, *zasypany śniegiem*, *zbyt głęboko osadzony*, *uszkodzony*. Opisy te nie wskazują jednak jednoznacznie na to, czy hydrant był lub nie był sprawny, np. brak czopa nie sprawia, że z hydrantu nie można skorzystać, niemniej już zasypanie go przez śnieg lub zamarznięcie powoduje niemożność korzystania z niego podczas interwencji. Z tych względów wyżej zaprezentowaną statystykę sprawności oparto o informację o tym, że dany hydrant działał lub nie działał, a więc w raporcie jawnie wystąpiło wyrażenie *działa* lub *nie działa*. Fakt ten mógł zostać odnotowany na podstawie przeprowadzenia próby działania hydrantu przez KDR, tj. jego odkręcenie i sprawdzenie ciśnienia wody. Do budowania powyższej statystyki nie brano też pod uwagę takich sformułowań, jak *tankowano z hydrantu numer…*, które jednoznacznie wskazują na to, że hydrant był sprawny. Z powyżej przedstawionych względów oszacowywana liczba hydrantów sprawnych może być większa poprzez zmniejszenie grupy hydrantów, w której brak było jawnego wyrażenia faktu o sprawności bądź niesprawności danych punktów czerpania wody.

Na podstawie zebranych danych można wyznaczyć także, ile i jaki procent stanowią hydranty oznakowane w dostępnym zbiorze informacji o hydrantach. Dane na temat oznakowania hydrantów zaprezentowano w tabeli 7.

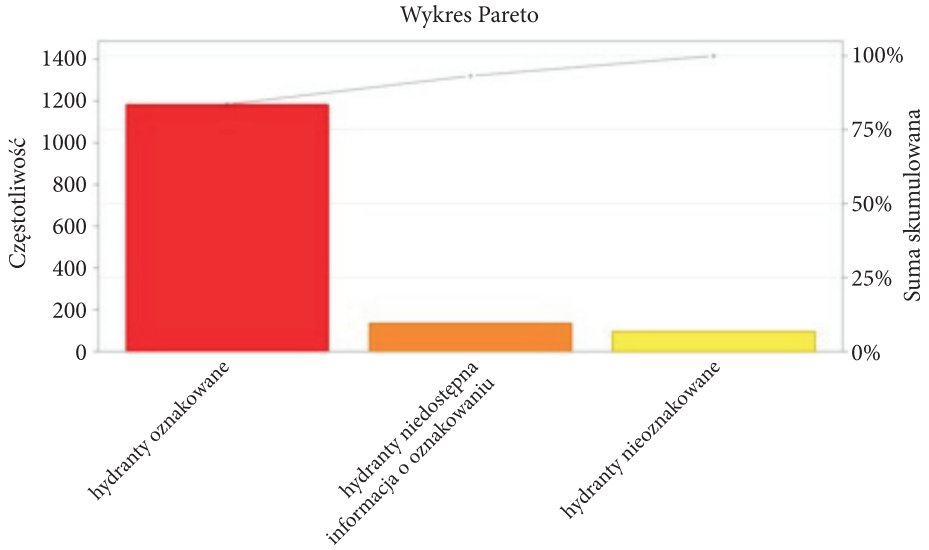
Na podstawie danych zaprezentowanych w tabeli 7 wykonano wykres Pareto. Przedstawiono go na rysunku 5. Zawiera on informację o oznakowaniu hydrantów, która została uzyskana z procesu ekstrakcji raportów tekstowych, a dokładniej

z ekstrakcji sekcji sprzęt. Widać, że w około 83,5% badanych przypadków KDR odnotowali fakt, że hydrant był oznakowany, tj. posiadał tabliczkę znamionową oraz można było odczytać numer identyfikacyjny hydrantu. Liczba nieoznakowanych hydrantów jest bliska 7% i wynika z braku tabliczki lub jej nieczytelności, co powodowało niemożność odczytania numeru identyfikacyjnego hydrantu. W przypadku 9,5% przeanalizowanych raportów KDR nie umieścili żadnej informacji o tym, czy tabliczka jest lub nie jest dostępna, tj. nie wiadomo, czy dany obiekt posiadał bądź też nie posiadał numeru identyfikacyjnego.

Tabela 7

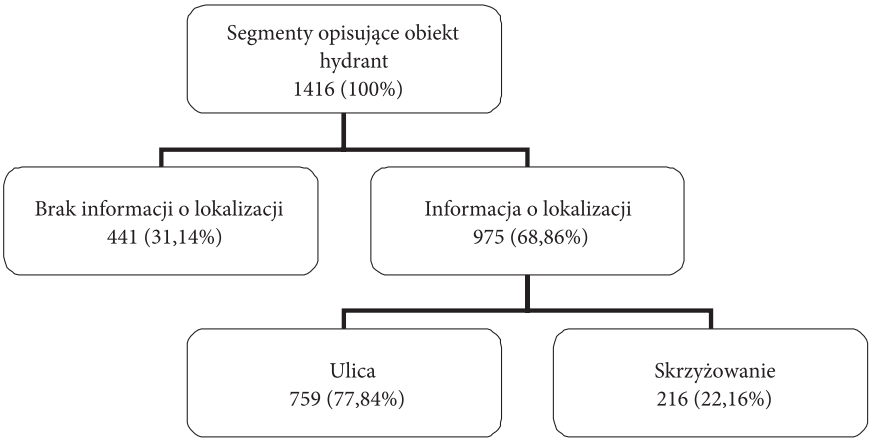
Dane na temat oznakowania hydrantów uzyskanych z badanych raportów tekstowych. Źródło: opracowanie własne

Hydranty	Częstotliwość	Skumulowana częstotliwość	Procent [%]	Skumulowany procent [%]
Oznakowane	1184	1184	83,62	83,62
Niedostępna informacja o oznakowaniu	136	1320	9,60	93,22
Nieoznakowane	96	1416	6,78	100



Rys. 5. Wykres Pareto danych na temat oznakowania hydrantów uzyskanych z badanych raportów tekstowych. Źródło: opracowanie własne, wykonane przy wykorzystaniu pakietu [21]

W skonstruowanym i zaimplementowanym rejestrze *punktów czerpania wody — Hydranty* w procesie ekstrakcji została pozyskana informacja o ich względnej lokalizacji. Podstawowe statystyki dotyczące lokalizacji pokazano na rysunku 6.



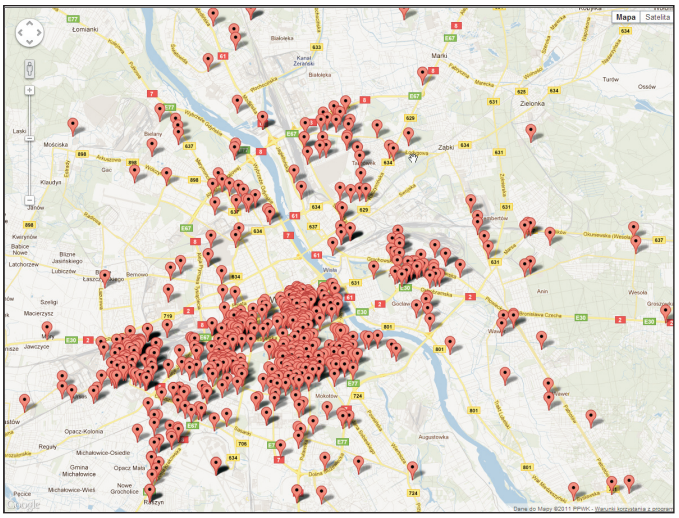
Rys. 6. Opis statystyk dotyczących lokalizacji *punktów czerpania wody — Hydrantów*. Źródło: opracowanie własne

Na wykresie w postaci drzewa (rys. 6) zaprezentowano informację o lokalizacji względnej hydrantów, która została uzyskana z procesu ekstrakcji raportów tekstowych. Widać, że w 31% analizowanych przypadków KDR nie uwzględnili informacji o położeniu hydrantów. W około 69% zbadanych przypadków KDR odnotowali fakt, że hydranty znajdują się przy ulicy lub na ich skrzyżowaniu, tj. posiadają lokalizację. Informacja o tym, że hydranty znajdują się przy ulicy, stanowi 77,8% opisów lokalizacji. Natomiast 22,2% raportów zawierających informację o lokalizacji informuje o tym, że jest ona związana ze skrzyżowaniem ulic.

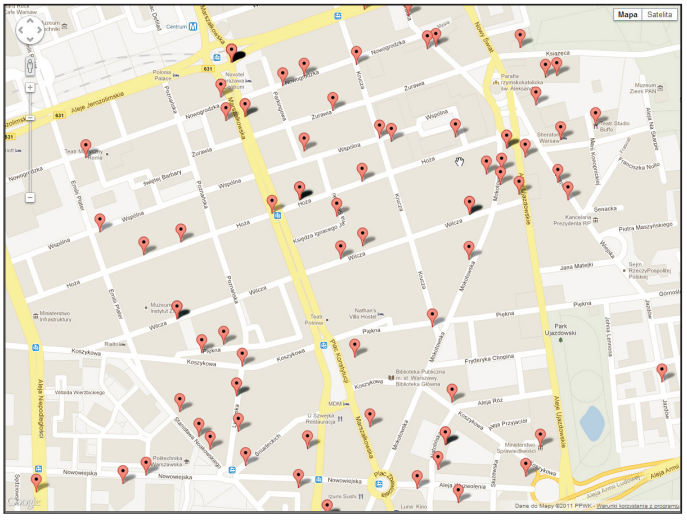
Dysponując informacją o lokalizacji hydrantu, można na jej podstawie w procesie geokodowania [53-58] ustalić względną szerokość oraz długość geograficzną danego hydrantu. Istnieje więc możliwość wizualizacji położenia zarejestrowanych hydrantów dostępnych w zbudowanym rejestrze. Przykładową wizualizację zaprezentowano kolejno na rysunkach 7 i 8.

Na rysunku 7 pokazano otrzymaną wizualizację, w dużej skali, położenia hydrantów z procesu ich geokodowania na podstawie informacji o ich lokalizacji zebranych w utworzonym rejestrze *Hydrantów*. Na mapie markerami zaznaczono położenie hydrantu na podstawie pozyskanej jego szerokości i długości geograficznej. Na rysunku widać, że większość punktów czerpania wody została sprawdzona w południowo-zachodniej części Warszawy, tj. dzielnice Mokotów, Ochota, Ursynów i Włochy. Sporadycznie sprawdzone zostały hydranty w rejonach Pragi Południe, Pragi Północ i Żoliborza. Widać również, że jednostki brały udział w interwencjach odbywających się także w obrębie miasta Warszawa w takich miejscowościach jak Raszyn czy Kabaty.

Na rysunku 8 widać otrzymaną wizualizację, w mniejszej skali (w przybliżeniu), położenia hydrantów z procesu ich geokodowania na podstawie informacji o ich



Rys. 7. Przykładowa wizualizacja położenia hydrantów w zaprojektowanym, zaimplementowanym oraz uzupełnionym w procesie ekstrakcji informacji rejestrze *punktów czerpania wody — Hydranty*. Źródło: opracowanie własne, wykonane przy wykorzystaniu pakietu [53-55]



Rys. 8. Przykładowa wizualizacja w przybliżeniu położenia hydrantów w zaprojektowanym, zaimplementowanym oraz uzupełnionym w procesie ekstrakcji informacji rejestrze *punktów czerpania wody — Hydranty*. Źródło: opracowanie własne, wykonane przy wykorzystaniu pakietu [53-55]

lokalizacji, zebranych w utworzonym rejestrze *Hydrantów*. Aktualnie prezentowany wycinek mapy przedstawia rejony Śródmieście-Mokotów. Widać na nim, że w tym rejonie sprawdzono około 80 obiektów czerpania wody w postaci hydrantów.

4. Wnioski i prace rozwojowe

W opisie metody projektowania bazy wiedzy SEI autor przedstawił, w jaki sposób komponować ze sobą poszczególne atrybuty w celu wytworzenia z nich bardziej rozbudowanych schematów pasujących do segmentów. Przyjęty sposób projektowania bazy wiedzy SEI można wyrazić jako projekt od szczegółu do ogółu. Zgodnie z przyjętą zasadą najpierw podejmowana jest próba dopasowania do pojawiającego się segmentu jak największej liczby atrybutów, po czym, jeśli to się nie udaje, następuje ich ograniczanie. Podczas analizy segmentów opisujących *punkty czerpania wody — Hydranty* mogą się w nich znaleźć nieprzydatne kombinacje atrybutów. Takimi kombinacjami atrybutów są np. *{wzorzec typ hydrantu; wzorzec opisu hydrantu; wyrażenie niesprawny}* czy też *{wyrażenie brak tabliczki; wyrażenie sprawny}*. Atrybuty te, ich kombinacje, nie wnoszą informacji o ewentualnym położeniu czy identyfikatorze obiektu, którego dotyczą, stąd mogą być mało przydatne. Niemniej w rozpatrywanej sytuacji dla autora nie miało to większego znaczenia. Należy to traktować jako uwagę autorską do opracowanego modelu i procesu ekstrakcji informacji w ewentualnym jego wykorzystaniu.

Wyżej przedstawiono i opisano tylko podstawowe możliwości, jakie płyną z utworzonego rejestru i metody jego projektowania. W dalszej kolejności, a w szczególności jeśli chodzi o warstwę wizualizacji lokalizacji hydrantów, można tworzyć maski (widoki) graficzne w celu wsparcia analityka *punktów czerpania wody — Hydrantów*. W ten sposób można wytworzyć różne warstwy wizualne do prezentacji uszkodzeń, częstotliwości sprawdzania hydrantów czy też mapy działających bądź niedziałających hydrantów.

W swych dotychczasowych pracach autor zweryfikował poprzez eksperymenty te aspekty, które leżały w zakresie jego badań. Po pierwsze autor w swych aktualnych badaniach wykazał, że algorytmy mogą dokonywać klasyfikacji segmentów z zadawalającym wynikiem obok ludzi-ekspertów [59]. System klasyfikacji dobrze symuluje działanie eksperta, osoby dokonującej klasyfikacji, a więc działa poprawnie. Tak samo dzieje się z systemem do segmentacji tekstu [60, 61]. Aktualnie opisany w artykule SEI do utworzonego modelu także dobrze radzi sobie z powierzonym zadaniem. Niemniej komponent ten jak i cały SI w postaci operacyjnej bazy danych na temat *punktów czerpania wody — Hydrantów*, który na nim bazuje, wymaga dalszej weryfikacji i oceny. Ocena całościowa zaprojektowanego i zaimplementowanego SI polegająca na jego bezpośrednim zastosowaniu podczas akcji ratowniczo-gaśniczych i ocena znajdujących się w nim informacji leżała poza aktualnymi badaniami autora. Niemniej aspekty te obejmują prace rozwojowe, które bazują na zastosowaniu crowdsourcingu dla PSP do weryfikacji i oceny SEI [62].

Artykuł wpłynął do redakcji 3.01.2012 r. Zweryfikowaną wersję po recenzji otrzymano w maju 2012 r.

LITERATURA

[1] M. Mirończuk, T. Maciak, *Proces i metody eksploracji danych tekstowych do przetwarzania raportów z akcji ratowniczo-gaśniczych*, Metody Informatyki Stosowanej, 4, 2011.

[2] M. Mirończuk, *Przegląd metod i technik eksploracji danych tekstowych*, Studia i Materiały Informatyki Stosowanej SIMIS, 2, 2011 (w cyklu recenzyjnym).

[3] M. Mirończuk, *Zmodyfikowana analiza FMEA z elementami SFTA w projektowaniu systemu wyszukiwania informacji na temat obiektów hydrotechnicznych w nierelacyjnym katalogowym rejestrze*, Studia Informatica. Gliwice: Wydawnictwo Politechniki Śląskiej, 2, 2011.

[4] Rozporządzenie Ministra Spraw Wewnętrznych i Administracji z dnia 29 grudnia 1999 r. w sprawie szczegółowych zasad organizacji krajowego systemu ratowniczo-gaśniczego, Dz.U.99.111.1311 § 34 pkt. 5 i 6.

[5] Abakus: System EWID99 (on-line, dostęp od 1 maja 2009 w Internecie: http://www.ewid.pl/?set=rozw_ewid&gr=roz).

[6] Abakus: System EWIDSTAT (on-line, dostęp od 1 maja 2009 w Internecie: http://www.ewid.pl/?set=ewidstat&gr=prod).

[7] Strona firmy abakus (on-line, dostęp od 1 marca 2009 w Internecie: http://www.ewid.pl/?set=main&gr=aba).

[8] A. Krasuski, K. Kreńsk, *Ewid 9x i co dalej?*, Przegląd Pożarniczy, 6, 2006.

[9] R. Simiński, S. Janus, *Wizualizacja wnioskowania w regułowych bazach wiedzy z wykorzystaniem sieci Petriego*, Studia Informatica, Wydawnictwo Politechniki Śląskiej, Gliwice, 2011.

[10] M.F. Moens, *Information Extraction: Algorithms and Prospects in a Retrieval Context*, The Information Retrieval Series, Springer, 2006.

[11] D.M. Bikel, R. Schwartz, R.M. Weischedel, *An Algorithm that Learns What's in a Name*, Machne Learning, 1999, 211-231.

[12] A. Mykowiecka, *Inżynieria lingwistyczna*, Komputerowe przetwarzanie tekstów w języku naturalnym, PJWSTK, Warszawa, 2007.

[13] M. Mirończuk, T. Maciak, *Propozycja mieszanego przetwarzania pół-strukturalnego modelu opisu zdarzeń z akcji ratowniczo gaśniczych Państwowej Straży Pożarnej PSP*, CNBOP „Bezpieczeństwo i Technika Pożarnicza" (w cyklu recenzyjnym).

[14] *The R Project for Statistical Computing* (dostęp od 1 stycznia 2011 w Internecie: http://www.r-project.org).

[15] A. Krasuski, *Rozproszona baza danych — możliwości wykorzystania w PSP*, Przegląd Pożarniczy, 5, 2006, 30-33.

[16] A. Krasuski, *Usługi katalogowe w architekturze rozproszonej bazy danych w procesie wspomagania decyzji w Państwowej Straży Pożarnej*, Seminarium Zakładu Logiki Matematycznej, 2009.

[17] A. Krasuski, *Możliwość wykorzystania usługi katalogowej w architekturze rozproszonej bazy danych jako podstawy systemu treningu i wspomagania decyzji w Państwowej Straży Pożarnej*, Politechnika Białostocka, Wydział Informatyki, Białystok, 2010.

[18] A. Krasuski, T. Maciak, *Rozproszone bazy danych w Państwowej Straży Pożarnej — model systemu*, Bazy danych, technologie, narzędzia, WKŁ, Warszawa, 2005, 135-142.

[19] A. Krasuski, T. Maciak, *Wykorzystanie rozproszonej bazy danych oraz wnioskowania na podstawie przypadków w procesach decyzyjnych Państwowej Straży Pożarnej*, Zeszyty Naukowe SGSP, 36, 2008, 17-35.

[20] A. Krasuski, T. Maciak, *Rozproszone bazy danych — architektura funkcjonalna*, Bezpieczeństwo i Technika Pożarnicza, 01, 2007.

[21] L. Scrucca, *qcc: an R package for quality control charting and statistical process control*, R. News, 4/1, 2004, 11-17.

[22] K.E. Wolff, *A first course in formal concept analysis*, 1994 (dostęp od 22 grudnia 2009 w Internecie: http://www.fbmn.fh-darmstadt.de/home/wolff/Publikationen/A_First_Course_in_Formal_Concept_Analysis.pdf).

[23] P. Patil, *Applying Formal Concept Analysis to Object Oriented Design and Refactoring*, Bombay: Department Of Computer Science and Engineering Indian Institute Of Technology, 2009.

[24] U. Priss, *Formal concept analysis in information science*, Annual Review of Information Science and Technology, 40, 2006, 521-543.

[25] A. Walendziak, *Podstawy algebry ogólnej i teorii krat*, Wydawnictwo Naukowe PWN, Warszawa, 2009.

[26] S.H. Hwang, H.G. Kim, H.S. Yang, *A FCA-Based Ontology Construction for the Design of Class Hierarchy*, Computational Science and Its Applications — ICCSA 2005, Springer Berlin/Heidelberg, 2005, 307-320.

[27] C. Carpineto, G. Romano, *Using Concept Lattices for Text Retrieval and Mining*, Formal Concept Analysis, Springer Berlin/Heidelberg, 2005, 3-45.

[28] P. Cimiano, A. Hotho, S. Staab, *Clustering concept hierarchies from text*, In Proceedings of LREC, 2004.

[29] Leksyka.pl Knowledge-based system (dostęp: od 5 maja 2011 w Internecie: http://megaslownik.pl/slownik/angielsko_polski/137416,knowledge-based+system).

[30] K. Mens, T. Tourw, *Delving source code with formal concept analysis*, Computer Languages, Systems and Structures, 31, 2005, 183-197.

[31] W. Muangon, S. Intakosum, *Retrieving design patterns by case-based reasoning and Formal Concept Analysis*, Computer Science and Information Technology, IEEE International Conference, Beijing, 2009.

[32] W. Muangon, S. Intakosum, *Adaptation of Design Pattern Retrieval Using CBR and FCA*, Proceedings of the 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, 2009.

[33] G. Arvalo, T. Mens, *Analysing Object-Oriented Application Frameworks Using Concept Analysis*, Proceedings of the Workshops on Advances in Object-Oriented Information Systems, 2002.

[34] M. Felleisen, *How to design class hierarchies*, Proceedings of the 2005 workshop on Functional and declarative programming in education, Tallinn, Estonia, 2005.

[35] V.K. Proulx, K.E. Gray, *Design of class hierarchies: an introduction to OO program design*, Proceedings of the 37 th SIGCSE Technical Symposium on Computer Science Education, 38, 2006, 288-292.

[36] R. Godin, H. Mili, G.W. Mineau, R. Missaoui, A. Arfi, T.T. Chau, *Design of class hierarchies based on concept (Galois) lattices*, Theory and Practice of Object Systems — Special issue high availability in CORBA, 4, 1998, 117-133.

[37] R. Godin, P. Valtchev, *Formal Concept Analysis-Based Class Hierarchy Design in Object-Oriented Software Development*, Formal Concept Analysis: Springer Berlin/Heidelberg, 2005, 209-231.

[38] G. Snelting, F. Tip, *Reengineering class hierarchies using concept analysis*, SIGSOFT Software Engineering Notes, 23, 1998, 99-110.

[39] G. Snelting, F. Tip, *Understanding class hierarchies using concept analysis*, ACM Transactions on Programming Languages and Systems (TOPLAS), 22, 2000, 540-582.

[40] P. Tonella, *Formal Concept Analysis in Software Engineering*, Proceedings of the 26th International Conference on Software Engineering, 2004.

[41] A. Laukaitis, O. Vasilecas, *Formal concept analysis and information systems modeling*, Proceedings of the 2007 International Conference on Computer Systems and Technologies, Bulgaria, 2007.

[42] W. Hesse, T. Tilley, *Formal Concept Analysis Used for Software Analysis and Modelling*, Formal Concept Analysis: Springer Berlin/Heidelberg, 2005, 259-282.

[43] B. Díaz-Agudo, P.A. González-Calero, *Formal concept analysis as a support technique for CBR*, Knowledge-Based Systems, 14, 2001, 163-171.

[44] D. Belén, A.G. Marco, P.G. Pedro, A.G. Pedro, *Dep Sistemas I. Formal concept analysis for knowledge refinement in case based reasoning*, Springer, 2005.

[45] P. Pattaraintakorn, V. Boonjing, J. Tadrat, *A New Case-Based Classifier System Using Rough Formal Concept Analysis*, Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology, 02, 2008.

[46] Y. Li, S.C.K. Shiu, S.K. Pal, *Combining Feature Reduction and Case Selection in Building CBR Classifiers*, IEEE Transactions on Knowledge and Data Engineering, 18, 2006, 415-429.

[47] K.T.J. Rancz, V. Varga, *A method for mining functional dependencies in relational database design using FCA*, Studia Universitatis „Babes-Bolyai" Cluj-Napoca, Informatica, 53, 2008, 17-28.

[48] H. Haav, *A semi-automatic method to ontology design by using FCA*, University of Ostrava, Department of Computer Science, Ostrava, 2004.

[49] W. Gliński, *Ontologie. Próba uporządkowania terminologicznego chaosu*, Instytut Informacji Naukowej i Studiów Bibliologicznych UW (dostęp od 10 sierpnia 2010 w Internecie: http://bbc.uw.edu.pl/Content/20/13.pdf).

[50] W. Hesse, *Ontologies in the Software Engineering process*, EAI 2005 — Proceedings of the Workshop on Enterprise Application Integration, 2005.

[51] J. Poelmans, P. Elzinga, S. Viaene, G. Dedene, *Formal concept analysis in knowledge discovery: a survey*, Proceedings of the 18th international conference on Conceptual structures: from information to intelligence, Kuching, Sarawak, Malaysia, 2010.

[52] M. Radvansky, *Formal concept analyse* (dostępny od 1 maja 2011 w Internecie: http://www.fca.radvansky.net/news.php).

[53] Google Maps API Family (dostępny od 20 września 2011 w Internecie: http://code.google.com/intl/pl-PL/apis/maps/index.html).

[54] M.R. Loecher, *GoogleMaps: Overlays on Google map tiles in R. 2011* (dostępny od 1 listopada 2011 w Internecie: http://CRAN.R-project.org/package=RgoogleMaps).

[55] M. Loecher, S. Networks, *Plotting on Google Static Maps in R. 2010* (dostępny od 1 listopad 2011 w Internecie: http://cran.r-project.org/web/packages/RgoogleMaps/vignettes/RgoogleMaps-intro.pdf).

[56] A. Kotulla, *Wyszukiwanie informacji z uwzględnieniem danych dotyczących lokalizacji*, Studia Informatica, Wydawnictwo Politechniki Śląskiej, Gliwice, 2011, 73-84.

[57] P. Christen, A. Willmore, T. Churches, *A Probabilistic Geocoding System Utilising a Parcel Based Address File Data Mining*, Springer Berlin/Heidelberg, 2006, 130-145.

[58] P. Christen, T. Churches, A. Willmore, *A Probabilistic Geocoding System based on a National Address File*, Proceedings of the 3rd Australasian Data Mining Conference, 2004.

[59] M. Mirończuk, *Structuring text documents from Fire Service of Poland using selected classifiers*, Fire Technology, 2012 (w cyklu recenzyjnym).

[60] M. Mirończuk, *Metoda projektowania bazy wiedzy oraz reguł segmentatora regułowego oparta o formalną analizę pojęć*, CNBOP „Bezpieczeństwo i Technika Pożarnicza" (w cyklu recenzyjnym).

[61] M. Mirończuk, T. Maciak, *System informacyjny na temat sieci hydrantów dla krajowego systemu ratowniczo-gaśniczego: metoda segmentacji tekstu i jej ocena*, Metody Informatyki Stosowanej, 1, 2012.

[62] M. Mirończuk, *Crowdsourcing in rescue fire service — proposed application*, Studia i Materiały Informatyki Stosowanej SIMIS, 4, 2011 (w cyklu recenzyjnym).

M.M. MIROŃCZUK

Application of formal concept analysis for information extraction system analysis

Abstract. This article describes a design process of information extraction system IES. The proposed projecting method is based on rules and formal concept analysis.

Keywords: formal concept analysis, information extraction, project of knowledge based systems