 Please cite as: **BiTP Vol. 34 Issue 2**, 2014, pp. 93-103

DOI:10.12845/bitp.34.2.2014.9

dr inż. **Marcin Michał MIROŃCZUK**[1]

Przyjęty/Accepted/Принята: 05.08.2013;
Zrecenzowany/Reviewed/Рецензирована: 22.05.2014;
Opublikowany/Published/Опубликована: 30.06.2014;

METODA PROJEKTOWANIA BAZY WIEDZY ORAZ REGUŁ SEGMENTATORA REGUŁOWEGO OPARTA O FORMALNĄ ANALIZĘ POJĘĆ*

The Method of Designing the Knowledge Database and Rules for a Text Segmentation Tool Based on Formal Concept Analysis

Метод проектирования базы знаний и правил правового сегментатора на основе формального анализа понятий

Abstrakt
Cel: Zaprezentowanie rozwiązania problemu segmentacji tekstu dziedzinowego. Badany tekst pochodził z raportów (formularza „Informacji ze zdarzenia", pola „Dane opisowe do informacji ze zdarzenia") sporządzanych po akcjach ratowniczo-gaśniczych przez jednostki Państwowej Straży Pożarnej.
Metody: W celu realizacji zadania autor zaproponował metodę projektowania bazy wiedzy oraz reguł segmentatora regułowego. Zaproponowana w artykule metoda opiera się na formalnej analizie pojęć. Zaprojektowana według proponowanej metody baza wiedzy oraz reguł umożliwiła przeprowadzenie procesu segmentacji dostępnej dokumentacji. Poprawność i skuteczność proponowanej metody zweryfikowano poprzez porównanie jej wyników z dwoma innymi rozwiązaniami wykorzystywanymi do segmentacji tekstu.
Wyniki: W ramach badań i analiz opisano oraz pogrupowano reguły i skróty występujące w badanych raportach. Dzięki zastosowaniu formalnej analizy pojęć utworzono hierarchię wykrytych reguł oraz skrótów. Wydobyta hierarchia stanowiła zarazem bazę wiedzy oraz reguł segmentatora regułowego. Przeprowadzone eksperymenty numeryczne i porównawcze autorskiego rozwiązania z dwoma innymi rozwiązaniami wykazały znacznie lepsze działanie tego pierwszego. Przykładowo otrzymane wyniki F-miary otrzymane w wyniku zastosowania proponowanej metody wynoszą 95,5% i są lepsze o 7-8% od pozostałych dwóch rozwiązań.
Wnioski: Zaproponowana metoda projektowania bazy wiedzy oraz reguł segmentatora regułowego umożliwia projektowanie i implementację oprogramowania do segmentacji tekstu z małym błędem podziału tekstu na segmenty. Podstawowa reguła dotycząca wykrywania końca zdania poprzez interpretację kropki i dodatkowych znaków jako końca segmentu w rzeczywistości, zwłaszcza dla tekstów specjalistycznych, musi być opakowana dodatkowymi regułami. Działania te znacznie podnoszą jakość segmentacji i zmniejszają jej błąd. Do budowy i reprezentacji takich reguł nadaje się przedstawiona w artykule formalna analiza pojęć. Wiedza inżyniera oraz dodatkowe eksperymenty mogą wzbogacać utworzoną sieć o nowe reguły. Nowo wprowadzana wiedza może zostać w łatwy sposób naniesiona na aktualnie utworzoną sieć semantyczną, tym samym przyczyniając się do polepszenia segmentacji tekstu. Ponadto w ramach eksperymentu numerycznego wytworzono unikalny: zbiór reguł oraz skrótów stosowanych w raportach, jak również zbiór prawidłowo wydzielonych i oznakowanych segmentów.

Słowa kluczowe: formalna analiza pojęć, segmentator tekstu, segmentator regułowy, projektowanie bazy wiedzy, metoda projektowania bazy wiedzy, FCA, wydzielanie segmentów, dzielenie tekstu na segmenty
Typ artykułu: oryginalny artykuł naukowy

Abstract
Objective: Presentation of a specialist text segmentation technique. The text was derived from reports (a form "Information about the event", field "Information about the event - descriptive data") prepared by rescue units of the State Fire Service after firefighting and rescue operations.

[1] Instytut Podstaw Informatyki PAN, Zespół Podstaw Sztucznej Inteligencji, ul. Jana Kazimierza 5, 01-248 Warszawa/Institute of Computer Science of the Polish Academy of Sciences, Poland; e-mail: m.marcinmichal@gmail.com
* Artykuł został wyróżniony przez Komitet Redakcyjny / The article was recognised by the Editorial Committee/ Эту статью наградил Редакционный Совет

DOI:10.12845/bitp.34.2.2014.9

Methods: In order to perform the task the author has proposed a method of designing the knowledge base and rules for a text segmentation tool. The proposed method is based on formal concept analysis (FCA). The knowledge base and rules designed by the proposed method allow performing the segmentation process of the available documentation. The correctness and effectiveness of the proposed method was verified by comparing its results with the other two solutions used for text segmentation.
Results: During the research and analysis rules and abbreviations that were present in the studied specialist texts were grouped and described. Thanks to the formal concepts analysis a hierarchy of detected rules and abbreviations was created. The extracted hierarchy constituted both a knowledge and rules base of tools for segmentation of the text. Numerical and comparative experiments on the author's solution with two other methods showed significantly better performance of the former. For example, the F-measure results obtained from the proposed method are 95.5% and are 7-8% better than the other two solutions.
Conclusions: The proposed method of design knowledge and rules base text segmentation tool enables the design and implementation of software with a small error divide the text into segments. The basic rule to detect the end of a sentence by the interpretation of the dots and additional characters as the end of the segment, in fact, especially in case of specialist texts, must be packaged with additional rules. These actions will significantly improve the quality of segmentation and reduce the error. For the construction and representation of such rules is suitable presented in the article, the formal concepts analysis. Knowledge engineering and additional experiments can enrich the created hierarchy by the new rules. The newly inserted knowledge can be easily applied to the currently established hierarchy thereby contributing to improving the segmentation of the text. Moreover, within the numerical experiment is made unique: a set of rules and abbreviations used in reports and set properly separated and labeled segments.

Keywords: formal concept analysis, FCA, project of knowledge database, segment extraction, text processing
Type of article: original scientific article

Цель: Представление решения проблемы сегментации специализированного текста. Исследованный текст исходил из отчётов (формуляра «Информации из места события", поля „Описывающие данные к информации о происшествии) составленных после спасательно-гасящих действиях подразделений Государственной Пожарной Службы.
Методы: Имея ввиду реализацию задачи автор предложил метод проектирования базы знаний и правил правового сегментатора. Предлагаемый в статье метод основан на формальном анализе понятий. Разработанная в соответствии предложенному методу база знаний и правил даёт возможность проведения процесса сегментации имеющейся документации. Правильность и эффективность предложенного метода проверены путём сравнения его результатов с двумя другими решениями использованными для сегментации текста.
Результаты: В рамках исследований и анализа описаны и погруппированы правила и сокращения появляющиеся в исследуемых отчётах. Благодаря применению формального анализа понятий создана иерархия обнаруженных правил и сокращений. Извлечённая иерархия представляет собой одновременно базу знаний и правил правового сегментатора. Проведены цифровые и сравнительные эксперименты авторского решения с двумя другими методами показали значительно лучшую производительность первого. Например результаты F-меры полученные в результате применения предлагаемого метода составляют 95,5% и являются на 7-8% лучшими от двух остальных решений.
Выводы: Предложенный метод проектирования базы знаний и правил правового сегментатора дает возможность проектировать и внедрять программное обеспечение для сегментации текста с небольшими ошибками разделения текста на сегменты. Основное правило по обнаружению конца предложения – наличие точки и дополнительных символов в качестве конца сегмента, на самом деле, особенно при сегментации специализированных текстов, должно быть оснащено дополнительными правилами. Эти действия значительно повышают качество сегментации и уменьшают её ошибочность. Для постройки и представления таких правил подходит представленный в статье формальный анализ понятий. Инженерные знания и дополнительные эксперименты могут обогатить создаваемую сеть новыми правилами. Нововведённые знания простым образом могут быть нанесены на только что разработанную семантическую сеть, тем самым, совершенствуя процесс сегментации текста. Кроме того, в рамках цифрового эксперимента созданы уникальные: набор правил и сокращений используемых в отчётах, а также набор правильно выделенных и означенных сегментов.

Ключевые слова: формальный анализ понятий, сегментатор текста, правовой сегментатор, проектирование базы знаний, FCA, выделение сегментов, разделение текста на сегменты
Вид статьи: оригинальная научная статья

1. Wstęp

W artykułach [1, 2] przedstawiono model obsługi akcji ratowniczo-gaśniczej wspierany przez hybrydowy system wspomagania decyzji HSWD. Proponowany HSWD dla Państwowej Straży Pożarnej stanowi połączenie grupowego systemu wspomagania decyzji GSWD (*ang. group decision support system – GDSS*) trzeciego rodzaju i inteligentnego systemu wspomagania decyzji bazującego na odkrywaniu wiedzy ISWD (*ang. intelligent decision support system based on knowledge discovery – IDSSKD*) [2]. Pierwszy wymieniony system stanowił platformę informatyczną do podsuwania pomysłów i instruowania osoby podejmującej decyzję na zasadzie konsultacji eksperckiej bazującej na zgromadzonych w nim informacjach i regułach. Druga platforma – IDSSKD – opierała się z kolei na włączeniu do systemu wspomagania decyzji SWD elementów odkrywania wiedzy w bazach danych (*ang. knowledge discovery in database – KDD*) z danych tekstowych. Całość tego rozwiązania z punktu widzenia technicznego bazowała i wspierała proces wnioskowania na podstawie przypadków zdarzeń (*ang. cased based resoning – CBR*) [3].

Autor dla wyżej wymienionego systemu podjął się opracowania (zaprojektowania i zaimplementowania) warstwy danych dotyczącej reprezentowania i przechowywania informacji na temat sieci hydrantów [4]. Dzię-

ki zebranym informacjom w tej warstwie danych kierujący działaniami ratowniczymi KDR mogliby lokalizować najbliższe punkty czerpania środka gaśniczego. Projekt tej warstwy został oparty na opracowanej przez autora metodzie eksploracyjnej analizy tekstu do jego strukturalizacji [5-7]. Analizowane teksty stanowiły dokumenty z prowadzonych działań ratowniczo-gaśniczych przez jednostki PSP, pochodzące z systemu ewidencji zdarzeń EWID [8-10]. Podczas komputerowej analizy dokumentacji pojawiły się problemy związane z podziałem jej na segmenty. Segmenty w literaturze poświęconej lingwistyce komputerowej i przetwarzaniu tekstów w języku naturalnym określa się też jako tokeny (ang. *tokens*). Podział ten polega na rozpoznawaniu granic między podstawowymi elementami tekstu – segmentami. Segmentacja tekstu definiowana jest też jako liniowy podział tekstu na co najmniej dwóch poziomach [11]. Pierwszy poziom stanowi podział tekstu na jednostki, zwykle zdania, które mogą być przetwarzane składniowo niezależnie od innych jednostek tego samego poziomu. Drugi poziom stanowi segmentacja tekstu prowadząca do tego, że tekst dzielony jest na jednostki nazwane tokenami lub segmentami, którym przypisuje się interpretacje morfo syntaktyczne, czyli informacje o częściach mowy (rzeczownik, czasownik itp.) i wartościach odpowiednich kategorii morfo syntaktycznych (rodzaju, przypadku itp.). Zazwyczaj segmentacja w tym sensie nazywana jest tokenizacją. Dodatkowo dla poprawy dalszej interpretacji tekstu, a więc i jakości, ważne jest rozpoznawanie segmentów charakterystycznych dla tekstów danego typu, np. dat, adresów, nazw ulic [12]. W badaniach prowadzonych przez autora ważny aspekt stanowił podział tekstu na pierwszym poziomie. Ważne jest to ze względu na fakt, że każdemu wydzielonemu segmentowi z raportu w procesie klasyfikacji nadawane jest znaczenie, określany jest jego kontekst. Odbywa się to poprzez analizę jego elementów składowych – wyrażeń. Na ich podstawie budowany jest klasyfikator, który przydziela segment do jednej z wydzielonych klas semantycznych (określających kontekst). Nieprawidłowa segmentacja może więc doprowadzić nie tylko do niepoprawnego podziału zdania na części, ale także do nieprawidłowej interpretacji semantycznej segmentu.

W literaturze dziedzinowej dotyczącej przetwarzania tekstów [11-13] mało miejsca poświęca się metodom projektowania segmentatorów regułowych na poziomie zdań. W niniejszym tekście omówiono więc metodę projektowania bazy wiedzy oraz reguł utworzonego i badanego przez autora segmentatora regułowego SR. Skonstruowana baza wiedzy umożliwiła przeprowadzenie segmentacji polegającej na rozpoznawaniu granicy zdań w dostępnych dla autora dokumentach tekstowych, w postaci raportów sporządzanych z akcji ratowniczo-gaśniczych, przechowywanych w systemie ewidencji zdarzeń EWID [8-10]. Okazało się, że zadanie to nie jest proste w przypadku próby segmentacji badanej dokumentacji. Do jego rozwiązania autor zaproponował prostą i skuteczną metodę, w kontekście analizowanej dokumentacji, opartą o regułowe dzielenie tekstu na segmenty. Do realizacji procesu segmentacji zaprojektowano, w oparciu o formalną analizę pojęć (ang. *formal concept analysis* – FCA), bazę wiedzy zawierającą używane w dokumentacji skróty oraz bazę reguł określającą warunki segmentacji.

W sekcji drugiej niniejszego artykułu opisano propozycje wykorzystania formalnej analizy pojęć do projektowania bazy wiedzy oraz reguł SR, na potrzeby segmentacji raportów. Wykorzystano także związane z tą analizą diagramy liniowe w celu wizualizacji relacji, jakie zachodzą pomiędzy wykrytymi obiektami. W sekcji trzeciej przedstawiono wyniki eksperymentu polegającego na wykorzystaniu wykrytych reguł oraz skrótów do segmentacji dostępnego zbioru danych tekstowych w postaci raportów. Przeprowadzona segmentacja za pomocą SR, skonstruowanego w oparciu o analizę utworzonej hierarchii pojęć, została poddana ocenie w odniesieniu do dostępnych autorowi dwóch segmentatorów. Pierwszy z segmentatorów wykorzystywał rozszerzone reguły segmentacji (ang. *segmentation rules exchange* – SRX), drugi natomiast pochodził z otwartego projektu związanego z przetwarzaniem języka polskiego (ang. *open source projects related to natural language processing* – openNLP) [14, 15]. W sekcji czwartej przedstawiono wnioski płynące z zastosowania opisywanej i proponowanej przez autora metody projektowania bazy wiedzy oraz reguł SR.

2. Segmentator regułowy – metoda projektowania i reprezentacja wiedzy

Formalna analiza pojęć wprowadzona została przez Rudolfa Wille'a w 1984 roku. Jej koncepcja zbudowana została na teorii sieci i częściowego porządku, które to zostały rozwinięte przez Birkhoffa i innych w latach 30. XX wieku [16-18]. FCA służy do matematyzacji pojęcia „pojęcie" (określane także jako „koncept") oraz daje formalne narzędzie stosowane 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 prezentowana za pomocą diagramu liniowego (ang. *line diagram*) nazywanego także diagramem Hassego (ang. *Hasse diagram*) [19, 20]. Diagram ten służy do konstruowania hierarchii pojęć. Składa się z węzłów (wierzchołków) oraz krawędzi. Każdy wierzchołek reprezentuje pojęcie natomiast krawędzie łączą wierzchołki w określony sposób [19]. Aktualnie FCA stosowana jest np. w [16]: psychologii, socjologii, antropologii, medycynie, biologii, lingwistyce, matematyce czy też informatyce. Autorowi najbliższe są zastosowania z zakresu technik informacyjnych i informatyki, w których niniejsza analiza wykorzystywana jest w szczególności do realizacji zadań z zakresu:

- wydobywania z tekstu hierarchii pojęć (ang. *concept hierarchies*) dla systemów bazujących na wiedzy [21] tj. systemów komputerowych stosujących wiedzę z danej dziedziny zapisanej w bazie wiedzy [22]. 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 w: eksploracyjnej analizie tekstów, przetwarzaniu informacji (ang. *information extraction* – IE) czy też wyszukiwaniu informacji (ang. *information retrievel* – IE)

DOI:10.12845/bitp.34.2.2014.9

w zbiorze dokumentów tekstowych. W ostatnim przykładzie FCA pełni najczęściej rolę silnika wspierającego systemy wyszukiwania informacji w tekście [20]. Natomiast diagramy liniowe służą do tworzenia i wizualizacji ich hierarchii oraz powiązań,

- analizy kodu źródłowego [23], a w szczególności pozyskiwania i grupowania wzorców projektowych [24, 25], jak też analizy, projektowania, tworzenia oraz refaktoryzacji hierarchii klas z zakresu paradygmatu projektowania obiektowego [17, 19, 26-32]. FCA w tym przypadku służy więc do zarządzania i rozwoju oprogramowania w myśl ogólnie pojętej inżynierii programowania [33] jak i modelowania całych systemów informatyczno-informacyjnych [34, 35],
- wspierania projektowania systemów CBR [36] oraz ich udoskonalania [37] poprzez np. grupowanie i selekcję przypadków zdarzeń [38, 39],
- wykrywania zależności funkcyjnych (ang. *functional dependencies*) w relacyjnych bazach danych [40],
- tworzenia metod półautomatycznych do konstruowania wybranych ontologii [41-43].

Propozycja analizy bazy wiedzy na temat skrótów oraz reguł dla SR bazuje na formalnej analizie pojęć oraz diagramach liniowych do wizualizacji wykrytych relacji między obiektami. Metoda analizy zawiera trzy podstawowe kroki, na które składają się następujące elementy: zdefiniowanie obiektów *O*, atrybutów *C* oraz relacji incydencji, następnie zdefiniowanie kontekstu formalnego *K* w terminach obiektu, atrybutu i relacji incydencji i na końcu zdefiniowanie pojęcia formalnego dla danego kontekstu formalnego.

Kontekstem formalnym *K* jest następująca trójka [41]:

$$K\,(O,C,R) \tag{1}$$

gdzie:
O – nieupsty zbiór obiektów,
C – niepusty zbiór atrybutów,
R – binarna relacja między obiektami a atrybutami $R \subseteq O \times C$ (oRc).

W niniejszym opracowaniu kontekst formalny stanowiły „elementy nie zawsze kończące segment". Kontekst ten został opisany za pomocą tablicy zawierającej: obiekty *o*, atrybuty *c* oraz relacje *r*. Zbiór obiektów stanowiły wykryte niepoprawnie rozbite segmenty, które zostały oznaczone jako $o_1,\ldots,o_n$ (n – liczba obiektów, n=310) i które prezentuje tabela 1. Zbiór atrybutów *C* stanowiły pojęcia określające, jakiego rodzaju przetwarzania segmentu należy użyć, aby prawidłowo podzielić segment. Wykryto i zdefiniowano 16 ($c_1,\ldots,c_k$, k = 16) takich charakterystycznych atrybutów (k) dla segmentów pochodzących z badanych raportów.

W celu zaprezentowania wyznaczania obiektów *o* i atrybutów *c* posłużono się następującym przykładem. Przyjmijmy, że do dyspozycji jest następujący segment:

„... podjęto decyzje że w dn. jutrzejszym zostanie zadysponowana przez gcrk na miejsce zdarzenia koparka, która wykona kanał do …".

W przypadku gdy będzie dostępna jedynie reguła mówiąca o tym, że znak kropki „." kończy segment, wówczas ww. segment zostanie nieprawidłowo podzielony na dwa podsegmenty. Tak więc można wykryć obiekt o_n w postaci wyrażenia „dn. jutrzejszym". Pierwszy element w rozważanym przypadku stanowi skrót, drugi natomiast resztę części segmentu. W celu prawidłowej segmentacji ww. zdania należy więc wprowadzić przetwarzanie polegające na wykrywaniu w tym przypadku atrybutu c_k w postaci skrótu – wyrażenie „dn." oznacza skrót od dnia.

Do wyznaczonych, podczas analizy dostępnych obiektów o_n, pozostałych atrybutów c_k należą atrybuty określające, czy dany segment powinien być zanalizowany pod kątem następujących elementów:

- skrótów (c_1, „skrót"), zbudowanych z jednej lub kilku liter i stanowiących wszelkie możliwe skrócone formy zapisu wyrazów lub wyrażeń, które występowały w badanych raportach,
- reguł (c_2, „reguła") określających i nakładających dodatkowe warunki co do podziału zdania na segmenty bądź braku takiego podziału (kropka nie zawsze implikuje koniec zdania),
- reguł z korekcjami (c_3, „reguła_korekcji") polegających na powierzchownym sprawdzeniu badanego segmentu, wykryciu oraz poprawie nieprawidłowo sformułowanych skrótów (w badanych tekstach najczęstszym błędem było bezpośrednie łączenie liczebników ze skrótami np. „… 10cm", które należało by poprawić na „… 10 cm."),
- reguł z badaniem otoczenia skrótu (c_4, „reguła_badania_otoczenia"), polegających na wykrywaniu, czy z lewej oraz prawej strony skrótu nie występują dodatkowe znaki. W tym przypadku analizowane były ciągi z segmentu pod kątem wykrywania w nich skrótów (w badanych tekstach istnieją zapisy używające wtrąceń w postaci nawiasów „(" oraz „)", po których następują skróty np. „… (dow. sierż.",
- reguł wykrywania nazwy ulicy ze skrótem (c_5, „nazwa_ulicy_ze_skrótem"), polegających na wykrywaniu ciągów w segmentach odnoszących się do nazw ulic, podczas zapisu których użyto skrótu imienia (w badanych tekstach istnieją zapisy w postaci np. „e. plater", który stanowi skrót od pełnej nazwy ulicy Emilii Plater,
- reguł wykrywania czasu (c_6, „czas") polegających na wykrywaniu ciągów w segmentach, które odnoszą się do określenia czasu akcji zapisywanego w formacie hh.mm (godzina.minut), tak więc znak kropki w takim zapisie nie powinien dzielić segmentu,
- reguł wykrywania liczby z kropką oraz adnotacją (c_7, „liczba_kropka_adnotacja"), polegających na wykrywaniu ciągów w segmentach zawierających liczbę, po której następuje kropka, a następnie symbol adnotacji np. „6. Ad.3", schemat taki wynika z tego, iż po zakończeniu akcji w polu nr. 6 pt. *Inne uwagi dotyczące danych ze strony poprzedniej* z sekcji *Dane opisowe do informacji ze zdarzenia* pochodzącej z papierowej wersji karty *Informacji ze zdarzenia* [44], KDR wpisują swoje uwagi dotyczące pozostałych pól z karty, które wypełniali. Z uwagi na brak wewnętrznej struk-

tury tego pola jak i struktury całego punktu *Dane opisowe do informacji ze zdarzenia* w cyfrowym systemie ewidencji zdarzeń EWID [5, 8-10], KDR stosują różne oznaczenia oraz zabiegi składniowe przy wprowadzaniu opisów do tego typu pól ww. systemu,

- reguł wykrywania gwiazdki z numerem oraz kropką (c_8, „gwiazdka_numer_kropka"), polegających na wykrywaniu ciągów w segmentach zawierających liczbę poprzedzoną znakiem gwiazdki lub innym znakiem, po której następuje kropka np. „*1.", schemat taki wynika z podobnych przesłanek, które omówiono powyżej. W elektronicznej wersji sekcji *Dane opisowe do informacji ze zdarzenia* brak jest wydzielonych odpowiednich sekcji, jak to ma miejsce w jej papierowej wersji, przez co KDR stosują różne nieformalne zabiegi w celu podkreślenia do jakiej części sekcji należy podany opis. Tego typu zabiegi nie są dominującą regułą, niemniej występują i powinny być brane pod uwagę podczas przetwarzania przez SR,
- reguł wykrywania liczby ze skrótem kończącym segment (c_9, „liczba_skrót_koniec_segmentu"), polegających na wykrywaniu ciągów w segmentach zawierających liczby ze skrótem, które kończą zdanie,
- reguł wykrywania kropki z liczbą oraz kropki, po której następuje wielka litera (c_{10}, „kropka_liczba_kropka_wielka_litera"), polegających na wykrywaniu ciągów na styku segmentów zawierających kropkę z liczbą oraz kropkę, po której następuje wielka litera. Schemat ten jest stosowany przez decydentów do wyliczania w opisywanym zdarzeniu kroków, jakie podjęli w celu neutralizacji powstałego zagrożenia np. „1. Wyważono drzwi. 2. Zabezpieczono miejsce zdarzenia.",
- reguł wykrywania wersji (c_{11}, „wersja") polegających na wykrywaniu ciągów w segmentach, które opisują wersje wykorzystywanych przyrządów pomiarowych np. zadymienia etc.,
- reguł wykrywania numerów z dowolnym znakiem oraz skrótem (c_{12}, „numer_znak_skrót_skrót") polegających na wykrywaniu ciągów w segmentach zawierających w sobie cyfry, po których może nastąpić znak ze skrótem. Schemat ten wyznacza w szczególności (w kontekście *Q* dotyczącym opisów hydrantów) opisy dotyczące obiektów hydrotechnicznych oraz ich sprawności np. „…,39582-n. spr." stanowi skrócony zapis dotyczący informacji o tym, że hydrant o numerze z tabliczki 39582 został sprawdzony i był niesprawny,
- nieinterpretowalnych skrótów (c_{13}, „skrót_nieinterpretowalny") zbudowanych z jednej lub kilku liter i stanowiących wszelkie możliwe skrócone formy zapisu wyrazów lub wyrażeń, które występowały w badanych raportach i nie można ich w żaden sposób zinterpretować jednoznacznie przy dyspozycji kontekstem *Q* segmentu np. „… b. Jan Kowalski", skrót „b." może oznaczać brygadier, brygadzista etc.,
- interpretowalnych skrótów (c_{14}, „skrót_interpretowalny") zbudowanych z jednej lub kilku liter i stanowiących wszelkie możliwe skrócone formy zapisu wyrazów lub wyrażeń, które występowały w badanych raportach i które można zinterpretować jednoznacznie przy dyspozycji kontekstem *Q* segmentu np. „… spłonęło 10 km. kwadratowych łąki.",
- poprawnych skrótów (c_{15}, „skrót_poprawny"), które z definicji są podobne do interpretowalnych skrótów przy czym skróty stanowią podzbiór zbioru poprawnych skrótów używanych w języku polskim,
- niepoprawnych skrótów (c_{16}, „skrót_niepoprawny"), które z definicji są podobne do interpretowalnych skrótów przy czym skróty stanowią nadzbiór zbioru poprawnych skrótów używanych w języku polskim np. „.. w. wym miejscu spłonęły śmieci" skrót „w. wym" odnosi się do wyżej wymieniony i poprawnie powinno być „ww.".

Informację o zależności pomiędzy wykrytymi obiektami stanowiącymi niepoprawne zakończone segmenty oraz określającymi ich atrybutami prezentuje tabela 1.

Tabela 1.

Tabela formalnego kontekstu
„elementy nie zawsze kończące segment"

Table 1.

Table of formal context
"elements which do not always end a segment"

Obiekty (Objects)	**Atrybuty (Attributes)**				
	c_1	c_2	c_3	…	c_k
o_1	1				
o_2		1	1		
o_3		1			
…					
o_n					

Źródło: opracowanie własne / Source: own work

Tabela 1 prezentuje 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 (oRc). Z kontekstu formalnego *K* można wywnioskować następujące zależności: zbiór obiektów $A \subseteq O$ generuje zbiór atrybutów $A' = \{c \in C \mid oRc, \forall o \in O\}$ (zbiór atrybutów dzielony przez obiekty z *A* np. A = {o2, o3} → A' = {c2}) i analogicznie zbiór atrybutów $B \subseteq C$ generuje zbiór obiektów $B' = \{o \in O \mid oRc, \forall c \in C\}$ (zbiór obiektów, które mają wszystkie atrybuty w *B* np. B = {c2} → B'={o2, o3}).

Formalne pojęcie (ang. *formal concept*) kontekstu K(O, C, R) stanowi para uporządkowana (A, B), gdzie [41]: $A \subseteq O$, $B \subseteq C$ oraz A' = B i B' = A. *A* nazywane jest ekstensją natomiast *B* nazywane jest intensją formalnego pojęcia (A, B).

Z każdym pojęciem związane więc są jego: ekstencja i intensja. Ekstensja to klasa przedmiotów (obiektów) opisywanych przez pojęcie. Natomiast intensja to klasa cech (własności, atrybutów) wspólnych dla wszystkich przedmiotów z ekstensji. Utworzone pojęcia dla omawianego formalnego kontekstu prezentuje tabela 2.

DOI:10.12845/bitp.34.2.2014.9

Tabela 2.

Pojęcia dla formalnego kontekstu „elementy nie zawsze kończące segment”

Table 2.

Concepts for a formal context “elements do not always ending segment”

Identyfikator pojęcia (ID concept)	Ekstensja (Extension)	Intensja (Intension)
c(0)	$\{o_{0,1}, \ldots, o_{0,a}\}$	{}
c(1)	$\{o_{1,1}, \ldots, o_{1,b}\}$	{reguła}
c(2)	$\{o_{2,1}, \ldots, o_{2,c}\}$	{reguła; numer_znak_skrót_skrót}
c(3)	$\{o_{3,1}, \ldots, o_{3,d}\}$	{reguła; wersja}
c(4)	$\{o_{4,1}, \ldots, o_{4,e}\}$	{reguła; gwiazdka_numer_kropka}
…	…	…
c(15)	$\{o_{15,1}, \ldots, o_{15,p}\}$	{skrót}
c(16)	$\{o_{16,1}, \ldots, o_{16,r}\}$	{skrót; skrót_niepoprawny}
c(17)	$\{o_{17,1}, \ldots, o_{17,s}\}$	{skrót; skrót_interpretowalny}
c(18)	$\{o_{18,1}, \ldots, o_{18,t}\}$	{skrót; skrót_interpretowalny; skrót_niepoprawny}
c(19)	$\{o_{19,1}, \ldots, o_{19,w}\}$	{skrót; skrót_interpretowalny; skrót_poprawny}
c(20)	$\{o_{20,1}, \ldots, o_{20,x}\}$	{skrót; skrót_nieinterpretowalny; skrót_niepoprawny}
c(21)	$\{o_{21,1}, \ldots, o_{21,y}\}$	{skrót; reguła; reguła_korekcji; reguła_badania_otoczenia; nazwa_ulicy_ze_skrótem; czas; … ; skrót_niepoprawny}

Źródło: opracowanie własne przy wykorzystaniu [45] / Source: own work based on [45]

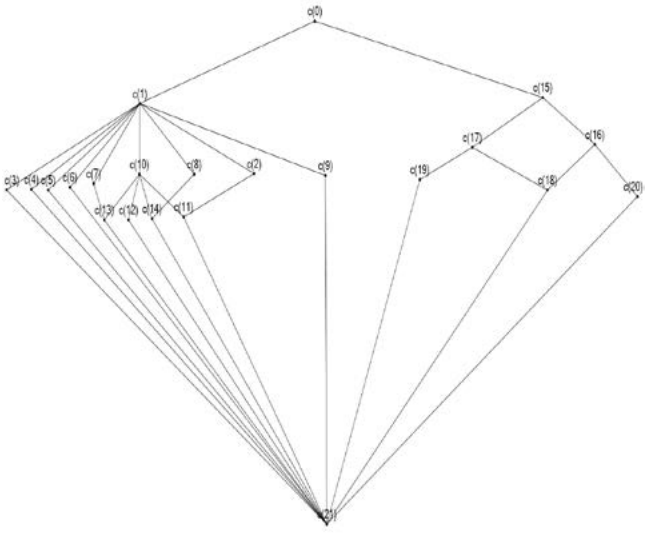
Tabela 2 prezentuje pojęcia dla formalnego kontekstu „elementy nie zawsze kończące segment”. Ze względu na dość znaczną liczbę przebadanych obiektów i wykorzystanie diagramu liniowego w celu prezentacji pomiędzy nimi relacji oraz zachowanie czytelności, wykorzystano następującą notację: identyfikator pojęcia c(l), gdzie l = 1,…,21 oznacza liczbę wyznaczonych formalnych pojęć, stanowi skrót zapisu pojęcia formalnego w postaci ($\{o_{xy},\ldots,o_{zk}\}$, $\{c_{xy},\ldots,c_{zk}\}$), tak więc w każdej parze pierwszy zbiór stanowi ekstensję pojęcia, natomiast zbiór drugi jego intensję.

Pojęcia (A1, B1) oraz (A2, B2) kontekstu K(O, C, R) są uporządkowane względem relacji, którą można zdefiniować w następujący sposób [41]:

$$(A_1,B_1) \le (A_2,B_2) \overset{def}{\Longleftrightarrow} A_1 \subseteq A_2 B_2 \subseteq B_1 \quad (2)$$

Zbiór wszystkich pojęć *S* kontekstu *K* wraz z relacją ≤ (S(K), ≤) tworzą kratę, która w analizie FCA nazywana jest kratą pojęć formalnego kontekstu K(O, C, R) [41]. Utworzoną przykładową kratę pojęć dla formalnego kontekstu „elementy nie zawsze kończące segment” prezentuje ryc. 1.

Ryc. 1 prezentuje utworzoną kratę pojęć dla formalnego kontekstu „elementy nie zawsze kończące segment”. Każdy węzeł sieci, który oznaczony jest jako czarna kropka, stanowi formalne pojęcie z tabeli 2. Na niniejszym rysunku widać wyraźnie rozdzielenie pomiędzy pojęciami związanymi z elementami prostymi w postaci skrótów (pojęcie c(15)) a elementami złożonymi w postaci reguł (pojęcie c(1)). Elementy proste, atomowe budują bazę wiedzy SR. Baza wiedzy może zostać podzielona na skróty niepoprawne (pojęcie c(16)) oraz skróty interpretowalne (pojęcie c(17)). Skróty niepoprawne nie występują w słowniku języka polskiego, ale są na tyle często stosowane w raportach, że można je uznać za część specyficznego języka, jakim posługują się KDR podczas tworzenia raportów. Tak więc w kontekście *Q* związanym z tworzeniem raportów uznane są jako równoważne skrótom poprawnym, które zarazem są interpretowalne (pojęcie c(19)). Interpretowalne mogą być skróty zarówno poprawne, jak i niepoprawne (pojęcie c(18)). Oznacza to, że użytkownik *U* może bez problemu na podstawie kontekstu *Q* segmentu zinterpretować znaczenie utworzonego zapisu w postaci skrótu. Niewielką część bazy wiedzy można wydzielić na wyjątki w postaci wykrytych niestandardowych, nieinterpretowalnych skrótów zdarzających się w raportach, które można powiązać albo z błędami, albo z pośpiesznym wpisywaniem i przenoszeniem raportu do bazy ewidencji zdarzeń. Drugą ważną gałąź budującą SR stanowi gałąź zawierająca reguły (pojęcie c(1)), które opisują, w jaki sposób wykorzystywać elementy zgromadzone w bazie wiedzy w postaci skrótów wraz z dodatkowymi regułami polepszającymi jakość segmentacji raportów. Do podstawowych reguł należą reguły związane z pojęciami: wykrywania wersji (c(3)), wykrywania podpunktów i adnotacji w dokumentacji (pojęcia c(4), c(5), c(6)), czy też wykrywania skrótów zamkniętych w nawiasy klamrowe czy też inne znaki (pojęcie c(9)). Niższy poziom przetwarzania, segmentacji raportu może obejmować analizę z odpowiednią korekcją segmentów. Korekcji mogą być poddawane takie elementy jak czas (pojęcie c(7), c(13)), złe połączenia skrótów (pojęcie c(12)), nazwy ulic (pojęcia c(8), c(14)) czy też numery, po których występują znaki wraz ze skrótem (pojęcia c(2), c(11)).



Ryc. 1. Krata pojęć formalnego kontekstu „elementy nie zawsze kończące segment".
Źródło: opracowanie własne na podstawie [45]
Fig. 1. Lattice for formal context "elements do not always ending segment".
Source: own work based on [45]

3. Eksperyment numeryczny

Wiedza na temat elementów nie zawsze kończących segment, została utrwalona i zaprezentowana za pomocą kraty pojęć (ryc. 1). Utworzono ją na podstawie analizy raportów oraz budujących je segmentów, jak również analiz nieprawidłowo rozbitych segmentów. Nieprawidłowo rozbite segmenty otrzymywano w procesie dostrajania bazy wiedzy oraz reguł SR. Towarzyszyło temu odkrywanie elementów bazy wiedzy zarówno w postaci dostępnych w raportach skrótów, jak i reguł określających wykrywanie prawidłowych zakończeń segmentów [46]. Utworzoną kratę pojęć można poddać procesowi transformacji na reguły SR. Dzięki temu istnieje możliwość zbudowania elastycznego oprogramowania z możliwością przełączania reguł. Posługując się utworzoną kratą pojęć, można zbudować oprogramowanie przetwarzające raporty na segmenty na trzech zasadniczych poziomach: a) korekcji potencjalnie źle użytych skrótów (warstwa utworzona z pojęć c(11) – c(13)), b) rozszerzonego wykrywania wyjątków polegających na tym, że znak kropki „." nie zawsze kończy segment (warstwa utworzona z pojęć c(2) – c(9)) oraz c) wykrywania podstawowych (standardowych) skrótów w segmentach związanych z wykorzystaniem bazy wiedzy (pojęcia z gałęzi c(1)) z bazową regułą mówiącą o tym, że jeśli analizator natrafi na element z gałęzi c(1) to znak kropki „." nie świadczy o tym, że jest to koniec segmentu.

Zademonstrowano, jak działa oprogramowanie oparte na trzech ww. warstwach na następującym przykładzie. Przyjmijmy, że do dyspozycji jest następujący raport:

„Spłonęło 10km. kwadratowych łąki. Akcję gaśniczą ukończono o godz. 8.00 po przybyciu dodatkowych sił i środków".

Raport po przejściu przez pierwszą warstwę (a) podlega korekcji tj. wyrażenie „10km". poprawiane jest na „10 km". Następnie tak poprawiony tekst trafia do warstwy drugiej (b) wykorzystującej elementy z warstwy trzeciej (c) oprogramowania. W ten sposób podczas analizy wyrażenia „km.", które znajduje się w standardowej bazie skrótów (c), program nie uzna tego za koniec segmentu i przejdzie dalej. Koniec segmentu nastąpi po odczytaniu wyrażenia „łąki". Wyraz „łąki" nie znajduje się w bazie skrótów, tak więc następuje w tym miejscu pierwszy podział tekstu. Podczas analizy drugiej części raportu zostanie wykryty standardowy skrót „godz." i tym samym nie nastąpi zakończenie segmentu. W dalszej kolejności nastąpi wykrycie wyrażenia „8.00". Dzięki warstwie drugiej (b) oprogramowania, także w tym przypadku nie nastąpi podział zdania. Podczas analizy wyrażenia „8.00" zostanie dopasowana reguła „czas" mówiąca o tym, że liczba zakończona znakiem kropki „." po której następuje znowu liczba, nie kończy segmentu. Program po osiągnięciu wyrażenia „środków." ze względu na to, że nie występuje ono w bazie wiedzy oraz reguł, wydzieli następnie drugi segment.

Utworzona krata pojęć w punkcie 2 była rezultatem badań dotyczących zastosowania opracowanego przez autora SR do segmentacji raportów z akcji ratowniczo-gaśniczych [46]. Autor nie zastosował w całości przedstawionego rozwiązania bezpośrednio w SR. Wykorzystane zostały tylko niektóre gałęzie z utworzonej kraty pojęć zawierającej wydzielone pojęcia. Segmentator regułowy wykorzystywał kompletną gałąź c(15) związaną z bazą wiedzy oraz podstawowe reguły z gałęzi c(1) związane z regułami wykrywania końca segmentu jak również sprawdzania, czy znak interpunkcyjny w postaci kropki („.") kończy skrót a nie segment. Pomimo pominięcia warstw korekcji oraz reguł związanych z wykrywaniem niektórych skrótów połączonych z liczbami, autor otrzymał satysfakcjonujące rezultaty. Do oceny rozwiązania oraz jego porównania w odniesieniu do wybranych segmentatorów autor użył zależności zaczerpniętych z zakresu przetwarzania informacji (ang. *information retrievel* – IR) [47]. Wykorzystane i przedstawione dalej zależności zostały wyprowadzone na podstawie następująco sformułowanego założenia:

Załóżmy, że jest dostępny jednoelementowy zbiór zapytań Q i zapytanie $q \in Q$ oraz zbiór segmentów S i, że dla zapytania q dany jest zbiór segmentów zwróconych przez segmentator (system segmentacji) $R_q \subseteq S$ oraz zbiór istotnych segmentów S_q oznakowanych ręcznie ze zbioru segmentów S tj. $S_q \subseteq S$.

Dzięki tak sformułowanemu problemowi możliwe jest wyznaczenie (wzór 3 – wzór 5) [48-50]:

- precyzji (ang. *precision*) – pozytywnie przewidziane wartości (ang. *positive prediction value*)

$$P = P_{IR} = \frac{|S_q \cap R_q|}{|R_q|} \tag{3}$$

- przywołania (ang. *recall*) – wrażliwość (ang. *sensitivity*)

$$R = R_{IR} = \frac{|S_q \cap R_q|}{|S_q|} \tag{4}$$

DOI:10.12845/bitp.34.2.2014.9

- błędu

$$E = \frac{|S_q \setminus (S_q \cap R_q)| + |R_q \setminus (S_q \cap R_q)|}{|S_q \cup R_q|} \quad (5)$$

Podczas porównywania działania wytypowanych i zbadanych segmentatorów posłużono się dodatkowymi zależnościami (wskaźnikami) w postaci harmonicznej i entropii [49]. Zależności na wymienione wskaźniki prezentują się następująco:

- harmoniczna (średnia harmoniczna) miara F

$$F = 2\frac{P \cdot R}{P + R} \quad (6)$$

gdzie:
P – wartość precyzji,
R – wartość przywołania.

- entropia

$$H = \sum_{j=1}^{k} \frac{n_j}{n} H_j \quad (7)$$

$$H_j = -\sum_{i=1}^{m} p_{ij} \log(p_{ij}) \quad (8)$$

$$p_{ij} = \frac{n_{ij}}{\sum_{i=1}^{m} n_{ij}} \quad (9)$$

gdzie:
n_j – liczba segmentów w grupie *j*,
n – całkowita liczba segmentów,
H_j – entropia dla grupy *j*,
p_{ij} – prawdopodobieństwo klasy *i* w grupie *j,*
n_{ij} – liczba wystąpień etykiety klasy *i* w grupie *j*.

Dodatkowo przeprowadzono także podstawowe testy z zakresu statystyki na niezależność segmentacji od rodzaju (typu) raportu (test niezależności χ^2) oraz zgodności otrzymanych zbiorów segmentów ze zbiorem segmentów oznakowanym tj. referencyjnym zbiorem segmentów (test zgodności Kołmogorowa-Smirnova). Założono więc, że „dobry" segmentator powinien być niezależny od tego, na jakim tekście pracuje tj. jego długości wyrażonej w segmentach. Wszystkie niezbędne obliczenia zostały dokonane za pomocą funkcji statystycznych znajdujących się w oprogramowaniu R-project [51].

Wybrane statystyki oraz wskaźniki do porównania wytypowanych, zbadanych przez autora segmentatorów w zestawieniu z referencyjnym zbiorem segmentów (RZS), stanowiącym poprawnie wydzielone segmenty z dostępnych raportów, prezentuje tabela 3. RZS utworzony został z dostępnego zbioru raportów. W drodze losowania wybrano 3735 raportów, które manualnie posegmentowano. Otrzymano w ten sposób zbiór składający się z 12753 segmentów. Dodatkowo dla celów dalszych analiz raporty pogrupowano według ich długości wyrażonej za pomocą liczby budujących ich segmentów.

Tabela 3.

Statystyki wytypowanych, przebadanych segmentatorów

Table 3.

Statistics of selected and tested segmentation tools

	RZS	**Segm. SRX**	**Segm. openNLP**	**SR**
Język (Language)	PL	PL	EN	PL
Prawidłowe segmenty (Correct segments)	12753	11805	11506	12317
Nieprawidłowe segmenty (Uncorrect segments)	0	2051	2188	720
Średnia (Mean)	4.726809	5.303695	5.291953	4.881414
Wariancja (Variance)	8.953906	10.91289	12.18409	9.69413
Pierwszy kwartyl (25. percentyl) (First quartile)	3	3	3	3
Mediana (Median)	4	5	5	4
Trzeci kwartyl (75. percentyl) (Third quartile)	6	7	7	6
IQR	3	4	4	3
Precyzja (Precision)	1	0.8519775	0.840222	0.9447726
Przywołanie (Recall)	1	0.9256645	0.902219	0.965812
Błąd E (Error E)	0	0.1127062	0.1298824	0.04482358
Błąd względny (Relative error)	0	8,65%	7,38%	2,23%
Test zgodności (Conformance test)	0	0.0996	0.082	0.0224
Test niezależności (Independence test)	0	1039.031	984.8984	319.6931

Źródło: opracowanie własne / Source: own work

Tabela 4.

Wskaźniki określające jakość działania wytypowanych, przebadanych segmentatorów

Table 4.

Indicators for selected and tested segmentation tools

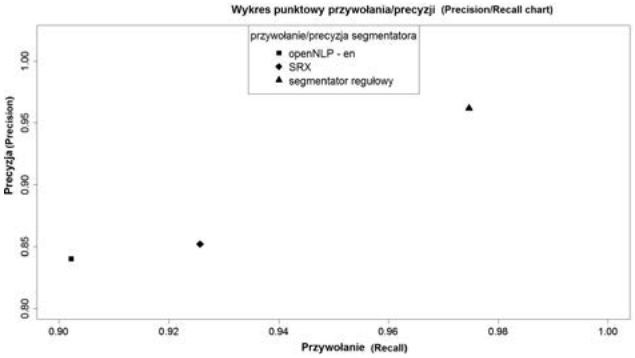
Segmentator (Segmentation toll name)	**Język (Language)**	**Precyzja (Precision)**	**Przywołanie (Recall)**	**Błąd (Error)**	**F-miara (F-mean.)**	**Entropia (Entropy)**
Segmentator regułowy SRX	PL	0.8519775	0.9256645	0.1127062	0.8872938	0.3265405
openNLP	EN	0.840222	0.902219	0.1298824	0.8701176	0.3661229
Segmentator regułowy	PL	0.9447726	0.965812	0.04482358	0.9551765	0.1790727

Źródło: opracowanie własne / Source: own work

Tabela 4 prezentuje zbiorcze porównawcze wartości wybranych wskaźników określających jakość działania przebadanych segmentatorów.

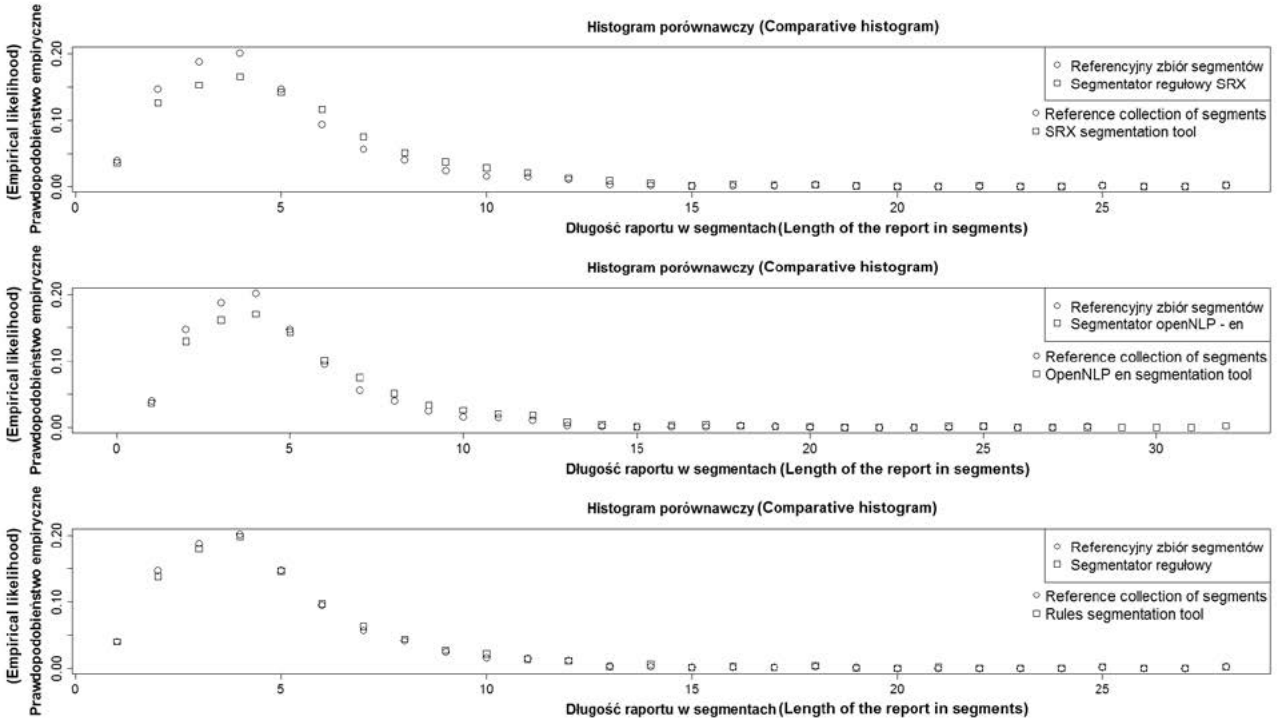
Wyniki, które prezentują tabele 3 i 4, wskazują wyraźnie lepsze działanie skonstruowanego segmentatora regułowego niż pozostałych użytych do porównania segmentatorów. Segmentacja pogarsza się wraz z próbą zmiany języka segmentatora, co wiąże się ze zmianą bazy wiedzy oraz reguł na temat wykrywania zakończeń segmentów, które w zależności od języka są różne. O pogorszeniu segmentacji świadczy zmiana statystyk w odniesieniu do RZS. Widać wyraźny wzrost średniej, jak również wariancji dla otrzymanych zbiorów segmentów z wybranych segmentatorów. Pociąga to za sobą zbytnie „rozdrobnienie" raportów. Zwiększa się liczba segmentów, a więc i liczba raportów o danym typie, które nie występują w RZS. Z tego wynika, że segmentacja, wykonana za pomocą segmentatorów wybranych do porównania z SR, stała się bardziej zależna od długości raportów. Małe wartości F miary oraz duże wartości entropii porównywanych segmentatorów i wytwarzanych przez nich zbiorów segmentów w odniesieniu do SR i RZS świadczą o znacznej ich „niejednorodności", a więc niesatysfakcjonującym ich działaniu i przetwarzaniu dostępnej dokumentacji.

Graficzna prezentacja wybranych statystyk i wartości wskaźników z badania wytypowanych, zbadanych segmentatorów w zestawieniu z referencyjnym zbiorem segmentów została dokonana za pomocą różnego rodzaju wykresów m.in.: precyzji/przywołania i histogramów. Wymienione wykresy prezentują ryc. 2 i 3.



Ryc. 2 Wykres punktowy przywołania/precyzji wytypowanych, zbadanych segmentatorów Źródło: opracowanie własne
Fig. 2. Precision/Recall chart for different segmentators
Source: own work

Ryc. 2 prezentuje wykres punktowy przywołania/precyzji wytypowanych, zbadanych segmentatorów. Zaprezentowano na nim zmianę wartości precyzji oraz przywołania w zależności od typu segmentatora (im bliżej punktu (1,1) tym lepsze działanie segmentatora). Można zaobserwować, że wraz ze zmianą reguł języka segmentatora jakość segmentacji wyraźnie spada. Zmniejsza się zarówno precyzja, jak i przywołanie. Widać, że segmentator SRX z regułami dla języka polskiego charakteryzuje się prawie taką samą precyzją jak segmentator openNLP dla języka angielskiego, wyróżnia go jedynie lepsze przywołanie. Oznacza to to, że ilościowo zwracają one taką samą liczbę segmentów, jednak w odniesieniu do RZS segmentator SRX jest nieznacznie lepszy. Precyzja i przywołanie utworzonego SR jest bliska punktu (1,1) świadczy to



Ryc. 3. Histogramy porównawcze rozkładu prawdopodobieństwa empirycznego segmentów
Źródło: opracowanie własne
Fig. 3. Comparative histograms
Source: own work

DOI:10.12845/bitp.34.2.2014.9

o tym, że wynikowy zbiór segmentów pokrywa się praktycznie z RZS.

Ryc. 3 prezentuje porównawcze histogramy rozkładu prawdopodobieństwa empirycznego będącego ilorazem liczby segmentów określających długość raportu do całkowitej liczby segmentów. Na każdym z wykresów prezentowanych na rycinie 3 znajduje się porównanie dopasowania obserwacji empirycznych pochodzących z otrzymanych zbiorów segmentów do RZS. Widać, że segmentator SRX oraz SR w odróżnieniu do segmentatora openNLP z ustawioną wersją angielską przywołują taki sam zbiór raportów o długości 28 segmentów. Niemniej opracowany SR znacznie lepiej pokrywa się z RZS w przedziale raportów składających się z 2-15 segmentów.

4. Wnioski

Zaproponowana metoda projektowania bazy wiedzy oraz reguł segmentatora regułowego umożliwia projektowanie oprogramowania, które w elastyczny sposób może dokonywać przełączania się pomiędzy różnymi, wybranymi gałęziami kraty pojęć.

Podstawowa reguła dotycząca wykrywania końca zdania poprzez interpretację kropki i dodatkowych znaków jako końca segmentu w rzeczywistości, zwłaszcza dla tekstów specjalistycznych musi być opakowana dodatkowymi regułami. Do budowy i reprezentacji takich reguł nadaje się przedstawiona w artykule formalna analiza pojęć. Wiedza inżyniera oraz dodatkowe eksperymenty mogą wzbogacać utworzoną sieć o nowe reguły. Nowo wprowadzana wiedza może zostać w łatwy sposób naniesiona na aktualnie utworzoną sieć semantyczną, tym samym przyczyniając się do polepszenia segmentacji tekstu. Tak więc FCA pełni doskonałą rolę wspierającą konstruowania oprogramowania w postaci segmentatora regułowego opartego o bazę wiedzy. Dzięki niej można w przejrzysty sposób skonstruować oprogramowanie komputerowe.

W ramach eksperymentu numerycznego wytworzono unikalny, w ramach dziedziny ratownictwa, zbiór reguł oraz elementów bazy wiedzy na temat stosowanych skrótów, jak również zbiór prawidłowo wydzielonych i oznakowanych segmentów z elektronicznej części *Karty informacji ze zdarzenia* w postaci sekcji zatytułowanej *Dane opisowe do informacji ze zdarzenia* [44]. W dalszej kolejności pozyskany zbiór segmentów będzie poddawany przetwarzaniu w torze formowania i strukturalizacji informacji. Na jego podstawie będzie podjęta próba budowy systemu informacyjnego dla krajowego systemu ratowniczo-gaśniczego.

Literatura

1. Mirończuk M., Karol K., *Koncepcja systemu ekspertowego do wspomagania decyzji w Państwowej Straży Pożarnej*, [w:] *Inżynieria Wiedzy i Systemy Ekspertowe*, Grzech A., Juszczyn K., Kwaśnicka H., Nguyen N.T. (red.), Akademicka Oficyna Wydawnicza EXIT, Warszawa 2009.
2. Mirończuk M., Maciak T., *Problematyka projektowania modelu hybrydowego systemu wspomagania decyzji dla Państwowej Straży Pożarnej*, „Zeszyty Naukowe SGSP", No 39, 2009.
3. Krasuski A., Maciak T. *Wykorzystanie rozproszonej bazy danych oraz wnioskowania na podstawie przypadków w procesach decyzyjnych Państwowej Straży Pożarnej*. „Zeszyty Naukowe SGSP", No 36, 2008, s. 17-35.
4. Mirończuk M. *Zmodyfikowana analiza FMEA z elementami SFTA w projektowaniu systemu wyszukiwania informacji na temat obiektów hydrotechnicznych w nierelacyjnym katalogowym rejestrze*, „Studia Informatica", Vol. 2, number 2B (97), 2011.
5. Mirończuk M. *Przegląd oraz zastosowanie metod eksploracji danych tekstowych do przetwarzania raportów z akcji ratowniczo-gaśniczych*. „Zeszyty Naukowe SGSP" (w cyklu recenzyjnym), 2011.
6. Słownik języka polskiego PWN. Hasło: s*trukturalizacja.* http://sjp.pwn.pl/slownik/2576375/strukturalizacja [dostęp: 1 kwietnia 2011]
7. Mirończuk M. Eksploracja Danych w kontekscie procesu Knowledge Discovery In Databases (KDD) i metodologii Cross-Industry Standard Process for Data Mining (CRISP-DM). Metody Informatyki Stosowanej, No 2, 2009.
8. Abakus: System EWID99, http://www.ewid.pl/?set=rozw_ewid&gr=roz, [dostęp: 1 maja 2009].
9. Abakus: System EWIDSTAT. http://www.ewid.pl/?set=ewidstat&gr=prod [dostęp: 1 maja 2009].
10. Strona firmy abakus. http://www.ewid.pl/?set=main&gr=aba [dostęp: 1 marca 2009].
11. Przepiórkowski A., *Techniki dezambiguacji morfo syntaktycznej. Powierzchniowe przetwarzanie języka polskiego.* Akademicka oficyna wydawnicza EXIT, Warszawa 2008. s. 17-45.
12. Mykowiecka A., *Elementy tekstu – segmenty, słowa, zdania. Inżynieria lingwistyczna. Komputerowe przetwarzanie tekstów w języku naturalnym*, Wydawnictwo PJWSTK, Warszawa 2007, s. 65-83.
13. Mykowiecka A., *Inżynieria lingwistyczna. Komputerowe przetwarzanie tekstów w języku naturalnym*, PJWSTK, Warszawa 2007.
14. Miłkowski M., Lipski J., *Using SRX Standard for Sentence Segmentation* In: *Human Language Technology Challenges for Computer Science and Linguistics*, Vetulani Z. (editor), Springer, Berlin/Heidelberg 2011. s. 172-182.
15. openNLP, http://incubator.apache.org/opennlp/ [dostęp: 1 kwietnia 20011].
16. Wolff KE., *A first course in formal concept analysis*. 1994. [dok. elektr.] http://www.fbmn.fh-darmstadt.de/home/wolff/Publikationen/A_First_Course_in_ Formal_ Concept_ Analysis.pdf [dostęp: 22 grudnia 2009].
17. Patil P,. *Applying Formal Concept Analysis to Object Oriented Design and Refactoring*, Bombay: Department Of Computer Science and Engineering Indian Institute Of Technology, 2009.
18. Priss U., *Formal concept analysis in information science*, "Annu Rev Inform Sci Tech", No 40, 2006, s. 521-543.
19. Hwang S. H., Kim H. G., Yang H. S., *A FCA-Based Ontology Construction for the Design of Class Hierarchy* In: Computational Science and Its Applications – ICCSA 2005, Gervasi O., Gavrilova M., Kumar V., Laganà A., Lee H., Mun Y., et al. (editors), Springer, Berlin/Heidelberg 2005. s. 307-320.
20. Carpineto C., Romano G. *Using Concept Lattices for Text Retrieval and Mining* In: Formal Concept Analysis, Ganter B., Stumme G., Wille R. (editors), Springer Berlin/Heidelberg, 2005. s. 3-45.
21. Cimiano P, Hotho A., Staab S. *Clustering concept hierarchies from text* In: *Proceedings of LREC*, 2004.
22. Leksyka.pl *Knowledge-based system,* http://megaslownik.pl/slownik/angielsko_polski/137416,knowledge-based+system [dostęp: 5 maja 2011].
23. Mens K., Tourw T., *Delving source code with formal concept analysis*. "Comput Lang Syst Struct", No 31, 2005, s. 183-197.
24. Muangon W., Intakosum S., *Retrieving design patterns by case-based reasoning and Formal Concept Analysis*.

[Beijing]: Computer Science and Information Technology, 2009 ICCSIT 2009 2nd IEEE International Conference, 2009.
25. Muangon W., Intakosum S., Adaptation of Design Pattern Retrieval Using CBR and FCA. Proceedings of the 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, 2009.
26. Arvalo G., Mens T., *Analysing Object-Oriented Application Frameworks Using Concept Analysis. Proceedings of the Workshops on Advances in Object-Oriented Information Systems*, 2002.
27. Felleisen M., *How to design class hierarchies*. [Tallinn, Estonia]: Proceedings of the 2005 workshop on Functional and declarative programming in education, 2005.
28. Proulx V. K., Gray K. E., *Design of class hierarchies: an introduction to OO program design*, "SIGCSE Bull", No 38, 2006, s. 288-292.
29. Godin R., Mili H., Mineau G. W., Missaoui R., Arfi A., Chau T. T., *Design of class hierarchies based on concept (Galois) lattices*, "Theor Pract Object Syst", No 4, 1998, s. 117-133.
30. Godin R., Valtchev P., *Formal Concept Analysis-Based Class Hierarchy Design in Object-Oriented Software Development* In: Formal Concept Analysis, Ganter B., Stumme G. and Wille R. (editors), Springer Berlin/Heidelberg 2005. p. 209-231.
31. Snelting G., Tip F. *Reengineering class hierarchies using concept analysis*, "SIGSOFT Softw Eng Notes", No 23, 1998, s. 99-110.
32. Snelting G., Tip F., *Understanding class hierarchies using concept analysis*, "ACM Trans Program Lang Syst", No 22, 2000, s. 540-582.
33. Tonella P., *Formal Concept Analysis in Software Engineering*, Proceedings of the 26th International Conference on Software Engineering, 2004.
34. Laukaitis A., Vasilecas O., *Formal concept analysis and information systems modeling*, [Bulgaria]: Proceedings of the 2007 international conference on Computer systems and technologies, 2007.
35. Hesse W., Tilley T., *Formal Concept Analysis Used for Software Analysis and Modelling,* In: Formal Concept Analysis, Ganter B., Stumme G. and Wille R. (editors), Springer Berlin/Heidelberg 2005. s. 259-282.
36. Díaz-Agudo B., González-Calero P. A., *Formal concept analysis as a support technique for CBR*, "Knowledge-Based Systems", No 14, 2001, s. 163-171.
37. Belén D. A., Marco A. G., Pedro P. G., Pedro A. G., *Formal concept analysis for knowledge refinement in case based reasoning*, Springer, 2005.
38. Pattaraintakorn P., Boonjing V., Tadrat J., *A New Case-Based Classifier System Using Rough Formal Concept Analysis*, Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology – Volume 02, 2008.
39. Li Y., Shiu S. C. K., Pal S. K., *Combining Feature Reduction and Case Selection in Building CBR Classifiers*, "IEEE Trans on Knowl and Data Eng", No 18, 2006, s. 415-429.
40. Rancz K. T. J., Varga V., *A method for mining functional dependencies in relational database design using FCA*, Studia Universitatis "Babes-Bolyai" Cluj-Napoca, Informatica, No LIII, 2008, s. 17-28.
41. Haav H., *A semi-automatic method to ontology design by using FCA*, University of Ostrava, Department of Computer Science. Ostrava, 2004.
42. Gliński W., *Ontologie. próba uporządkowania terminologicznego chaosu*, Instytut Informacji Naukowej i Studiów Bibliologicznych UW. [dok. elektr.] http://bbc.uw.edu.pl/Content/20/13.pdf [dostęp: 10 sierpnia 2010].
43. Hesse W., *Ontologies in the Software Engineering process*, EAI 2005 – Proceedings of the Workshop on Enterprise Application Integration, 2005.
44. 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.
45. Radvansky M., *Formal concept analyse*, [dok. elektr.] http://www.fca.radvansky.net/news.php [dostęp: 1 maja 2011]
46. Mirończuk M., *System informacyjny na temat sieci hydrantów dla krajowego systemu ratowniczo-gaśniczego: metoda segmentacji tekstu i jej ocena*, Białystok, 2011.
47. Markov Z., Larose D. T., *Wyszukiwanie informacji tekstowych i wyszukiwanie w Internecie. Eksploracja zasobów internetowych. Analiza struktury, zawartości i użytkowania sieci WWW*, Wydawnictwo Naukowe PWN, Warszawa 2009, s. 3-47.
48. Hand D., Mannila H., Smith P., *Eksploracja danych*. Wydanie 1., Wydawnictwo Naukowo-Techniczne, Warszawa 2005.
49. Markov Z., Larose D. T., *Eksploracja zasobów internetowych. Analiza struktury, zawartości i użytkowania sieci WWW*, Wydawnictwo Naukowe PWN, Warszawa 2009.
50. Christopher D. Manning, Prabhakar Raghavan., Schütze H., *Introduction to Information Retrieval* In: Press C.U., editor, 2008.
51. The R Project for Statistical Computing, http://www.r-project.org [dostęp: 1 stycznia 2011]

Praca naukowa współfinansowana ze środków Europejskiego Funduszu Społecznego, środków Budżetu Państwa oraz ze Środków Budżetu Województwa Podlaskiego w ramach projektu „Podlaska Strategia Innowacji – budowa systemu wdrażania"

dr inż. Marcin Michał Mirończuk – absolwent Wydziału Elektrycznego Politechniki Białostockiej, na którym także ukończył studia doktoranckie. Swoją rozprawę doktorską obronił na Wydziale Informatyki Politechniki Białostockiej w 2013 r. Aktualnie pracuje w Instytucie Podstaw Informatyki PAN w Warszawie.