Procedury recenzowania i doboru recenzentów

Tom drugi

red. naukowa Jarosław Protasiewicz

INNOWACYJNA GOSPODARKA
NARODOWA STRATEGIA SPÓJNOŚCI

OPI
OŚRODEK PRZETWARZANIA INFORMACJI

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO

Projekt współfinansowany przez Unię Europejską z Europejskiego Funduszu Rozwoju Regionalnego

Procedury recenzowania i doboru recenzentów, tom drugi
red. naukowa dr Jarosław Protasiewicz

Recenzent naukowy:
prof. dr hab. inż. Witold Pedrycz

Redakcja i korekta:
Anna Knapińska

Publikacja powstała w ramach realizacji subprojektu 5.1 „System wspomagania wyboru recenzentów" projektu systemowego Ministerstwa Nauki i Szkolnictwa Wyższego „Wsparcie systemu zarządzania badaniami naukowymi oraz ich wynikami" (Program Operacyjny Innowacyjna Gospodarka 2007–2013, Priorytet I, Działanie 1.1., Poddziałanie 1.1.3)

Autorzy:
Jan Artysiewicz, Sławomir Dadas, Małgorzata Gałężewska, Marek Kozłowski, Agata Kopacz, Jarosław Protasiewicz, Tomasz Stanisławek

Wydawca:
Ośrodek Przetwarzania Informacji – Instytut Badawczy
al. Niepodległości 188 b
00-608 Warszawa
tel. 22 570 14 00, fax 22 825 33 19
e-mail: opi@opi.org.pl
www.opi.org.pl

Warszawa 2012

ISBN 978-83-63060-07-7 (całość)
ISBN 978-83-63060-06-0 (tom drugi)

Projekt okładki i makiety:
Studio Artis sp. z o.o., ul. Wiśniowa 19, 02-562 Warszawa

Projekt graficzny publikacji, makiety, skład, druk i oprawa:
Grafpol Agnieszka Blicharz-Krupińska
ul. Czarnieckiego1
53-650 Wrocław
www.argrafpol.pl

SPIS TREŚCI

OD RECENZENTA

(profesor Witold Pedrycz)

Niniejsze opracowanie zajmuje szczególne miejsce na rynku wydawniczym. To wyjątkowo aktualna, obszerna i wysoce pożądana publikacja poświęcona procesowi recenzowania wniosków grantowych i publikacji naukowych. Kierowany przez dr. Jarosława Protasiewicza zespół autorski z laboratorium inteligentnych systemów informatycznych w Ośrodku Przetwarzania Informacji – Instytucie Badawczym podjął się ambitnego zadania przedstawienia tematyki, która jest ogromnie skomplikowana i niesie ze sobą mnóstwo wyzwań.

Recenzowanie grantów wymaga bardzo dużo uwagi, a sam proces może być wysoce kontrowersyjny. Wybór najbardziej obiecujących, innowacyjnych oraz w pełni realizowalnych grantów jest procesem wielokryterialnym, uwzględniającym aspekty, które muszą być poddane szczegółowej analizie. Biorąc pod uwagę istniejące uwarunkowania oraz złożoność samego procesu recenzowania, dobór recenzentów ma tutaj charakter wyjątkowo krytyczny. Każdy z nas, niezależnie od tego czy występował w roli recenzenta, czy też autora wniosku grantowego, zdaje sobie sprawę, że nie ma rozwiązań idealnych. Tym niemniej frapujące wydaje się krytyczne spojrzenie na to, jak procedury opiniowania przeprowadzane są w powszechnie uznanych agencjach grantowych o dużej tradycji (NSF, NIH, SFB) oraz jak przebiega recenzowanie prac w renomowanych czasopismach międzynarodowych. Wartościowe jest porównanie tych procesów, analiza ich głównych cech i identyfikacja ewentualnych słabych punktów, szczególnie w kontekście efektywnego (i konstruktywnego) wykorzystania elementów tych systemów na gruncie krajowym.

Autorzy w dogłębny sposób identyfikują i analizują główne czynniki postępowania recenzenckiego, zarówno pod kątem jakościowym, jak i ilościowym. Z punktu widzenia analizy jakościowej szczególnie ważne jest uwzględnienie heurystyk zniekształceń poznawczych. Aspekt ilościowy reprezentują analizy statystyczne dotyczące recenzentów. Duże znaczenie praktyczne ma nowatorska, automatyczna klasyfikacja często występującej informacji tekstowej, która tworzona jest z wykorzystaniem zaawansowanych metod rozpoznawania obrazów. *Pattern recognition* to technologia informatyczna o coraz ważniejszej pozycji w konstrukcji systemów inteligentnych, szczególnie tych ukierunkowanych na użytkownika.

Zaproponowany przez Autorów projekt systemu informatycznego oparty na wszechstronnej analizie istniejących rozwiązań oferuje interesującą architekturę, pozwalającą na skonstruowanie mechanizmów wyboru recenzentów, zbierania danych oraz wyszukiwania i rankowania rozwiązań.

Podsumowując, publikacja ma charakter unikatowy i z pewnością przyczyni się do lepszego zrozumienia procesów recenzowania, a także do poprawy ich jakości i skuteczności.

WPROWADZENIE

(dr Jarosław Protasiewicz)

Pierwszy tom opracowania wprowadził Czytelników w problematykę recenzowania. Niniejsze opracowanie jest drugim tomem, w którym przedstawiamy projekt, wybrane aspekty implementacji oraz niektóre testy systemu wspomagania wyboru recenzentów[1]. System został wykonany w laboratorium inteligentnych systemów informatycznych[2] Ośrodka Przetwarzania Informacji – Instytutu Badawczego na podstawie niepublikowanych, wcześniejszych prac: raportów analitycznych pod tytułem *Analiza porównawcza stosowanych metod doboru recenzentów*[3] i *Analiza porównawcza narzędzi informatycznych wspomagających dobór recenzentów*[4]; projektu[5] i prototypu systemu[6].

Celem przeprowadzonego badania ankietowego było zebranie opinii na temat procesu oceniania projektów i prac naukowych w Polsce. Ankiety skierowano do osób recenzujących wnioski grantowe składane w naszym kraju oraz do badaczy starających się o granty. Na większość pytań należało odpowiedzieć precyzyjnie, natomiast ostatnie pytanie, w którym respondenci wypowiadali się w formie swobodnej wypowiedzi, miało charakter otwarty. Wyniki badania zostały szczegółowo omówione w tomie pierwszym opracowania. Analiza pytania otwartego była wspomagana metodami uczenia maszynowego. W pierwszym rozdziale niniejszego tomu przedstawiono proces budowy automatycznego klasyfikatora swobodnych wypowiedzi recenzentów.

O metodach doboru ekspertów i procedurach recenzowania stosowanych przez wybrane instytucje (do oceny wniosków o granty) oraz czasopisma naukowe (do oceny artykułów naukowych) traktował tom pierwszy. W rozdziale drugim skupiono się na implementacji procedur doboru recenzentów i recenzowania w systemach informatycznych. Należy pamiętać, że szczegóły wdrożeniowe wielu systemów, szczególnie komercyjnych, są niedostępne; analizę ograniczono zatem do wybranych przypadków. W dalszej części rozdziału dosyć szczegółowo przedstawiono systemy informatyczne Ośrodka Przetwarzania Informacji – Instytutu Badawczego.

Biorąc pod uwagę krytykę systemów grantowych w Polsce oraz dysponując analizą metod recenzowania (tom pierwszy)[7,8], w rozdziale trzecim zaproponowano wstępną koncepcję systemu wspomagania wyboru recenzentów. Koncepcja miała za zadanie wyznaczyć kierunki dalszych poszukiwań optymalnego rozwiązania. Na jej podstawie opracowano projekt docelowego systemu wspomagającego dobór recenzentów[9]. Dołożono również starań, aby nie powielać błędów charakterystycznych dla innych systemów.

System zawiera pięć modułów merytorycznych: zbierania danych, klasyfikacji, identyfikacji osób, ekstrakcji słów kluczowych oraz rankingowania. Moduły te to właściwe algorytmy odpowiedzialne za przetwarzanie

[1] Subprojekt „System wspomagania wyboru recenzentów", zadanie 5.1 projektu systemowego „Wsparcie systemu zarządzania badaniami naukowymi oraz ich wynikami", w ramach priorytetu I, działanie 1.1., poddziałanie 1.1.3. Programu Operacyjnego Innowacyjna Gospodarka 2007–2013.

[2] http://lis.opi.org.pl, dostęp 18.07.2012.

[3] *Analiza porównawcza metod doboru recenzentów*, raport opracowany w ramach projektu „System wspomagania wyboru recenzentów", praca niepublikowana, OPI, Warszawa 2011.

[4] *Analiza porównawcza narzędzi informatycznych wspomagających dobór recenzentów*, raport opracowany w ramach projektu „System wspomagania wyboru recenzentów", praca niepublikowana, Warszawa 2011.

[5] *Projekt systemu*, opracowany w ramach projektu „System wspomagania wyboru recenzentów", niepublikowany, OPI, Warszawa 2011.

[6] System wspomagania wyboru recenzentów (prototyp), opracowany w ramach projektu „System wspomagania wyboru recenzentów", niepublikowany, OPI, Warszawa 2012.

[7] Zespół laboratorium inteligentnych systemów Informatycznych OPI, *Analiza porównawcza metod*…, op.cit.

[8] Zespół laboratorium inteligentnych systemów Informatycznych OPI, *Analiza porównawcza narzędzi*…, op.cit.

[9] Zespół laboratorium inteligentnych systemów Informatycznych OPI, *Projekt systemu*, op.cit.

danych. Użytkownik nie ma do nich bezpośredniego dostępu, dopiero wynik ich działania jest prezentowany przez interfejs użytkowania. W rozdziale czwartym przedstawione zostały mechanizmy działania modułów merytorycznych oraz testy wybranych algorytmów.

Podstawy teoretyczne dotyczące robotów internetowych, wybranych klasyfikatorów i metod analizy tekstu oraz algorytm identyfikacji autorów publikacji znajdą Czytelnicy w ostatnim, piątym rozdziale. Warto podkreślić, że wiedzę tę wykorzystuje się w wykonanym systemie.

Rozdział pierwszy

AUTOMATYCZNA KLASYFIKACJA ANKIETY RECENZENTÓW I WNIOSKODAWCÓW

(Jarosław Protasiewicz, Tomasz Stanisławek, Marek Kozłowski, Sławomir Dadas, Małgorzata Gałężewska, Agata Kopacz)

Autorzy dziękują prof. dr hab. Witoldowi Pedryczowi za konsultacje w zakresie uczenia maszynowego.

I. Kategorie wypowiedzi swobodnej

Omawianą szczegółowo w pierwszym tomie ankietę skierowaną do recenzentów i wnioskodawców wypełniło 8190 osób. W pytaniu otwartym wypowiedziało się 2615 z nich (32%). Automatyczną klasyfikację tekstu swobodnej wypowiedzi przeprowadzono w pięciu kategoriach problemów, które składały się z piętnastu podkategorii. Kategorie i podkategorie zostały wyodrębnione przez ekspertów[10], ich szczegółowy podział znajduje się w tabeli 1.

Tabela 1. Kategorie i podkategorie wypowiedzi swobodnej

Kategoria	Podkategoria
Recenzowanie	Sposób wyboru recenzentów Odwołanie Wytyczne Dialog Kontrola recenzenta
Ocena	Skala Kryteria Agregacja Rozbieżność ocen
Jakość pracy	Jakość recenzji Wiedza recenzenta Uczciwość Subiektywizm
Formalizm	–
Anonimowość	–

Źródło: opracowanie własne autorów rozdziału

II. Założenia przeprowadzonych doświadczeń

Doświadczenia miały na celu zbudowanie klasyfikatora, który automatycznie przyporządkuje podkategorię (klasę) każdej odpowiedzi na pytanie otwarte w ankiecie. Dzięki temu przyspieszona zostałaby praca eksperta

[10] Ekspertami byli członkowie zespołu laboratorium interaktywnych technologii, OPI.

dokonującego analizy jakościowej. Przy konstrukcji klasyfikatora potrzebny był zbiór uczący. Otwarta wypowiedź mogła poruszać kilka kwestii merytorycznych, wtedy do jednej wypowiedzi należało kilka podkategorii. Zdając sobie sprawę z takiej możliwości, opracowano specyficzny sposób zbierania danych treningowych – komentarz respondenta z przyporządkowaną więcej niż jedną klasą dzielono na fragmenty odpowiadające merytorycznie przydzielonym klasom. Takie ujęcie zapewniało spójność danych wejściowych w obrębie jednej podkategorii. Do klasyfikacji użyto metod uczenia maszynowego. Przed przystąpieniem do klasyfikacji, każdą wypowiedź poddawano wstępnemu przetwarzaniu tekstu. Proces ten składał się z trzech etapów[11]:

1. lematyzacja, wykonywana przy użyciu narzędzia Morfologik[12];
2. usunięcie wyrazów z listy *stop words*[13];
3. wykorzystanie postaci TF/IDF (*term frequency – inverse document frequency*) do określenia ważności słowa w zbiorze wypowiedzi.

Do oceny każdego modelu klasyfikatora posłużono się miarami: precyzja i dopasowanie.[14]

Przykładowe wyliczenia miary dopasowania zaprezentowano w tabeli 2.

Tabela 2. Przykład wyników klasyfikacji dla pięciu wypowiedzi respondenta

Nr testu	Prawidłowe podkategorie	Klasy przedzielone przez klasyfikator	Liczba poprawnych podkategorii	Miara dopasowania
1	C1, C2	C1	1	0
2	C3	C3, C5	1	0
3	C5	C5	1	1
4	C4	C2	0	0
5	C6, C7	C6, C7	2	1

Źródło: opracowanie własne autorów rozdziału

Przeprowadzone eksperymenty można podzielić na dwa etapy:

1. Testy różnych modeli klasyfikatorów oraz sposobów reprezentacji odpowiedzi na pytanie otwarte w ankiecie, w celu wyboru najlepszych modeli klasyfikatora – opisane poniżej eksperymenty 1–4.
2. Przeprowadzenie klasyfikacji wszystkich wypowiedzi – eksperymenty 5–17.

III. Testy różnych modeli klasyfikatorów

Aby wybrać najlepsze modele klasyfikatora, przeprowadzono testy różnych modeli oraz sposobów reprezentacji odpowiedzi na pytanie otwarte w ankiecie. Algorytm testów był następujący:

1. eksperci tworzą zbiór uczący dla wszystkich podkategorii o ustalonej liczności wypowiedzi;
2. na podstawie utworzonego zbioru przeprowadzane są testy wybranych modeli klasyfikatorów metodą walidacji krzyżowej, a następnie wybiera się model najlepszy;
3. wypowiedzi nowe, nieposiadające przydzielonej podkategorii, klasyfikowane są przez klasyfikator wybrany w punkcie 2;
4. eksperci weryfikują poprawność klasyfikacji (zazwyczaj trzeba było sprawdzić około stu wypowiedzi);
5. analizowana jest jakość klasyfikatora;
6. modyfikowane są parametry klasyfikatora;
7. następuje powrót do punktu 1, przy jednoczesnym zwiększeniu zbioru trenującego o wypowiedzi zweryfikowane w punkcie 4.

[11] Zagadnienia teoretyczne wyjaśniono w dodatku.
[12] Morfologik, http://morfologik.blogspot.com, dostęp 07.08.2012.
[13] Manning C., Schutze H., Prabhakar R., *Introduction to Information Retrieval*, Cambridge University Press, 2008.
[14] Zagadnienia teoretyczne wyjaśniono w dodatku.

Algorytm posłużył do przeprowadzenia czterech eksperymentów, które pozwoliły na klasyfikację 944 pytań otwartych (36,1% wszystkich wypowiedzi). Badano różne modele klasyfikatorów, stosując dwa rodzaje organizacji klasyfikacji:

- **klasyfikacja pozioma**, działająca tylko w obrębie podkategorii;
- **klasyfikacja hierarchiczna**, gdzie najpierw wykonuje się klasyfikację w obrębie kategorii, a następnie – w obrębie podkategorii.

Przed każdym z doświadczeń wykonano – metodą walidacji krzyżowej – testy modeli Multinomial Naive Bayes (MNB) i Support Vector Machine[15] (SVM) dla poziomej i hierarchicznej organizacji klasyfikacji. Wyniki zaprezentowano w tabeli 3.

Dla doświadczenia 1 utworzono zbiór uczący, składający się z 14 wypowiedzi dla każdej podkategorii. Dla takiej konstrukcji modelu testy metodą walidacji krzyżowej wykazały najlepszą skuteczność klasyfikatora MNB z organizacją poziomą. Z tego też powodu posłużono się nim przy klasyfikacji nowych wypowiedzi.

Zbiór uczący w doświadczeniu 2 składał się z 24 wypowiedzi. Zarówno dla klasyfikacji poziomej, jak i hierarchicznej, najwyższą efektywność zaprezentował MNB. W celu porównania skuteczności klasyfikacji poziomej i hierarchicznej, do klasyfikacji nowego zbioru wypowiedzi posłużyły oba modele. Wyniki tego eksperymentu ukazują wyższość podejścia poziomego nad hierarchicznym.

Przy doświadczeniu 3 i 4 zwiększono zbiory uczące, kolejno do 34 i 43 wypowiedzi. Najlepsze wyniki w teście osiągał poziomy klasyfikator MNB, wykorzystano go więc do klasyfikacji nowego zbioru wypowiedzi w owych eksperymentach.

Dużym problemem okazało się określenie poprawnej liczby podkategorii, jaką miał przyporządkowywać poszczególnym wypowiedziom konkretny model klasyfikatora. Największą precyzję i dopasowanie osiągnięto, gdy klasyfikator przypisywał do wypowiedzi tylko jedną najbardziej prawdopodobną klasę (tabela 3). Dzięki zastosowaniu MNB możliwe było sprawdzenie, czy występuje jakakolwiek korelacja między wartością prawdopodobieństwa dla klasy a liczbą podkategorii przyporządkowanych przez klasyfikator. Opierając się na eksperymentach 1–4, wykazano brak zależności pomiędzy wartością prawdopodobieństwa a liczbą klas przyporządkowanych przez klasyfikator. Taki stan rzeczy pokazał, że na tym etapie nie można poprawnie ocenić liczby podkategorii, które należy przyporządkować do danej wypowiedzi.

Aby rozwiązać problem liczby klas przyporządkowanych przez klasyfikator, zbudowano model, który sprawdzał przynależność do każdej podkategorii z osobna. Do jego konstrukcji wykorzystano 16 klasyfikatorów. Każdy z nich posiadał dwie klasy decyzyjne (przynależność do podkategorii oraz jej brak), co umożliwiało przyporządkowanie kilku klas jednocześnie, dla każdego komentarza respondenta, bez konieczności określania stałej liczby podkategorii zwracanej przez klasyfikator. Dalej model ten występuje pod nazwą *jeden vs. reszta*, co odnosi się do metody ewaluacji przynależności do pojedynczej podkategorii, w porównaniu z przynależnością do wszystkich pozostałych kategorii. Do analizy tego modelu ponownie wykorzystano eksperymenty 1–4.

Podczas początkowych doświadczeń na modelu *jeden vs. reszta* zaobserwowano istotny problem: klasyfikator przyporządkowuje nadmiarową liczbę podkategorii do jednej wypowiedzi. W celu wyeliminowania tego błędu podjęto próby ograniczenia liczby klas poprzez ustalenie progu określającego minimalne prawdopodobieństwo przyporządkowania do podkategorii. Pomimo wyznaczenia wysokiego progu prawdopodobieństwa wynoszącego 0,95, w dalszym ciągu model przypisywał dużą liczbę podkategorii do jednej wypowiedzi respondenta.

Na otrzymywane rezultaty pozytywnie wpłynęło zwiększenie zbioru uczącego dla każdego z klasyfikatorów. W wyniku tej próby liczba podkategorii przypisanych do jednej wypowiedzi respondenta oscylowała w gra-

[15] Zagadnienia teoretyczne wyjaśniono w dodatku.

nicach od jednej do trzech. Jednocześnie pojawiła się nowa trudność – nie każdej wypowiedzi została przyporządkowana któraś z podkategorii. Gdy komentarz nie uzyskał żadnego przyporządkowania, zastosowano model klasyfikatora z klasyfikacją poziomą.

Tabela 3. Porównanie jakości klasyfikacji w kolejnych eksperymentach dla klasyfikatora Multinomial Naive Bayes z klasyfikacją poziomą oraz hierarchiczną

	Liczba klas przydzielonych przez klasyfikator	Eksperyment 1		Eksperyment 2		Eksperyment 3		Eksperyment 4	
Klasyfikator MNB z klasyfikacją poziomą		*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*
	Jedna	35,71	41,07	42,48	55,75	24,73	54,84	25	48,08
	Dwie	0	27,68	39,82	51,09	16,13	48,48	21,15	42,94
	Trzy	0	20,68	40,71	48,34	13,98	40,51	–	–
	Cztery	0	17,86	–	–	–	–	–	–
Klasyfikator MNB z klasyfikacją hierarchiczną		*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*
	Jedna	–	–	30,97	44,25	–	–	–	–
	Dwie	–	–	30,09	40	–	–	–	–
	Trzy	–	–	30,09	37,67	–	–	–	–
	Cztery	–	–	–	–	–	–	–	–

Źródło: opracowanie własne autorów

Tabela 4. Skuteczność modelu *jeden vs. reszta* z wykorzystaniem klasyfikatora Multinomial Naive Bayes

Klasyfikator *jeden vs. reszta* z zastosowaniem klasyfikacji poziomej	Eksperyment 1		Eksperyment 2		Eksperyment 3		Eksperyment 4	
	Fit	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*
	28,57	38,33	36,28	49,23	22,09	56,04	26,67	52,85

Źródło: opracowanie własne autorów

Jak pokazują tabele 3 i 4, w dwóch pierwszych eksperymentach (eksperyment 1 i 2) klasyfikacja MNB uzyskiwała większą precyzję niż klasyfikacja *jeden vs. reszta*. Dopiero od trzeciego doświadczenia nastąpił nieznaczny wzrost jakości klasyfikacji metodą *jeden vs. reszta*. Testy pokazały, że zastosowanie tej metody klasyfikacji nie wpłynęło znacząco również na poprawę jakości przyporządkowań podkategorii do komentarzy respondentów.

Wykonano ponowne badania na doświadczeniach 1–4, dla kolejnych modeli klasyfikatorów, które potencjalnie mogłyby poprawić jakość klasyfikacji:

1. **Model z zastosowaniem słownika języka polskiego[16].** Część z wolnych wypowiedzi respondentów nie zawierała polskich znaków, co zakłócało proces lematyzacji, a tym samym cały proces klasyfikacji. Dzięki zastosowaniu sjp.pl można było sprawdzić niepoprawne wyrazy zawarte w badanym zbiorze i zamienić je na poprawną formę. Wykorzystano do tego odległość Levenshteina[17].
2. **Model z zastosowaniem słów kluczowych.** Wykorzystano autorską metodę (*Polish KeyWord Extractor*[18]) wydobywania słów kluczowych dla języka polskiego.
3. **Model z użyciem całego dostępnego zbioru trenującego.** Zbiór uczący został oparty o wszystkie komentarze, którym do tej pory przyporządkowano podkategorie. Stosując takie podejście, klasy wystę-

[16] *Słownik języka polskiego*, http://www.sjp.pl, dostęp 07.08.2012.
[17] Zagadnienia teoretyczne wyjaśniono w dodatku.
[18] Zagadnienia teoretyczne wyjaśniono w dodatku.

pujące częściej były liczniejsze w zbiorze uczącym, co zwiększyło tym samym prawdopodobieństwo ich wystąpienia. Zbiór uczący uwzględniał statystyczny rozkład wystąpień poszczególnych podkategorii na tle przebadanych dotychczas wypowiedzi.

Podczas eksperymentów przeprowadzono analizę zależności jakości klasyfikacji w zależności od liczby przydzielonych klas (podkategorii) (tabela 5). Najlepsze wskaźniki zapewniło przyporządkowanie jednej, najbardziej prawdopodobnej podkategorii. Model z zastosowaniem słów kluczowych osiągnął najgorsze rezultaty, prawdopodobnie z powodu nadmiernej redukcji merytorycznej krótkich tekstów wypowiedzi. W stosunku do modelu pierwotnego (wyuczonego na pełnych tekstach) spadek wynosił przeważnie kilka punktów procentowych. Począwszy od trzeciego eksperymentu, czyli dla większej liczby komentarzy w zbiorze trenującym, lepszą skuteczność uzyskiwał model z zastosowaniem słownika języka polskiego.

Tabela 5. Porównanie jakości klasyfikacji dla zmodyfikowanych modeli klasyfikatorów

	Liczba klas przydzielonych przez klasyfikator	**Eksperyment 1**		**Eksperyment 2**		**Eksperyment 3**		**Eksperyment 4**	
Model pierwotny, klasyfikator użyty w doświadczeniach pierwotnych		*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*
	Jedna	**35,71**	41,07	42,48	55,75	24,73	54,84	**25**	48,08
	Dwie	0	27,68	39,82	51,09	16,13	48,48	21,15	42,94
	Trzy	0	20,68	40,71	48,34	13,98	40,51	–	–
	Cztery	0	17,86	–	–	–	–	–	–
Model z zastosowaniem sjp.pl, punkt 1		*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*
	Jedna	28,57	35,71	36,28	48,67	25,81	59,14	**25**	**50,96**
	Dwie	0	21,43	34,51	45,99	15,05	48,48	20,19	41,81
	Trzy	0	20,83	34,51	44,37	12,9	40,93	–	–
	Cztery	0	18,3	–	–	–	–	–	–
Model z zastosowaniem słów kluczowych, punkt 2		*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*
	Jedna	25,36	30,36	28,32	40,71	21,51	53,76	21,15	42,31
	Dwie	1,79	25	27,43	39,42	13,98	45,45	17,31	37,85
	Trzy	0	20,83	26,55	37,75	11,83	40,51	–	–
	Cztery	0	16,96	–	–	–	–	–	–
Model z użyciem całego dostępnego zbioru trenującego, punkt 3		*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*	*Fit*	*Precision*
	Jedna	33,93	**44,64**	**49,56**	**62,83**	**29,03**	**63,44**	**25**	**50,96**
	Dwie	0	30,36	46,9	59,12	18,28	52,73	22,12	46,33
	Trzy	0	25,6	46,9	55,63	15,05	42,19	–	–
	Cztery	0	20,54	–	–	–	–	–	–

Pogrubioną czcionką oznaczono najlepsze wyniki dla poszczególnych eksperymentów

Źródło: opracowanie własne autorów

Poza pierwszym doświadczeniem, wyniki były lepsze niż w pierwotnej wersji klasyfikatora. Spośród różnych modeli zdecydowanie najlepszy okazał się MNB z klasyfikacją poziomą. Gorsze wyniki uzyskiwał model, w którym wykorzystano klasyfikację hierarchiczną; być może z tego względu, że podkategorie w niej zawarte nie były ze sobą znacząco powiązane.

IV. Klasyfikacja wszystkich wypowiedzi

Przed przystąpieniem do dalszego etapu eksperymentów zespół ekspertów poprawił zbiór uczący, poprzez weryfikację i uzupełnienie tekstów o dodatkowe kategorie. Początkowo pod uwagę wzięto teksty, które w metodzie walidacji krzyżowej uzyskały w fazie walidacji odmienne kategorie od manualnie przypisanych przez ekspertów. Następnie sprawdzono próbki tekstów o najmniejszej liczbie znaków, mogące negatywnie wpływać na jakość wyników klasyfikatora.

Doświadczenia 1–4 wykazały poprawę skuteczności klasyfikacji po zastosowaniu mechanizmu zamiany słów niepoprawnie napisanych, wykorzystując do tego słownik języka polskiego oraz odległość Levenshteina. Ponadto wykluczono model oparty o klasyfikację hierarchiczną, w którym skuteczność była zawsze niższa niż w modelu z klasyfikacją poziomą. W poprzednich eksperymentach dla klasyfikatora SVM użyto domyślnych parametrów, przez co jakość działania modelu z wykorzystaniem tego algorytmu mogła nie być optymalna.

Stosując się do wymienionych uwag, stworzono listę modeli klasyfikatorów:
1. klasyfikator Multinomial Naive Bayes (MNB);
2. klasyfikator Multinomial Naive Bayes (MNB), *jeden vs. reszta;*
3. klasyfikator Support Vector Machine (SVM);
4. klasyfikator Support Vector Machine (SVM) ze zmodyfikowanymi parametrami;
5. klasyfikator Support Vector Machine (SVM), *jeden vs. reszta;*

Wybór parametrów dla klasyfikatora SVM nie był zadaniem trywialnym. Nie istnieje automatyczna metoda, która umożliwia dobór najlepszych parametrów do zadanego problemu. Zatem w tym przypadku dopasowanie parametrów odbyło się w sposób intuicyjny. Modyfikacji uległ również schemat przebiegu doświadczeń:
1. losowy wybór stu nowych wypowiedzi, jeszcze niesklasyfikowanych;
2. klasyfikacja losowo wybranych wypowiedzi klasyfikatorem z najlepszym wynikiem precyzji z poprzedniego doświadczenia;
3. weryfikacja nowo sklasyfikowanych wypowiedzi przez eksperta;
4. klasyfikacja stu powyżej wybranych wypowiedzi wszystkimi rodzajami klasyfikatorów oraz wybór modelu o najlepszej precyzji;
5. przeprowadzenie kolejnego doświadczenia z uwzględnieniem w zbiorze uczącym manualnie zweryfikowanych wypowiedzi z aktualnego doświadczenia.

Tabela 6. Średnie dopasowanie i precyzja dla różnych modeli klasyfikatorów, w eksperymentach 5–9

Rodzaj klasyfikatora	Liczba przydzielonych klas	*Fit*	*Precision*
Klasyfikator MNB	Jedna	28,78	63,51
	Dwie	19,98	51,72
Klasyfikator SVM	Jedna	28,61	61,26
	Dwie	18,61	51,03
Klasyfikator SVM dla zmienionych ustawień (kernel gausowski RBF; gamma – 0,01; parametr C – 21)	Jedna	29,08	58,53
	Dwie	19,76	49,51
Klasyfikator MNB, *jeden vs. reszta*	Jedna	25,11	59,09
Klasyfikator SVM, *jeden vs. reszta*	Jedna	27,2	60,27

Źródło: opracowanie własne autorów

Tak jak w eksperymentach 1–4, w doświadczeniach 5–9 najlepszą skuteczność osiągnięto przy założeniu, że liczba podkategorii przydzielonych przez klasyfikator będzie równa 1. Dla doświadczeń przeprowadzonych w tej części optymalne rezultaty uzyskiwał model z wykorzystaniem klasyfikatora MNB, gdzie średnie dopasowanie wyniosło 28,78%, a średnia precyzja 63,51% (tabela 6). Niewiele gorsze efekty miał klasyfikator SVM, z dopasowaniem 28,61% i precyzją 61,26%. Warto zwrócić uwagę na to, że dla początkowych doświadczeń zdecydowanie doskonalszy był MNB. Przy wzroście liczby próbek w zbiorze trenującym, a tym samym wzroście liczby atrybutów, coraz lepiej spisywał się SVM. Mimo mniejszej wartości średniej obu wskaźników skuteczności klasyfikacji, w niektórych doświadczeniach osiągał on znacznie lepsze wyniki.

Na tym etapie ponownie zweryfikowano zbiór uczący. Podkategorie, które miały być przeanalizowane przez eksperta, zostały wybrane na podstawie ilości błędnie sklasyfikowanych wypowiedzi dla kolejnych doświadczeń. Najczęściej występującą pomyłką było przydzielanie klasy „anonimowość" do klasy o etykiecie „jawność". Po uzgodnieniach z ekspertem, klasy te połączono w jedną, oznaczoną jako „anonimowość". Zmodyfikowany zbiór trenujący, wraz z połączonymi dwiema podkategoriami posłużył do przeprowadzania kolejnych eksperymentów.

W celu otrzymania bardziej reprezentatywnej próbki danych, w dotychczasowym schemacie przeprowadzania doświadczeń do 150 zwiększono liczbę wypowiedzi biorących udział w badaniu. W ten sposób wyniki uzyskane w kolejnych eksperymentach nie powinny mieć dużych wahań dopasowania oraz precyzji.

Tabela 7. Średnie dopasowanie i precyzja dla różnych modeli klasyfikatorów, w eksperymentach 10–17

Rodzaj klasyfikatora	Liczba przydzielonych klas	*Fit*	*Precision*
Klasyfikator MNB	Jedna	27,14	72,46
	Dwie	21,34	59,74
Klasyfikator SVM	Jedna	27,82	73,24
	Dwie	22,22	60,96
Klasyfikator SVM, dla zmienionych ustawień (kernel gaussowski RBF; gamma – 0,01; parametr C – 21)	Jedna	29,03	76,18
	Dwie	24,36	61,7
Klasyfikator MNB, *jeden vs. reszta*	Jedna	23,29	65,67
Klasyfikator SVM, *jeden vs. reszta*	Jedna	26,62	70,3

Źródło: opracowanie własne autorów

Po korekcie zbioru uczącego oraz połączeniu dwóch podkategorii w jedną, najlepszą skuteczność uzyskano dla modelu wykorzystującego klasyfikator SVM przy zmienionych ustawieniach (tabela 7). Model oparty na klasyfikatorze MNB nie zanotował już tak dużej poprawy obu wskaźników. Najlepszą skuteczność osiągnięto w przypadku otrzymywania wyłącznie jednej klasy dla wypowiedzi respondenta. Zwiększenie i korekta zbioru uczącego przyczyniła się do poprawy wskaźnika precyzji o ponad 10% w stosunku do poprzedniej serii doświadczeń. Wysoka wartość precyzji przy dość niskim średnim wskaźniku dopasowania pokazuje, że podstawowym problemem zbudowania modelu klasyfikatora zastępującego pracę człowieka jest określenie liczby podkategorii, jaką ma przyporządkować model klasyfikatora.

V. Optymalizacja parametrów klasyfikatora Support Vector Machine

Podczas kilkunastu eksperymentów klasyfikator SVM z manualnie dobranymi parametrami wykazywał się nieznacznie większą – w porównaniu z pozostałymi modelami – średnią skutecznością klasyfikacji. Wynika

z tego, że warto optymalizować ustawienia SVM, jednak przy intuicyjnym sposobie dobierania parametrów nie zawsze uzyskuje się zadowalającą poprawę jakości. Kwestia znalezienia optymalnych parametrów jest problemem nieliniowym, a w rozważanym problemie klasyfikacji ankiet dodatkowo wiąże się z dużym nakładem obliczeniowym. Z uwagi na to, przeprowadzono doświadczenia polegające na optymalizacji parametrów klasyfikatora SVM przy użyciu metod ewolucyjnych. Do tego celu wykorzystano algorytm Differential Evolution (DE)[19]. Dla problemu klasyfikacji ankiet funkcja kosztu została oparta na średniej harmonicznej[20] parametrów dopasowania oraz precyzji, wyznaczonych na przestrzeni eksperymentów 5–17. Można to zaprezentować za pomocą następującego wzoru:

$$funkcja\ kosztu = 100 - \frac{2 * f_{śr\ 5-17} * p_{śr\ 5-17}}{f_{śr\ 5-17} + p_{śr\ 5-17}}$$

gdzie:
$f_{śr\,5-17}$ – średnia wartość dopasowania na przestrzeni eksperymentów 5–17;
$p_{śr\,5-17}$ – średnia wartość precyzji na przestrzeni eksperymentów 5–17.

Danymi wejściowymi dla algorytmu DE były wektory zbudowane z poszczególnych wartości parametrów klasyfikatora SVM. Przed uruchomieniem doświadczeń należało również wyznaczyć wartości minimalne i maksymalne dla wszystkich uwzględnionych parametrów. Dla każdego rodzaju funkcji jądra (*kernel function*) wykonano doświadczenia; najlepsze wyniki przedstawiono w tabeli 8, wraz z pierwotnie wykonanymi eksperymentami.

Tabela 8. Średnie dopasowanie i precyzja dla różnych ustawień parametrów klasyfikatora Support Vector Machine po ewaluacji algorytmem Differential Evolution

Ustawienia parametrów dla klasyfikatora SVM	Liczba przydzielonych klas	*Fit*	*Precision*
Wyniki eksperymentów przeprowadzonych pierwotnie			
Oryginalne ustawienia Jądro – wielomianowe Eksponent – 1 Parametr C – 1	Jedna	28,12	68,63
Ustawienia manualne Jądro – gaussowskie RBF Gamma – 0,01 Parametr C – 21	Jedna	29,05	69,4
Wyniki eksperymentów przeprowadzonych po optymalizacji algorytmem DE			
Jądro – wielomianowe Eksponent – 1,3815622762154418 Parametr C – 0,158598981557647	Jedna	28,84	69,16
Jądro – gaussowskie RBF Gamma – 3,48013818145419E-4 Parametr C – 148,40666325867156	Jedna	29,6	71,13

Źródło: opracowanie własne autorów

Wykonanie doświadczeń przy użyciu algorytmu DE w celu optymalizacji parametrów klasyfikatora SVM pozwoliło w przypadku jądra gaussowskiego na zwiększenie średniej wartości dopasowania o 1,48 punktu procentowego oraz precyzji o 2,5 punktu procentowego w stosunku do oryginalnych ustawień. W porównaniu

[19] Zagadnienia teoretyczne wyjaśniono w dodatku.
[20] Średnią harmoniczną nazywamy odwrotność średniej arytmetycznej odwrotności składowych liczb.

z ustawieniami manualnymi wartości te wzrosły o kolejne 0,55 i 1,73 punktu procentowego. Dodatkowo widoczna jest wyższość jądra gaussowskiego względem jądra wielomianowego. Uzyskanie nieznacznie wyższych wskaźników jakości klasyfikacji oznacza, że w niektórych przypadkach warto jest zadbać o dopasowanie jak najlepszych parametrów dla klasyfikatora SVM.

VI. Podsumowanie

Szczegółowe wyniki ilościowe i jakościowe badania ankietowego zostały omówione w tomie pierwszym. Spośród ośmiu tysięcy osób związanych z polską nauką, które wypełniły kwestionariusz, w pytaniu otwartym wypowiedziało się ponad 2,5 tysiąca z nich (32%). Analizę jakościową tego pytania wspomagały metody uczenia maszynowego. Uwagi zgrupowano w pięć głównych kategorii: „proces recenzowania jako całość", „ocena", „jakość pracy", „formalizm" i „anonimowość". Dalej kategorie te dzieliły się na podkategorie, co pokazuje tabela 1.

Automatyczne przyporządkowanie wypowiedzi respondentów do podkategorii było złożonym zadaniem, ponieważ niektóre wypowiedzi mogły należeć do wielu klas jednocześnie. Najwyższą jakość klasyfikacji otrzymywano, gdy wypowiedź należała tylko do jednej podkategorii. W rezultacie nie udało się uzyskać na tyle wysokich wskaźników miar dopasowania i precyzji, by całkowicie zastąpić pracę eksperta. Wykonanie automatycznej klasyfikacji wspomogło jednak manualny proces przydzielania podkategorii wypowiedziom.

Doświadczenia wykonane w ramach klasyfikacji ankiet pozwoliły na wyciągnięcie kilku interesujących wniosków:

- Do klasyfikacji wolnych wypowiedzi, czyli danych tekstowych, co do których nie ma pewności użycia poprawnej formy słownej, warto zastosować metody korekty tekstu.
- Wykorzystanie modelu klasyfikacji z organizacją hierarchiczną nie zawsze musi poprawiać jakość klasyfikacji.
- Klasyfikator Support Vector Machine jest lepszy od klasyfikatora Multinomial Naive Bayes w przypadkach bardziej złożonych, gdy słownictwo dla różnych klas decyzyjnych jest podobne.
- Optymalizacja parametrów klasyfikatora SVM przy wykorzystaniu algorytmów genetycznych poprawia wskaźniki jakości klasyfikacji.

Wyniki klasyfikacji są następujące: ponad 30% uwag dotyczyło całości procesu recenzowania, niecałe 30% – oceny, ponad 20% – jakości pracy recenzentów, a jedynie 10% – formalizmu i anonimowości.

VII. Bibliografia

Manning C., Schutze H., Prabhakar R., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

Zespół laboratorium inteligentnych systemów informatycznych Ośrodka Przetwarzania Informacji – Instytutu Badawczego, *Analiza porównawcza metod doboru recenzentów*, raport niepublikowany, OPI, Warszawa 2011.

Zespół laboratorium inteligentnych systemów informatycznych Ośrodka Przetwarzania Informacji – Instytutu Badawczego, *Analiza porównawcza narzędzi informatycznych wspomagających dobór recenzentów*, raport niepublikowany, OPI, Warszawa 2011.

Zespół laboratorium inteligentnych systemów informatycznych Ośrodka Przetwarzania Informacji – Instytutu Badawczego, *Projekt systemu*, niepublikowany, OPI, Warszawa 2011.

Zespół laboratorium inteligentnych systemów informatycznych Ośrodka Przetwarzania Informacji – Instytutu Badawczego, *System wspomagania wyboru recenzentów (prototyp)*, raport niepublikowany, OPI, Warszawa, 2011.

Źródła internetowe

Dane systemu OSF, kwiecień 2011, https://osf.opi.org.pl.

Laboratorium inteligentnych systemów informatycznych Ośrodka Przetwarzania Informacji – Instytutu Badawczego, http://lis.opi.org.pl, dostęp 07.08.2012.

Morfologik, http://morfologik.blogspot.com, dostęp 07.08.2012.

Słownik języka polskiego, http://www.sjp.pl, dostęp 07.08.2012.

Rozdział drugi

WYBRANE SYSTEMY INFORMATYCZNE DOBORU RECENZENTÓW

(Jarosław Protasiewicz, Sławomir Dadas, Małgorzata Gałężewska, Tomasz Stanisławek, Marek Kozłowski, Jan Artysiewicz, Agata Kopacz)

I. Przykłady produktów komercyjnych

1. Elsevier Editorial System

Elsevier Editorial System (EES) wydawnictwa Elsevier wykorzystywany jest do zarządzania procesem recenzowania. Pozwala autorom na złożenie swoich manuskryptów, recenzentom na ich ocenianie, a edytorom na zarządzanie on-line całym procesem, zapewniając jego płynny przebieg od momentu zgłoszenia artykułu do publikacji. System został uruchomiony w 2002 roku, a obecnie liczy 3,5 miliona użytkowników zarejestrowanych i ponad dwa miliony użytkowników aktywnych. EES składa się z trzech modułów: autora publikacji, recenzenta i edytora[21].

1.1. Moduł autora publikacji

Baza zawiera około 2000 czasopism z różnych dziedzin nauki. Dzięki modułowi autora naukowiec może odnaleźć magazyn, w którym chce opublikować pracę, a także uzyskać wskazówki dotyczące jej przygotowania. Wymagania każdego czasopisma należącego do Elsevier ujęte są w *Guide for Author* umieszczonym na stronach internetowych. Przewodnik podaje informacje o takich zagadnieniach, jak[22]: etyka, konflikt interesów, prawa autorskie, struktura artykułu, abstrakt, słowa kluczowe *etc.* Zgłoszenie publikacji przez autora następuje po wybraniu czasopisma i opracowaniu manuskryptu według odpowiednich wymogów. Artykuł zgłaszany jest elektronicznie za pomocą systemu EES. Wszystkie prace konwertowane są do formatu PDF i w takiej formie wykorzystywane w procesie recenzowania. Autor zyskuje możliwość monitorowania statusu swojego manuskryptu oraz uzyskuje dostęp do recenzji.

1.2. Moduł recenzenta

EES wspomaga pracę recenzenta poprzez zapewnienie trzydziestodniowego, nielimitowanego dostępu do serwisu Scopus[23], dzięki czemu oceniający może przeczytać prace, do których odwołuje się autor. Scopus to największa baza abstraktów i cytowań zrecenzowanych publikacji; posiada około 19 tysięcy czasopism i książek od pięciu tysięcy wydawców. Spośród 45,5 milionów rekordów 70% zawiera abstrakty. Recenzent ma dostęp do pakietów szkoleniowych oraz listy najczęściej zadawanych pytań. Dodatkowo może sprawdzić ostateczny termin wykonania recenzji oraz śledzić status swojej recenzji[24].

1.3. Moduł edytora

Podobnie jak recenzenci, edytorzy otrzymują dostęp do bazy Scopus. Istnieje szansa takiego zaprojektowania i konfiguracji systemu EES, aby pasował do obiegu dokumentów w redakcji edytora. Dostępne są funkcjonalności przekazywania instrukcji dla autorów i recenzentów, zapraszania recenzentów oraz wysyłania przy-

[21] Elsevier Editorial System, http://www.elsevier.com/wps/find/authors.authors/EES-author, dostęp 08.08.2012.
[22] Elsevier, moduł autora publikacji, http://www.elsevier.com/wps/find/authorsview.authors/landing_main, dostęp 08.08.2012.
[23] Scopus, *About* Scopus, http://www.info.sciverse.com/scopus/about, dostęp 08.08.2012.
[24] Elsevier, moduł recenzenta, http://www.elsevier.com/wps/find/reviewershome.reviewers/ees, dostęp 08.08.2012.

pomnień przez spersonalizowaną korespondencję e-mailową. Moduł wspiera edytorów w obsłudze procesu recenzowania poprzez:

- zastosowanie odpowiedniej klasyfikacji recenzentów, co pomaga dobrać recenzentów do konkretnej publikacji;
- generowanie raportów zawierających wcześniejsze dokumenty recenzentów;
- wyróżnianie recenzentów, którzy jeszcze nie odpowiedzieli na zapytania;
- zawieranie odniesienia do historii prac i cytowań recenzentów;
- udostępnianie ponad 50 interaktywnych kursów on-line, pakietów szkoleniowych oraz listy najczęściej zadawanych pytań[25].

2. ScholarOne

Firma ScholarOne oferuje webową aplikację ScholarOne Manuscripts[26], która umożliwia składanie publikacji, przeprowadzanie procesu recenzji oraz monitorowanie każdego z jego etapów. System ma 13 milionów zarejestrowanych użytkowników, zawiera zaimplementowane procesy obiegu dokumentów specyficzne dla ponad 3400 czasopism. Oferuje następujące wparcie procesu recenzowania:

- system detekcji plagiatów;
- konwersja publikacji do formatów HTML i PDF;
- automatyczne odwołania do cytowań w bazie PubMed;
- wielojęzyczny interfejs użytkownika;
- możliwość wgrywania plików multimedialnych;
- algorytm sugerujący recenzentów;
- przeszukiwanie baz PubMed, HighWire Press i Google.

Firma ScholarOne zainwestowała w takie elementy infrastruktury, jak baza danych IBM DB2[27], BEA WebLogic[28] i Crystal Decisions[29]. Wszystkie dane przechowywane są w Eagan Data Center zlokalizowanym w Minnesocie w Stanach Zjednoczonych.

II. Przykłady produktów bezpłatnych

1. Repository-Centric Peer-Review Model

Model zbudowany jest z centralnego repozytorium, wykorzystującego do przechowywania publikacji naukowych protokół Open Archives Initiative (OAI). Prace te pobierane są z repozytoriów instytucji, które udostępniają swoje artykuły. Dzięki bazie publikacji zbudowanej w głównym repozytorium możliwe jest stworzenie automatycznych metod wyboru recenzentów. Oceniający korzystają z interfejsu webowego, gdzie mogą zgłaszać recenzje. Na ich podstawie generowane są metadane, które służą jako informacje zwrotne dla dostawców publikacji[30].

W celu określenia relacji między poszczególnymi autorami konstruuje się graf, którego wierzchołki reprezentują poszczególnych autorów, a krawędzie – relacje współautorstwa. Graf ten jest skierowany. Jeżeli dwóch autorów posiada wspólne publikacje, to występują dwie krawędzie: pierwsza – między autorem pierwszym a drugim, druga – między autorem drugim a pierwszym. Każdej krawędzi przypisana jest waga związana z liczbą wspólnych publikacji między autorami. Każda publikacja zwiększa całkowitą wagę krawędzi o pewną liczbę od 0 do 1. W publikacjach z dużą liczbą autorów wpływ danej pracy na całkowitą wagę będzie niewielki, natomiast publikacja z najmniejszą liczbą autorów będzie na nią oddziaływać w większym stopniu. Wpływ ten dla krawędzi współautorstwa twórców i oraz j liczony jest ze wzoru:

[25] Elsevier, moduł edytora, http://www.elsevier.com/wps/find/editorshome.editors/onlinesubmission, dostęp 08.08.2012.
[26] ScholarOne, *The online manuscript submission and peer review* process, http://scholarone.com/media/manuscripts_fs.pdf, dostęp 08.08.2012.
[27] IBM DB2, http://www-01.ibm.com/software/data/db2, dostęp 08.08.2012.
[28] BEA WebLogic, http://download.oracle.com/docs/cd/E13222_01/wls/docs100/index.html, dostęp 08.08.2012.
[29] Crystal Reports, *Crystal Decisions*, http://www.crystalreports.com, dostęp 08.08.2012.
[30] Rodriguez M.A., Bollen J., van de Sompel H., *The convergence of digital-libraries and the peer-review process*, „Journal of Information Science", 32(2), 149–159, 2006.

$$m_{i,j} = \frac{1}{x - 1}$$

gdzie:
m – wpływ publikacji;
x – liczba autorów publikacji;

Następnie waga liczona jest poprzez sumowanie wpływów wszystkich publikacji, w których występują wspólni autorzy:

$$w_{i,j} = \sum_{j=0}^{|M|} m_{i,j}$$

gdzie:
m – wpływ publikacji;
M – zbiór publikacji, które napisali wspólnie autorzy i oraz j;
w – waga publikacji.

Ostatecznie wartość wagi normalizowana jest przez podzielenie jej przez sumę wszystkich wag krawędzi wychodzących dla danego autora, gdzie indeksem $i, j_{(t+1)}$ oznaczono wagę znormalizowaną, a $i, j_{(t)}$ wagę nieznormalizowaną:

$$w_{i,j\,(t+1)} = \frac{w_{i,j\,(t)}}{\sum_{x=0}^{|out\,(n_i)|} w_{i,x\,(t)}}$$

gdzie:
$out(n_i)$ – zbiór wszystkich krawędzi wychodzących do węzła i;
w – waga publikacji;
x – iterator.

Gdy sieć współautorstwa zostanie zbudowana, na jej podstawie można przyporządkować publikacjom potencjalnych recenzentów. Polega to na doborze najbardziej odpowiednich naukowców na podstawie odwołań bibliograficznych znajdujących się w publikacjach. Jeżeli zgłaszana praca zawiera bibliografię zapisaną w formacie OAI, to możliwe jest bezpośrednie wyodrębnienie cytowań. W przeciwnym razie treść publikacji musi być przetworzona przez narzędzie służące do ekstrakcji cytowań, np. Open Citation Project (OpCit).

Poszukiwanie recenzentów dla publikacji polega na zastosowaniu metody roju cząstek (*particle swarm*). Metoda opiera się na rozprzestrzenianiu się cząstek w sieci współautorstwa. Każda cząstka p posiada pewną przypisaną wartość energii ε, gdzie $p(\varepsilon) \in [0,1]$. Kiedy wierzchołek w grafie jest odwiedzony przez cząstkę, do jego wewnętrznej pamięci dodawana jest wartość energii tej cząstki $n(\varepsilon)_{t+1} = n(\varepsilon)_t + p(\varepsilon)_t$,

gdzie:
$n(\varepsilon) \in R$;
$n(\varepsilon)$ – pamięć wierzchołka;
t – dyskretny punkt w czasie.

W każdym kroku algorytmu, kiedy cząstka przechodzi pomiędzy wierzchołkami, jej energia obniżana jest proporcjonalnie do pewnego zdefiniowanego współczynnika *ds*: $p(\varepsilon)_{t+1} = p(\varepsilon)_t - (p(\varepsilon)_t * ds)$, gdzie ds $\in [0,1]$.

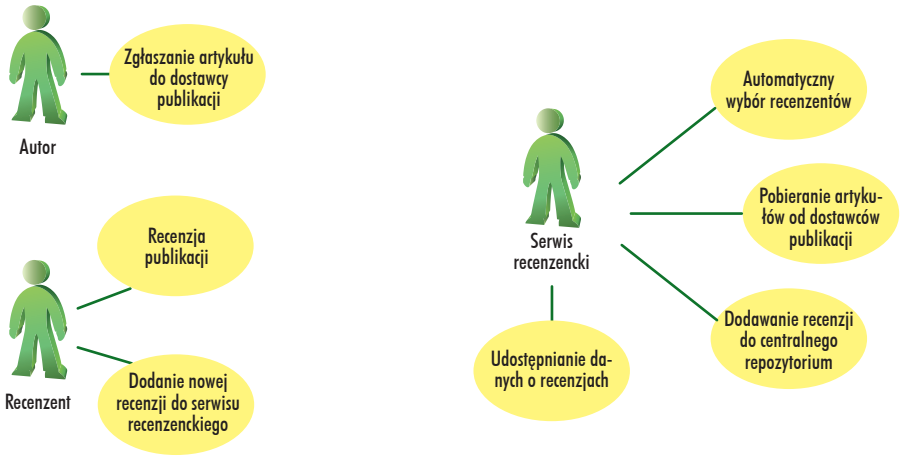
Algorytm rozpoczyna się od przekazania każdemu wierzchołkowi reprezentującemu autora cytowanego przez analizowaną publikację dziesięciu cząstek energii równej 1. Cząstki propagowane są na kolejne wierzchołki zgodnie z prawdopodobieństwami przypisanymi tym wierzchołkom. Algorytm kończy się, kiedy energia wszystkich cząstek obniży się do wartości 0. Jeżeli w pamięci wierzchołka znajduje się niezerowa wartość energii, to dany wierzchołek może być reprezentowany przez potencjalnego recenzenta publikacji. Wartość energii można interpretować jako poziom dopasowania recenzenta do danej publikacji. W ostatnim etapie algorytmu, wszystkie wartości energii normalizowane są do zakresu [0,1] z użyciem wzoru[31]:

$$n(\varepsilon)_{t+1} = \frac{n(\varepsilon)_t}{\sum_{x=0}^{|N|} n_x(\varepsilon)_t}$$

gdzie:
$n(\varepsilon)$ – pamięć wierzchołka;
N – zbiór wszystkich wierzchołków w grafie;
t – dyskretny punkt w czasie;
x – iterator po zbiorze wierzchołków w grafie.

Role i czynności użytkowników systemu zobrazowano na rysunku 1, a kolejne etapy oceny wniosków na rysunku 2.

Rysunek 1. Role uczestników procesu recenzji w Repository-Centric Peer-Review Model



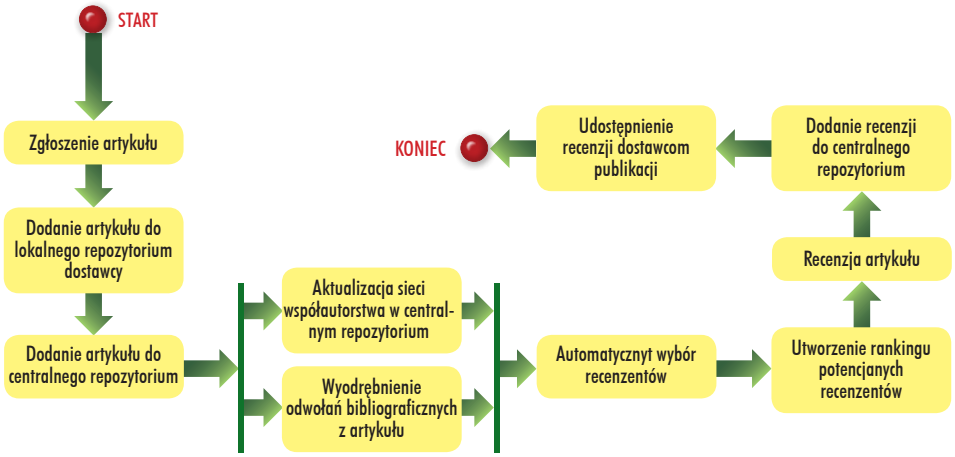
Źródło: opracowanie własne autorów

2. Conference on Knowledge Discovery and Data Mining

Na potrzeby międzynarodowej konferencji poświęconej *knowledge discovery and data mining* powstał system informatyczny wspomagający dobór recenzentów do oceny abstraktów konferencyjnych[32]. W implementacji systemu wykorzystano wiele języków programowania, środowisk programistycznych i platform: Java, C++, Prolog, Matlab,.NET Framework. Dokumentacja oraz kod aplikacji zostały udostępnione na licencji *open source* w serwisie Google Code[33]. Autorzy pozwalają na personalizację oprogramowania, wymagając przy tym odpowiedniego udokumentowania ustawień oraz zapewnienia wsparcia dla tzw. *mash-ups*[34] przez takie narzędzia jak na przykład Yahoo! Pipes[35].

[31] Ibidem.
[32] Flach P.A., Spiegler S., Golenia B., Price S., Guiver J., Herbrich R., Graepel T., Zaki M.J., *Novel tools to streamline the conference review process: Experiences from SIGKDD'09*, „SIGKDD Explorations", 11(2), 63–67, 2009.
[33] Google Code, http://code.google.com, dostęp 08.08.2012.
[34] Mash-up to strona internetowa, która łączy w sobie aplikacje on-line z różnych źródeł.
[35] Yahoo! Pipes to aplikacja pozwalająca na łączenie treści z wielu źródeł internetowych i publikowanie tych treści jako samodzielnych aplikacji webowych, http://pipes.yahoo.com, dostęp 08.08.2012.

Rysunek 2. Etapy oceny wniosków w Repository-Centric Peer-Review Model



Źródło: opracowanie własne autorów

Do komitetu organizacyjnego International Conference on Knowledge Discovery and Data Mining 2009 (SIGKDD'09) wpłynęło 537 zgłoszeń od 199 badaczy. W celu efektywnej oceny abstraktów użyto technik sztucznej inteligencji oraz metod *text mining*, dzięki którym przyspieszono trzy najważniejsze etapy doboru recenzentów: dopasowanie abstraktów, przypisanie naukowców do konkretnych prac oraz ocenę uzyskanych rezultatów. Zamiarem autorów było stworzenie systemu, który już na pierwszym etapie pozwalałby wybrać najodpowiedniejsze dla każdego recenzenta abstrakty. W dalszej fazie procesu każdy oceniający mógł samodzielnie dokonać korekty (przyjąć lub odrzucić zaproponowany tekst), zasługą twórców systemu jest jednak dokładne wyselekcjonowanie prac dla poszczególnych recenzentów[36].

2.1. Wstępny etap dopasowania abstraktów

Pierwszym krokiem tworzenia systemu było obliczenie dwóch wskaźników określających stopień dopasowania abstraktu do konkretnego recenzenta. Pierwszy wskaźnik określa podobieństwo między obszarem specjalizacji recenzenta a zgłoszonym abstraktem, które liczono porównując słowa pojawiające się w abstraktach ze słowami występującymi w tytułach artykułów danego recenzenta. Dla każdej pary recenzent – abstrakt ustalono wektory słów im odpowiadających. Następnie wektory te zostały znormalizowane do postaci TF-IDF (*term frequency – inverse document frequency*) i na ich podstawie powstał ranking abstraktów – propozycji przedstawianych każdemu oceniającemu. Wektory porównano ze sobą wykorzystując miarę podobieństwa cosinusowego (*cosine similarity*), która odpowiada kątowi pomiędzy wektorem abstraktu a wektorem recenzenta.

Ponadto, dla każdego recenzenta i każdego abstraktu ustalono drugi wskaźnik podobieństwa, który odpowiadał liczbie dziedzin wspólnych dla recenzenta i abstraktu. Każdy ekspert i wnioskodawca wybierał – z listy 65 dziedzin – obszary nauki właściwe dla siebie. Spis powstał na podstawie tytułów sesji, które pojawiły się podczas poprzednich konferencji na przestrzeni ostatniej dekady.

Ostatecznie miarę podobieństwa liczono poprzez zsumowanie tych dwóch wartości: miary wspólnego słownictwa i liczby pokrywających się dziedzin. Aby uzyskać optymalny rozkład wstępnych ofert dla recenzentów, wartość miary słownictwa przemnożono przez wartość $\alpha = 15$, a od wyniku odjęto 1. Dla recenzenta pc_i oraz zgłoszonego abstraktu a_j, łączna miara podobieństwa między nimi była liczona zgodnie ze wzorem:

$$s(pc_i, a_j)=\alpha * s_1(pc_i, a_j) + s_2(pc_i, a_j) - 1$$

[36] Flach P.A. et al., op.cit.

gdzie:
i – indeks recenzenta;
j – indeks abstraktu;
s_1 – podobieństwo cosinusowe między tekstem abstraktu i tekstami przypisanymi do recenzenta;
s_2 – liczba wspólnych dziedzin dla recenzenta i abstraktu;
α – waga miary s_1.

Oryginalne wartości miar podobieństwa $s(pc_i, a_j)$ przekształcono na zmienną o trzech wartościach. Określała ona stopień prawdopodobieństwa, z jakim recenzent gotowy będzie podjąć się recenzji danego abstraktu:

- wartość 1 – „jeśli zajdzie konieczność" (*in a pinch*);
- wartość 2 – „chętnie" (*willing*);
- wartość 3 – „z wielką chęcią" (*eager*).

Autorzy ilustrują to następującym przykładem: aby wstępnie przyporządkowany abstrakt zainteresował recenzenta (wartość 3), musiał on spełniać następujące reguły:

- cztery dziedziny wspólne dla recenzenta i abstraktu;
- trzy wspólne dziedziny i podobieństwo $s_1(pc_i, a_j)$ na poziomie przynajmniej 0,067 lub dwie dziedziny wspólne i podobieństwo $s_1(pc_i, a_j)$ nie mniejsze niż 0,133.

Logując się do systemu informatycznego, recenzent otrzymywał przedwstępnie wyselekcjonowane abstrakty – propozycje. Średnio każdemu oceniającemu przypisywano 7,5 abstraktów na poziomie trzecim, 16,6 na poziomie drugim oraz 52 na poziomie pierwszym. Pomimo wyjściowych ustaleń recenzentowi dano szansę skorygowania początkowo przyznanych abstraktów – mógł zmienić miejsce danego abstraktu w rankingu i ocenić go jako mniej lub bardziej godny uwagi niż początkowo. Autorzy porównali wybrane i posortowane przez system wstępne propozycje z propozycjami skorygowanymi przez recenzentów i obliczyli dwie miary – *precision* i *recall* oraz wartość współczynnika F[37].

2.2. Wybór ostatecznych propozycji recenzji

W kolejnym etapie wybrane abstrakty wysłano do recenzentów, z uwzględnieniem ich sugestii dotyczących wstępnych propozycji. Problem przydzielenia artykułów do konkretnych osób został sformułowany w postaci zadania całkowitoliczbowego programowania liniowego, gdzie r jest liczbą potencjalnych recenzentów, a p – liczbą wszystkich publikacji, które mają zostać oddane do recenzji. W poprzednim kroku algorytmu każdemu z recenzentów przypisano miarę określającą jego podobieństwo do każdej z publikacji w postaci liczby całkowitej od 1 do 3, gdzie 3 oznacza największe podobieństwo. Na podstawie tych podobieństw zdefiniowana została macierz przydziału B^{r*p}, określająca preferencje recenzentów wobec publikacji. Jeżeli dana publikacja j nie została przypisana do recenzenta i, wartość B_{ij} równa jest 0. W przeciwnym razie wartość odpowiadającego elementu macierzy równa się obliczonej mierze podobieństwa powiększonej o 2 (autorzy dodają stałą do elementów niezerowych, aby zminimalizować prawdopodobieństwo przydzielenia recenzentowi publikacji o zerowym podobieństwie).

Dodatkowo, recenzenci mieli możliwość określenia konfliktu interesów dla konkretnych publikacji. Na podstawie tych danych zdefiniowano macierz konfliktu interesów $C^{r\times p}$. Jest to macierz binarna, z elementami równymi 1 w miejscach, gdzie występuje konflikt interesów. Autorzy założyli również, że do każdej publikacji powinno zostać przydzielonych dokładnie r_r recenzentów (gdzie $r_r = 3$), natomiast żaden z recenzentów nie powinien recenzować więcej niż r_p publikacji (gdzie $r_p \approx 8$). Aby możliwe było określenie funkcji celu, zdefiniowana została jeszcze jedna macierz binarna – $A^{r\times p}$. Wartość 1 w elemencie a_{ij} oznacza, że autor i został przydzielony do recenzowania publikacji j. Funkcja celu jest definiowana jako:

$$\max : \sum_{i=1}^{r} \sum_{j=1}^{p} B_{ij} \cdot A_{ij}$$

[37] Zagadnienia teoretyczne wyjaśniono w dodatku.

Ograniczenia w modelu są definiowane następująco:

- ograniczenia ze względu na ilość recenzentów na publikację:

$$\sum_{i=1}^{r} A_{ij} = r_r, \forall 1 \leq j \leq p$$

- ograniczenia ze względu na maksymalną liczbę publikacji przypisanych do jednego recenzenta:

$$r_p \leq \sum_{i=1}^{r} A_{ij} \leq r_p, \forall 1 \leq i \leq r$$

- ograniczenia ze względu na występujące konflikty:

$$A_{ij} = 0 \quad jeżeli \quad C_{ij} = 0$$

Dla tak zdefiniowanego modelu autorom udało się znaleźć rozwiązanie, w którym 94,3% przypisanych par recenzent – publikacja miało drugi (*willing* –„chętnie") lub trzeci (*eager* – „z wielką chęcią") poziom podobieństwa.

Po przyporządkowaniu sugerowanych przez system abstraktów, w kilku przypadkach należało dokonać manualnego przypisania artykułów do sędziów (*manual tweaking*). Spisano listę abstraktów, w której do każdego abstraktu dopasowano dziesięciu recenzentów z najwyższymi miarami podobieństwa $s(pc_i, a_j)$; abstrakty najmniej podobne znalazły się na górze listy. Potem na tej podstawie wybrano recenzentów[38].

2.3. Ocena trafności systemu

Ostatnią częścią procesu wyboru publikacji było określenie ostatecznej oceny, decydującej o odrzuceniu lub zaakceptowaniu artykułu. Opierając się na otrzymanych recenzjach, autorzy opracowali model kalibracji ocen w nich zawartych, wykorzystujący wiedzę na temat różnic w ocenach danej publikacji przez poszczególnych recenzentów oraz ich doświadczenie w określonej dziedzinie. Zastosowana metoda wykorzystywała generatywny model probabilistyczny ze zbiorem znanych zmiennych:
R – *reviews* (recenzje);
S – *submissions* (wnioski);
E – *expertise level* (wiedza recenzentów);
J – *reviewers/judges* (recenzenci/sędziowie)

oraz zmiennych ukrytych:
$q_s \sim N(\mu_s, v_s)$ – *quality* (jakość) dla każdej publikacji ze zbioru S;
$\lambda_e \sim G(k_\lambda, \beta_\lambda)$ – *precision* (precyzja) dla każdego poziomu wiedzy z E;
$a_j \sim G(k_j, \beta_j)$ – *accuracy* (trafność) dla każdego recenzenta j ze zbioru J.

Celem modelu było określenie wartości zmiennych ukrytych. Otrzymane wyniki nie zostały bezpośrednio wykorzystane do podjęcia decyzji o zaakceptowaniu bądź odrzuceniu publikacji, natomiast zastosowano je jako pomoc przy podejmowaniu tych decyzji oraz wskazaniu obszarów, gdzie konieczna okazała się weryfikacja ocen[39].

III. Systemy informatyczne Ośrodka Przetwarzania Informacji – Instytutu Badawczego

Analizę metod doboru recenzentów i procesu recenzowania na podstawie rozwiązań stosowanych w Ośrodku Przetwarzania Informacji przedstawiono w tomie pierwszym. Badaniu poddano następujące fundusze: Obsługa Strumieni Finansowania Nauki w Ministerstwie Nauki i Szkolnictwa Wyższego (OSF-MNiSW) oraz

[38] Ibidem.
[39] Ibidem.

w Narodowym Centrum Badań i Rozwoju (OSF-NCBiR), Program Operacyjny Innowacyjna Gospodarka (PO IG), Polsko-Norweski Fundusz Badań Naukowych (PN FBN) i Polsko-Szwajcarski Program Badawczy (PSPB). W tym miejscu zaprezentowana zostanie analiza systemów informatycznych implementujących procesy recenzowania i doboru recenzentów dla wymienionych funduszy i programów badawczych. Technicznie, wszystkie informatyczne systemy recenzowania (OSF[40], PN FBN[41], PSPB[42]) zostały zbudowane na bazie tego samego projektu informatycznego; dalej będziemy je krótko nazywać OSF. Wyjątkiem jest system recenzowania dla Programu Operacyjnego Innowacyjna Gospodarka, który wykonano jako część systemu LSI (Lokalny System Informatyczny); dalej będziemy posługiwać się skrótem LSI-PO IG[43].

1. Model logiczny

Systemy OSF i LSI-PO IG są aplikacjami o budowie modułowej – każdy moduł pełni określone funkcje. Niezależnie od funduszu grantowego wyróżnić można następujące części:

- edycja i składanie wniosku – dla wnioskodawcy;
- przetwarzanie wniosku – dla operatora funduszu;
- recenzowanie – dla recenzenta;
- baza recenzentów wraz z procedurami ich naboru do bazy i doboru do wniosku grantowego;
- zawieranie umów (opcjonalnie).

Przebieg składania i przetwarzania wniosków o granty, procedury recenzowania, formularze i metody doboru ekspertów różnią się w zależności od funduszu. Każdy z funduszy posiada dedykowany tylko jemu zestaw powyższych modułów, generalna zasada działania systemu jest jednak niezmienna. Najpierw – w odpowiednim module – wnioskodawcy rejestrują się i wypełniają aplikację. Następnie pracownicy merytoryczni operatora funduszu formalnie oceniają wniosek, wskazują recenzentów, obsługują proces recenzowania oraz – na podstawie recenzji – przygotowują ranking wniosków; pomaga im w tym baza recenzentów. Oceniający wypełniają elektroniczny formularz recenzji.

2. Technologia analizy, projektowania i wykonania

Ponieważ na początku nie istniała kompletna specyfikacja systemów, powstawały one iteracyjnie. Ten etap polegał na wykonaniu analizy, projektu, implementacji, a także wdrożeniu pewnego wycinka funkcjonalności, który mógł być całym modułem systemu bądź tylko częścią modułu. Potem przystępowano do weryfikacji założeń lub projektowania następnego elementu (modułu). Zastosowana technologia musiała zapewnić podatność na ciągłe modyfikacje. W celu organizacyjnego sprostania takim wyzwaniom, analiza wymagań i projektowanie odbywały się w języku UML[44], natomiast projektanci i programiści korzystali z takich narzędzi, jak:

- Bugzilla[45] jako repozytorium błędów i zadań;
- Enterprise Architect[46] jako narzędzie pracy projektantów i repozytorium projektów UML;
- Java/J2EE[47] jako języki programowania oraz Eclipse[48] i InteliJ[49] jako środowiska pracy programistów;
- baza danych Oracle[50] i język PL/SQL[51] oraz PL/SQL Developer[52] jako środowiska pracy programistów;
- Subversion[53] jako repozytorium kodu źródłowego Java oraz PL/SQL.

[40] System OSF jest dostępny pod adresem https://osf.opi.org.pl, dostęp 08.08.2012.
[41] System PNRF jest dostępny pod adresem https://pnrf.opi.org.pl, dostęp 08.08.2012.
[42] System PSRP (Polish-Swiss Reasearch Programme, po polsku: PSPB – Polsko-Szwajcarski Program Badawczy) jest dostępny pod adresem https://psrp-system.opi.org.pl, dostęp 08.08.2012.
[43] System LSI-PO IG jest dostępny pod adresem https://poig-wnioski.opi.org.pl, dostęp 08.08.2012.
[44] Unified Modeling Language, http://www.uml.org, dostęp 08.08.2012.
[45] Bugzilla, http://www.bugzilla.org, dostęp 08.08.2012.
[46] Enterprise Architect, http://www.sparxsystems.com.au, dostęp 08.08.2012.
[47] Java Platform, Enterprise Edition, http://www.java.com, dostęp 08.08.2012.
[48] Eclipse, http://www.eclipse.org, dostęp 08.08.2012.
[49] InteliJ, https://www.jetbrains.com/idea, dostęp 08.08.2012.
[50] Oracle, http://www.oracle.com, dostęp 08.08.2012.
[51] PL SQL to proceduralny język SQL.
[52] PL/SQL Developer, http://www.allroundautomations.nl/plsqldev.html, dostęp 08.08.2012.
[53] Subversion, http://subversion.tigris.org, dostęp 08.08.2012.

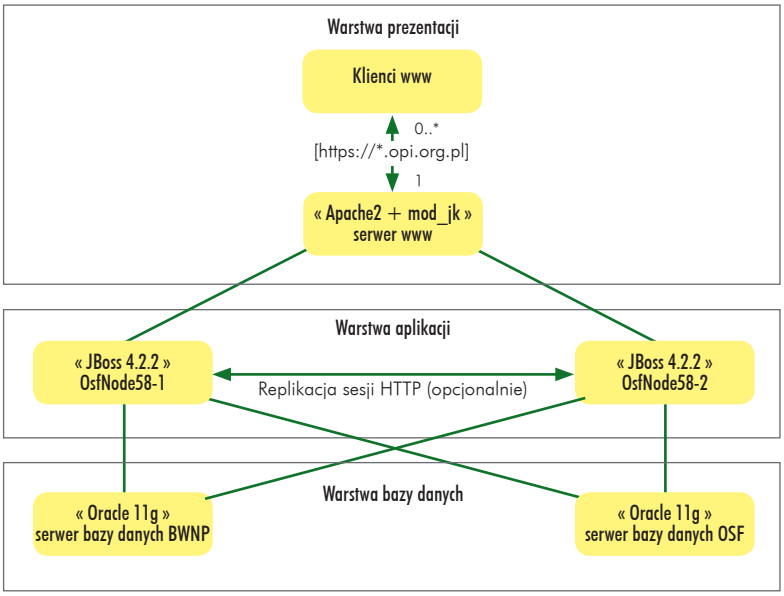
3. Architektura systemów

Uproszczony schemat architektury systemu OSF przedstawiono na rysunku 3. Architektura systemu LSI-PO IG jest analogiczna; wyróżniono trzy podstawowe warstwy:

1. **Warstwa bazy danych.** Oracle w wersji 11g to silnik bazy danych, który przechowuje dane i obsługuje operacje dodawania, modyfikowania danych oraz zapytań o dane. Zawiera procedury składowane PL/SQL wykonujące operacje na danych oraz zaawansowane, wymagające dużej wydajności przetwarzania danych.
2. **Warstwa aplikacji.** Serwery JBoss[54] w wersji 4.x pracują w klastrze, możliwe jest włączanie dodatkowych węzłów w miarę wzrostu obciążenia. Realizują procesy biznesowe, generują dynamicznie odpowiedzi na żądania klientów. Tutaj osadzone są główne komponenty systemu.
3. **Warstwa prezentacji.** Serwer Apache[55] w wersji 2.x rozkłada i stabilizuje obciążenie poszczególnych serwerów aplikacji (do tego celu używany jest moduł *mod_jk*), pełni też funkcję serwera statycznych plików.

Komunikacja z klientami odbywa się za pomocą szyfrowanego protokołu HTTPS[56]. Gwarantuje on, że nikt nie może przechwycić czy zmienić danych przesyłanych pomiędzy klientem a serwerem. Protokół jest uwierzytelniany podpisem cyfrowym SSL[57]. Wszystkie warstwy systemu pracują pod kontrolą systemów operacyjnych z rodziny Linux.

Rysunek 3. Uproszczony schemat architektury systemu OSF



Źródło: opracowanie własne autorów

IV. Podsumowanie

Przedstawione wyżej dwa komercyjne systemy, w szczególności ten wydawnictwa Elsevier, z całą pewnością nazwać można dojrzałymi produktami. Są to jednak rozwiązania dedykowane określonemu środowisku. Nie jest znany zakres prac, jaki należałoby wykonać, aby wdrożyć je w polskich realiach.

Na uwagę zasługują dwa systemy otwarte: Repository-Centric Peer-Review Model oraz narzędzie powstałe na potrzeby International Conference on Knowledge Discovery and Data Mining. Pierwszy z nich to baza

[54] JBoss, http://www.jboss.org, dostęp 08.08.2012.
[55] Apache, http://www.apache.org, dostęp 08.08.2012.
[56] Wikipedia, *HyperText Transfer Protocol* Secure, http://pl.wikipedia.org/wiki/HTTPS, dostęp 16.08.2012.
[57] Zastosowano certyfikat firmy Certum, http://www.certum.pl, dostęp 08.08.2012.

publikacji wykorzystująca grafy do opisu relacji między autorami, pomaga ona w doborze recenzentów za pomocą metody roju cząstek. Szczególnie ciekawy jest dobór recenzentów zastosowany podczas konferencji w 2009 roku – użyto metod analizy statystycznej tekstu i miary podobieństwa pomiędzy wektorem cech osoby i dokumentu do recenzji. Ten system z pewnością warto byłoby rozwijać, choć z drugiej strony znane jest tylko jedno jego wdrożenie, a ponadto nic nie wiadomo o jego skalowalności i możliwości integracji z innymi lokalnymi systemami.

Uzasadniona wydaje się budowa własnego systemu wspomagania wyboru recenzentów. Powody takiej rekomendacji łatwo wyjaśnić. Zachowana byłaby pełna kontrola stosowanych algorytmów oraz możliwość ciągłego doskonalenia systemu. Wreszcie, podczas gdy oszacowanie kosztów adaptacji w polskich realiach każdego z przedstawionych systemów wydaje się niewykonalne, to koszty budowy autorskiego rozwiązania można dokładnie obliczyć.

V. Bibliografia

Flach P.A., Spiegler S., Golenia B., Price S., Guiver J., Herbrich R., Graepel T., Zaki M.J., *Novel tools to streamline the conference review process: Experiences from SIGKDD'09*, „SIGKDD Explorations", 11(2), 63–67, 2009.

Rodriguez M.A., Bollen J., van de Sompel H., *The convergence of digital-libraries and the peer-review process*, „Journal of Information Science", 32(2), 149–159, 2006.

Źródła internetowe:

Apache, http://www.apache.org, dostęp 08.08.2012.

BEA WebLogic, http://download.oracle.com/docs/cd/E13222_01/wls/docs100/index.html, dostęp 08.08.2012.

Bugzilla, http://www.bugzilla.org, dostęp 08.08.2012.

Certum, http://www.certum.pl, dostęp 08.08.2012.

Crystal Reports, *Crystal Decisions*, http://www.crystalreports.com, dostęp 08.08.2012.

Eclipse, http://www.eclipse.org, dostęp 08.08.2012.

Enterprise Architect, http://www.sparxsystems.com.au, dostęp 08.08.2012.

Elsevier Editorial System, http://www.elsevier.com/wps/find/authors.authors/EES-author, dostęp 08.08.2012.

Elsevier, moduł autora publikacji, http://www.elsevier.com/wps/find/authorsview.authors/landing_main, dostęp 08.08.2012.

Elsevier, moduł edytora, http://www.elsevier.com/wps/find/editorshome.editors/onlinesubmission, dostęp 08.08.2012.

Elsevier, moduł recenzenta, http://www.elsevier.com/wps/find/reviewershome.reviewers/ees, dostęp 08.08.2012.

Google Code, http://code.google.com, dostęp 08.08.2012.

IBM DB2, http://www-01.ibm.com/software/data/db2, dostęp 08.08.2012.

InteliJ, https://www.jetbrains.com/idea, dostęp 08.08.2012.

Java Platform, Enterprise Edition, http://www.java.com, dostęp 08.08.2012.

JBoss, http://www.jboss.org, dostęp 08.08.2012.

Oracle, http://www.oracle.com, dostęp 08.08.2012.

PL/SQL Developer, http://www.allroundautomations.nl/plsqldev.html, dostęp 08.08.2012.

ScholarOne, *The online manuscript submission and perr review process*, http://scholarone.com/media/manuscripts_fs.pdf, dostęp 08.08.2012.

Scopus, *About Scopus*, http://www.info.sciverse.com/scopus/about, dostęp 08.08.2012.

Subversion, http://subversion.tigris.org, dostęp 08.08.2012.

System informatyczny OSF, https://osf.opi.org.pl, dostęp 08.08.2012.

System informatyczny PO IG, https://poig-wnioski.opi.org.pl, dostęp 08.08.2012.

System informatyczny PN FBN, https://pnrf.opi.org.pl, dostęp 08.08.2012.

System informatyczny PSPB, https://psrp-system.opi.org.pl, dostęp 08.08.2012.

Unified Modeling Language, http://www.uml.org, dostęp 08.08.2012.

Wikipedia, *HyperText Transfer Protocol* Secure, http://pl.wikipedia.org/wiki/HTTPS, dostęp 16.08.2012.

Yahoo! Pipes, http://pipes.yahoo.com, dostęp 08.08.2012.

Rozdział trzeci

PROJEKT SYSTEMU DOBORU RECENZENTÓW

(Jarosław Protasiewicz, Sławomir Dadas, Małgorzata Gałężewska, Tomasz Stanisławek, Marek Kozłowski, Jan Artysiewicz)

I. Koncepcja nowego systemu

1. Procesy biznesowe

W pierwszym etapie pracy nad systemem wspomagania wyboru recenzentów stworzony zostanie dynamiczny i adaptacyjny model gromadzenia, analizy i oceny ekspertów dla poszczególnych artykułów lub wniosków badawczych. Następnie model ten posłuży do utworzenia inteligentnej bazy gromadzącej informacje o potencjalnych recenzentach; na podstawie tych danych system będzie przedstawiać ranking oceniających dla zadanego problemu. Zapytanie do bazy będzie miało postać: „Wskaż recenzentów do oceny projektu, do oceny artykułu *etc.*". System odpowiadający na takie pytanie będzie musiał zbierać informacje o pracownikach naukowych, tworzyć wiedzę na ich temat oraz formować ranking recenzentów[58].

1.1. Zbieranie informacji o pracownikach naukowych

Zaangażowane będą roboty internetowe i lokalne, czyli programy gromadzące – w sieci lub lokalnych bazach danych – informacje o poszczególnych osobach. Dla każdego pracownika naukowego znajdującego się w Bazie Wiedzy o Nauce Polskiej (BWNP) wywoływany jest cyklicznie robot internetowy (jedna instancja robota równa się jednemu procesowi serwera aplikacji). Robot lokalny przeszukuje bazy OPI, natomiast robot internetowy – zasoby zewnętrzne. W pierwszej fazie projektu informacje zbierane są z baz artykułów, a następnie z sieci WWW.

Zebrana wiedza o pracowniku naukowym jest zapisywana w bazie danych. Wykorzystane zostaną następujące źródła:

- BWNP – dane o pracownikach naukowych;
- OSF – dane o recenzowaniu i klasyfikacji wniosków;
- WWW – przeszukiwane przy wykorzystaniu istniejących wyszukiwarek internetowych lub własnych mechanizmów;
- bazy artykułów – dostępne w internecie.

Ostatecznie pozyskane zostaną informacje o pracowniku naukowym, czyli lista jego artykułów naukowych zawierająca: tytuł artykułu, nazwę czasopisma, listę autorów, pozycję w kolejności autorów, streszczenie, słowa kluczowe, treść artykułu (opcjonalnie) oraz różne klasyfikacje dziedzin/dyscyplin pracownika naukowego.

1.2. Tworzenie wiedzy o pracowniku naukowym

Robot lokalny, czyli proces działający na serwerze aplikacji będzie odpowiedzialny za przetworzenie surowych informacji o naukowcu w wiedzę o nim. Źródłem danych są tekstowe informacje o danej osobie oraz BWNP i OSF. Przekształcenie informacji w wiedzę polega na wyodrębnieniu zbioru słów unikatowych dla każdego naukowca. Słowa będą wyodrębniane z artykułów naukowych (tytułów, słów kluczowych, abstraktów). Podję-

[58] *Projekt systemu*, opracowany w ramach projektu „System wspomagania wyboru recenzentów", niepublikowany, OPI, Warszawa 2011.

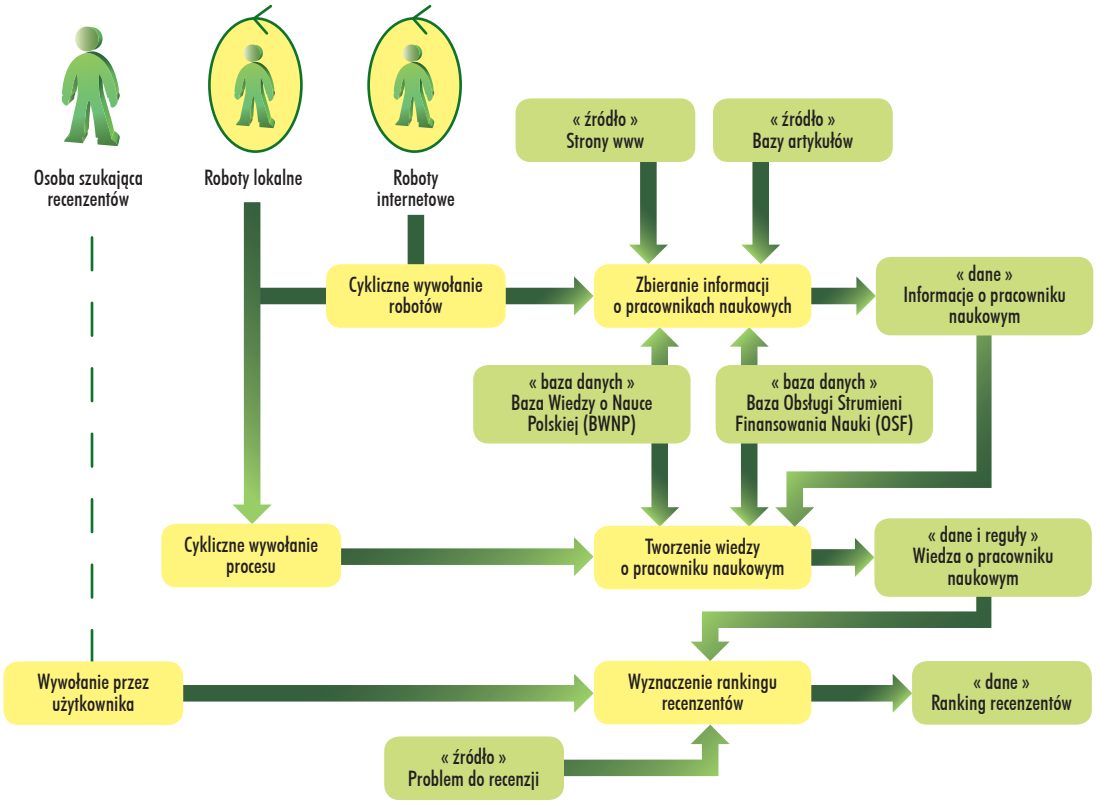
ta będzie też próba zbudowania zestawu konceptów oraz relacji między słowami (ontologii) dla pracownika naukowego.

1.3. Wyznaczenie rankingu recenzentów

Użytkownik systemu może zadać zapytanie do bazy wiedzy w dwojaki sposób: podać słowa kluczowe lub wskazać dokument, dla którego poszukuje recenzentów. W drugim przypadku słowa kluczowe z dokumentu wyodrębni odpowiedni algorytm. W efekcie pozyskana zostanie wiedza o problemie do recenzji, czyli zbiór słów kluczowych, które mogą być także powiązane regułami.

Do obliczenia rankingu recenzentów służy algorytm rankingowy, czyli program implementujący algorytmy dopasowania profili pracowników naukowych do dokumentu przeznaczonego do recenzji. Ranking może być utworzony poprzez znalezienie najlepszego dopasowania recenzowanego problemu (słów kluczowych z dokumentu) z wiedzą o pracownikach naukowych (słowami kluczowymi z profili uczonych), łącznie z wykorzystaniem ontologii. Należy zastanowić się, jak uwzględnić w zestawieniu dane o recenzjach wniosków z PN FBN, PSPB, MNiSW, NCBiR, PO IG. Ranking sygnalizuje również potencjalne konflikty interesów. Koncepcja systemu przedstawiona jest na rysunku 4.

Rysunek 4. Wstępna koncepcja systemu wspomagania wyboru recenzentów



Źródło: opracowanie własne autorów

2. Założenia i ograniczenia

System automatycznie gromadzi wiedzę o potencjalnych recenzentach. Problem wielojęzyczności dokonań naukowców (głównie publikacji) jest bardzo trudny do rozwiązania, ponieważ wymaga tłumaczenia słów kluczowych na wiele języków. Przyjęto zatem założenie, że w tej fazie projektu system będzie brał pod uwagę publikacje napisane tylko w języku angielskim i polskim. Dzięki udostępnieniu serwisu WWW dowolny użytkownik będzie miał dostęp do systemu z publicznej sieci, przez przeglądarkę internetową. Usługa sieciowa (*web service*) pozwoli innym systemom informatycznym zadawać pytania do bazy wiedzy o recenzentach. Interfejs użytkownika ma dwie wersje językowe: angielską i polską. Rdzeniem, na którym zbudowany został system, jest BWNP – istniejące profile naukowców dotyczą tylko osób znajdujących się w tej bazie.

System został wykonany i wdrożony z wykorzystaniem następujących technologii:
- język programowania Java i technologia J2EE;
- serwer aplikacji Boss;
- baza danych Oracle;
- system operacyjny Linux.

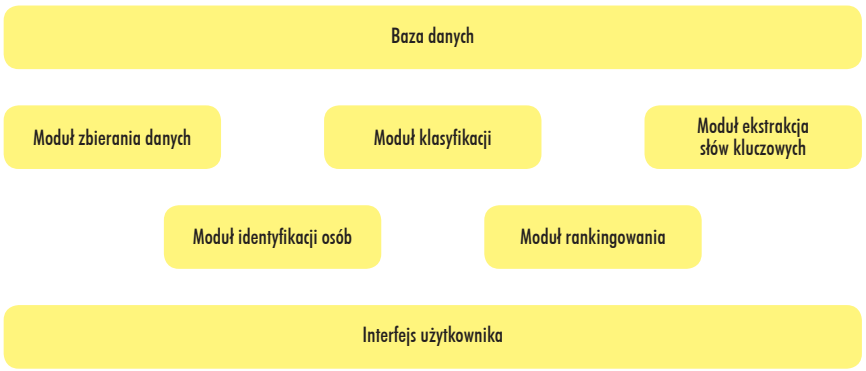
II. Architektura systemu

1. Warstwy i składowe systemu

W związku z przyjętymi ograniczeniami zaprojektowano system składający się z następujących elementów (rysunek 5):
1. baza danych;
2. moduły merytoryczne:
   - moduł zbierania danych;
   - moduł klasyfikacji;
   - moduł identyfikacji osób;
   - moduł ekstrakcji słów kluczowych;
   - moduł rankingowania;
3. interfejs użytkownika.

Pięć modułów merytorycznych odpowiada za przetwarzanie danych. Bezpośredni użytkownicy ich nie dostrzegają, poprzez interfejs użytkownika widoczne są jedynie efekty ich działań.

Rysunek 5. Elementy systemu wspomagania wyboru recenzentów



Źródło: opracowanie własne autorów

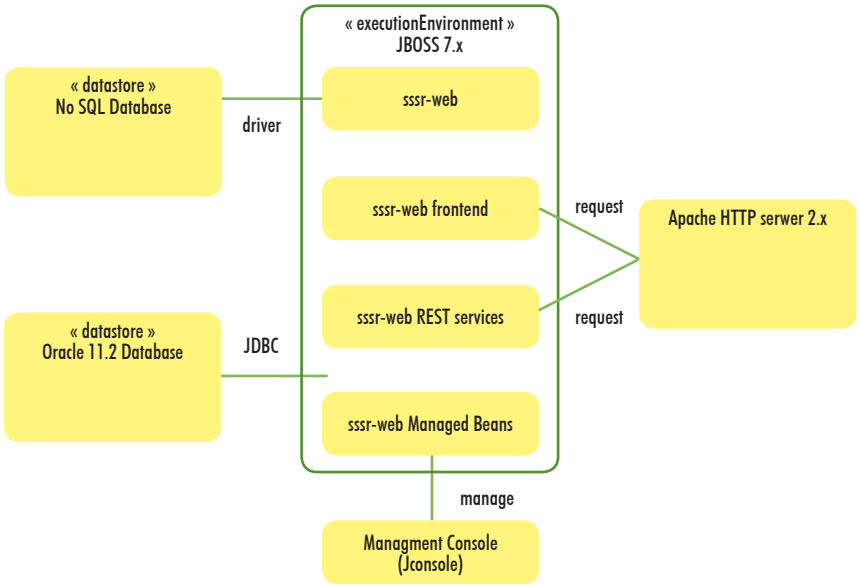
2. Architektura fizyczna

Projekt architektury systemu przedstawiono na rysunku 6. System pracuje pod kontrolą serwera aplikacyjnego JBoss[59] w wersji 7. Kod źródłowy zawierający implementację systemu zorganizowany jest w projekcie o nazwie *sssr-web*, który kompilowany jest jako archiwum Web Application Archive (WAR) i wdrażany na serwer aplikacyjny. *Sssr-web* składa się z trzech odrębnych interfejsów, umożliwiających dostęp do różnych funkcjonalności:

1. interfejs WWW, dostępny dla użytkowników końcowych przez przeglądarkę internetową;
2. serwisy Representational State Transfer (REST), pozwalające na integrację z innymi systemami informatycznymi;
3. obiekty Managed Beans, służące administratorom systemu do zarządzania procesami działającymi na serwerze aplikacyjnym.

Interfejsy WWW oraz REST są udostępnione publicznie poprzez Apache HTTP Server[60] w wersji 2.x. Zarządzanie procesami systemu poprzez obiekty Managed Beans jest możliwe tylko z sieci lokalnej z użyciem konsoli administracyjnej. System korzysta z dwóch źródeł danych: bazy relacyjnej Oracle w wersji 11g oraz bazy NoSQL[61] (MongoDB[62]).

Rysunek 6. Architektura fizyczna systemu wspomagania wyboru recenzentów



Źródło: opracowanie własne autorów

3. Architektura logiczna

System wspomagania wyboru recenzentów zbudowano na podstawie trzech składników: **warstwy bazy danych, warstwy biznesowej** oraz **interfejsu użytkownika**. Architektura logiczna systemu przedstawiona została na rysunku 7.

Model danych przechowywany jest w bazie danych Oracle oraz – częściowo – w bazie NoSQL. Baza relacyjna przechowuje wszystkie dane niezbędne do działania systemu, natomiast NoSQL wykorzystuje się w celu zwiększenia wydajności operacji wymagających szybkiego dostępu do konkretnych danych, gdy pobieranie

[59] JBoss, http://www.jboss.org, dostęp 08.08.2012.
[60] Apache, http://www.apache.org, dostęp 08.08.2012.
[61] NoSQL to nierelacyjne bazy danych nowej generacji.
[62] MongoDB, http://www.mongodb.org, dostęp 08.08.2012.

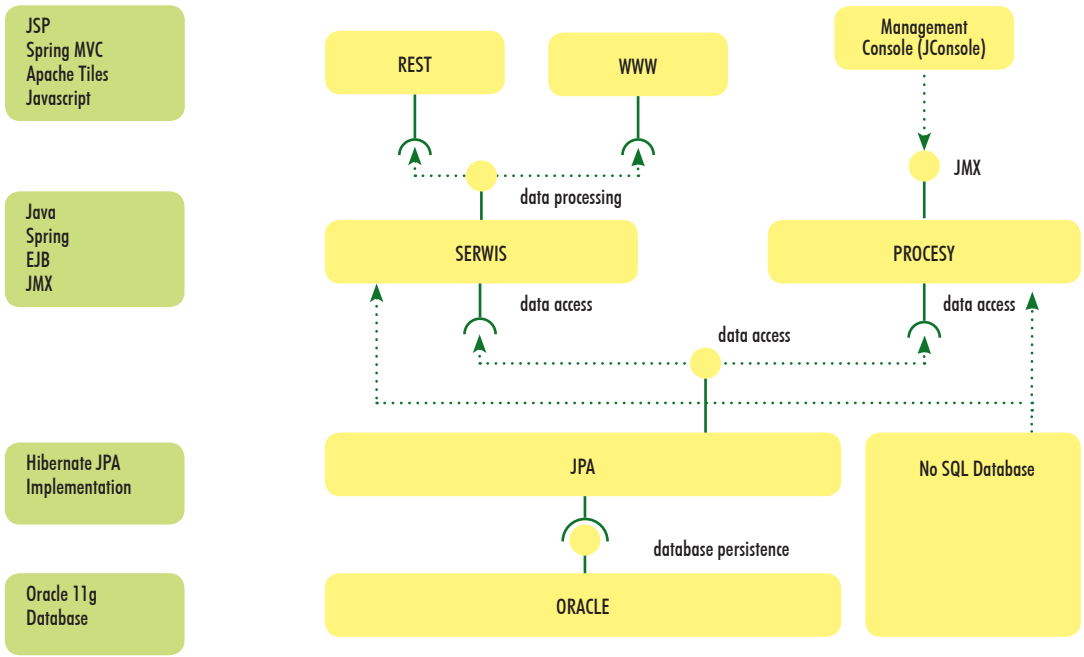
tych danych z Oracle trwa zbyt długo. Synchronizacja i ewentualne przetworzenie danych do oczekiwanego formatu wykonywane są na żądanie procesu, który korzysta z NoSQL. Ponieważ procesy synchronizują tylko dane, z których będą korzystać, operacja nie wymaga dużego nakładu obliczeniowego. Dostęp do danych z poziomu języka Java realizowany będzie przez Java Persistence API (JPA). Baza NoSQL umożliwia bezpośredni dostęp do danych i przechowywanie ich w postaci obiektów Java.

Warstwa biznesowa zrealizowana jest w języku Java, z wykorzystaniem technologii Java Management Extensions (JMX) oraz Spring Framework[63]. Część obliczeniowa, zajmująca się głównie przetwarzaniem danych, dzieli się na dwie jednostki logiczne:

- **serwisy** – ich zadanie to przetwarzanie danych do celów prezentacji na interfejsie użytkownika lub przekazania do interfejsu REST;
- **procesy** – ich celem jest wykonywanie długotrwałych obliczeń z wykorzystaniem danych. Procesy mogą być uruchamiane cyklicznie przez serwer aplikacyjny na podstawie wcześniej ustalonej konfiguracji lub wywoływane na żądanie administratora systemu. Większość procesów realizować będzie algorytmy niezbędne do prawidłowego działania systemu, takie jak: klasyfikacja publikacji, identyfikacja autorów, zbieranie danych o publikacjach w internecie, ekstrakcja słów kluczowych.

Dane zwracane przez serwisy trafiają do interfejsu użytkownika. Mogą być prezentowane na stronach WWW lub przekazywane do interfejsu REST. Warstwa prezentacji zaimplementowana została z wykorzystaniem technologii Java Server Pages (JSP), Spring MVC, Apache Tiles, AJAX oraz JavaScript. REST wykorzystuje Spring MVC. Administracja procesami nie jest elementem interfejsu użytkownika. Odbywa się ona przez obiekty udostępnione przez rozszerzenia JMX (Managed Bean). Do zarządzania procesami można wykorzystać dowolną aplikację umożliwiającą podłączenie się do serwera JMX maszyny wirtualnej. Jedną z takich aplikacji jest JConsole, domyślnie udostępniana wraz z dystrybucją JDK Java.

Rysunek 7. Architektura logiczna systemu wspomagania wyboru recenzentów



Źródło: opracowanie własne autorów

[63] Spring, http://www.springsource.org, dostęp 08.08.2012.

III. Funkcjonalności interfejsu użytkownika

Interfejs użytkownika jest dostępny przez przeglądarkę WWW. Do wybranych funkcjonalności będzie można dotrzeć w ramach usługi sieciowej (WebService); skorzystają z niej także inne systemy informatyczne. Interfejs posiada angielską i polską wersję językową.

Menu użytkownika zawiera następujące opcje:
1. *ranking of reviewers* (ranking recenzentów):
   - *for document* (dla dokumentu);
   - *for specified criteria* (dla zadanych kryteriów);
2. *analyze document* (analiza dokumentu);
3. *databases* (bazy danych):
   - *scientific fields* (dziedziny naukowe);
   - *people* (ludzie);
   - *keywords* (słowa kluczowe);
   - *publications* (publikacje);
   - *sources* (źródła);
4. *administration* (administracja);
5. *about* (o systemie).

Narzędzia administracyjne zawierają operacje takie, jak: logowanie do systemu, zmiana hasła, zmiana danych osobowych użytkownika. System znajduje się w internecie, ale dostęp do wrażliwych danych mają wyłącznie autoryzowani odbiorcy. Ograniczonym dostępem zostały również objęte narzędzia administracyjne oraz funkcja generowania rankingu recenzentów.

1. Ranking recenzentów dla dokumentu

Wygenerowanie rankingu recenzentów możliwe jest na dwa sposoby. Pierwszy z nich tworzy ranking na podstawie dokumentu, który można dodać z poziomu interfejsu (rysunek 8). Dokument do analizy może być wklejony lub wprowadzony z klawiatury do odpowiedniego pola tekstowego (opcja „wprowadź tekst"). Długość uzupełnianego tekstu nie może przekroczyć stu tysięcy znaków. Istnieje także możliwość wskazania zewnętrznego pliku zawierającego dokument do analizy (opcja „dodaj plik"). System wspiera pliki z rozszerzeniami DOC, DOCX, ODT, PDF oraz TXT, a dopuszczalny rozmiar pliku wynosi 10 megabajtów.

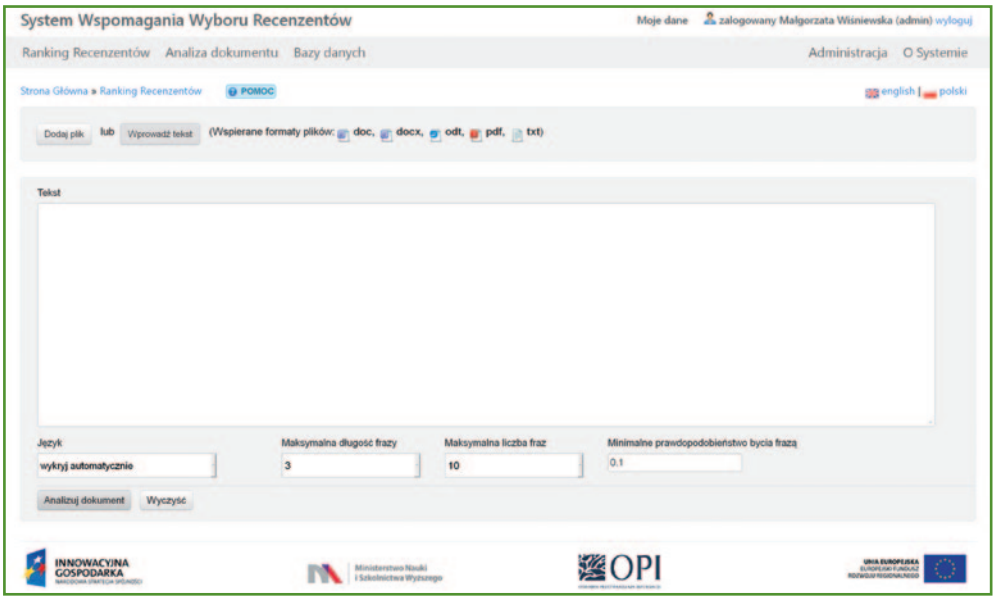
Na podstawie wpisanego tekstu system wyodrębnia słowa kluczowe, a następie bierze je pod uwagę przy tworzeniu rankingu recenzentów. Dodatkowe opcje związane z ekstrakcją słów kluczowych pozwalają określić maksymalną długość frazy, maksymalną liczbę fraz oraz minimalne prawdopodobieństwo bycia słowem kluczowym. Język dokumentu może być określony manualnie (polski, angielski) lub automatycznie przez system. Wciśnięcie przycisku „analizuj dokument" powoduje przesłanie tekstu do systemu oraz przejście do następnego kroku, w którym wydzielone słowa kluczowe zostają automatycznie dodane jako jedno z kryteriów wyszukiwania do wygenerowania rankingu. Dodatkowe kryteria, które po analizie dokumentu można uzupełnić, zostaną dokładniej opisane w dalszej części rozdziału.

2. Ranking recenzentów dla zadanych kryteriów

Drugi sposób generowania rankingu recenzentów pozwala na przejście do kryteriów wyszukiwania z pominięciem możliwości dołączenia dokumentu (rysunek 9). Interfejs umożliwia wprowadzenie takich informacji, jak: słowa kluczowe w języku polskim i angielskim, dziedziny naukowe, wnioskodawcy, wykonawcy. Wymaga się wpisania co najmniej jednego słowa kluczowego. Kolejność słów można zmieniać, przeciągając je lewym przyciskiem myszki. W wyborze dziedziny pomaga dodatkowe okno dialogowe, w którym można wybrać określoną dziedzinę, poruszając się w hierarchii klasyfikacji dziedzin naukowych. Wprowadzenie kilku pierw-

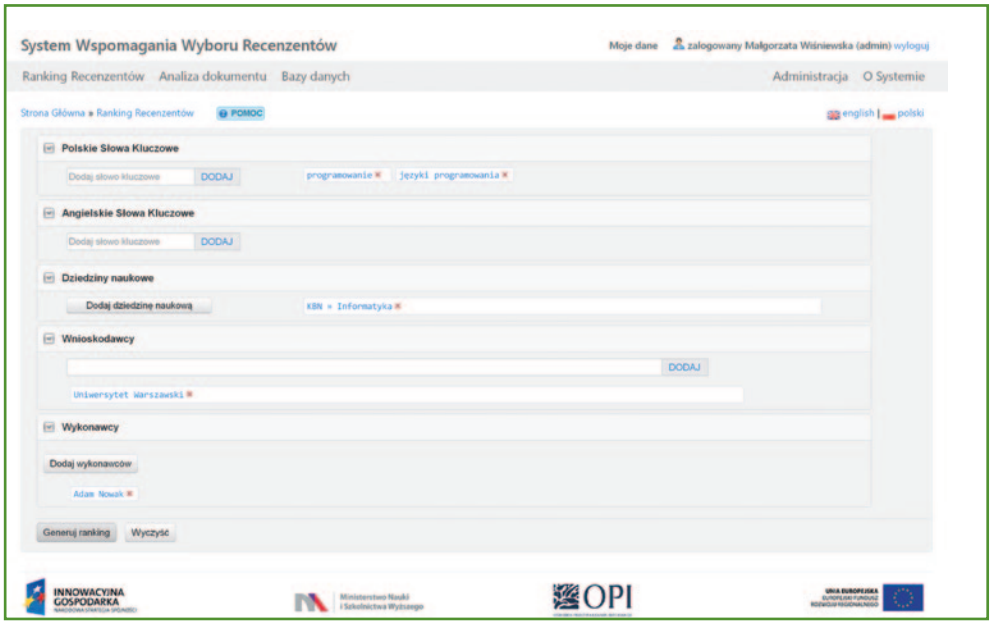
szych znaków w polu wnioskodawcy spowoduje wyświetlenie podpowiedzi z nazwami instytucji. Możliwe jest też uszczegółowienie, poprzez dodanie nazwy wydziału bądź instytutu. Wykonawców można wybrać z okna, które daje szansę wyszukiwania ludzi nauki przez podanie imienia lub nazwiska osoby. Przycisk „generuj ranking" służy do tworzenia listy recenzentów, posortowanej według zgodności z zadanymi kryteriami.

Rysunek 8. Przykład interfejsu prezentującego analizę dokumentu



Źródło: System Wspomagania Wyboru Recenzentów, OPI

Rysunek 9. Przykład interfejsu definicji rankingu



Źródło: System Wspomagania Wyboru Recenzentów, OPI

Kliknięcie w konkretną osobę spowoduje wyświetlenie szczegółowych informacji o niej. Ewentualne konflikty interesów wynikające z afiliacji zaznaczone są kolorem czerwonym. Ponadto, w szczegółach dotyczących naukowca znajduje się zestawienie potencjalnych konfliktów z instytucją wnioskującą o grant (ze względu na zatrudnienie, pełnione funkcje, członkostwo w organizacjach) oraz z poszczególnymi wykonawcami projektu (ze względu na wspólną pracę, członkostwo w tych samych organizacjach, promowanie prac doktorskich, recenzowanie prac doktorskich i habilitacyjnych, współautorstwo publikacji). W każdej z tych kategorii podawana jest pełna lista instytucji, organizacji, prac doktorskich i habilitacyjnych oraz publikacji, które mogą powodować konflikt interesów.

3. Analiza dokumentu

Analiza dokumentu umożliwia wyodrębnienie słów kluczowych z wprowadzonego tekstu lub z dołączonego pliku (rysunek 10). Wymagania odnośnie do dopuszczalnych formatów pliku oraz długości tekstu są takie same jak w przypadku funkcji generowania rankingu recenzentów. Dodatkowe opcje odpowiadają za określenie języka (automatyczne wykrywanie: polski, angielski), maksymalnej długości frazy (domyślnie ustawiona wartość 3), maksymalnej liczby fraz (domyślnie 10), minimalnego prawdopodobieństwa bycia frazą kluczową (domyślnie 0,1).

Wybór opcji „analizuj dokument" powoduje wyświetlenie trzech zakładek przedstawiających wyniki analizy. Zakładka „tekst" zawiera podstawowe statystyki tekstu (język tekstu, język modelu ekstrakcji słów kluczowych, liczba słów, liczba znaków *etc.*), a dodatkowo wyróżnia kolorem granatowym słowa kluczowe. W zakładce „ranking fraz kluczowych" znajduje się lista wyodrębnionych słów wraz z ich formą oryginalną, formą zlematyzowaną oraz prawdopodobieństwem bycia słowem kluczowym. Ostatnia zakładka – „chmura fraz kluczowych" jest graficzną reprezentacją listy słów kluczowych, gdzie rozmiar czcionki zależy od istotności tego słowa.

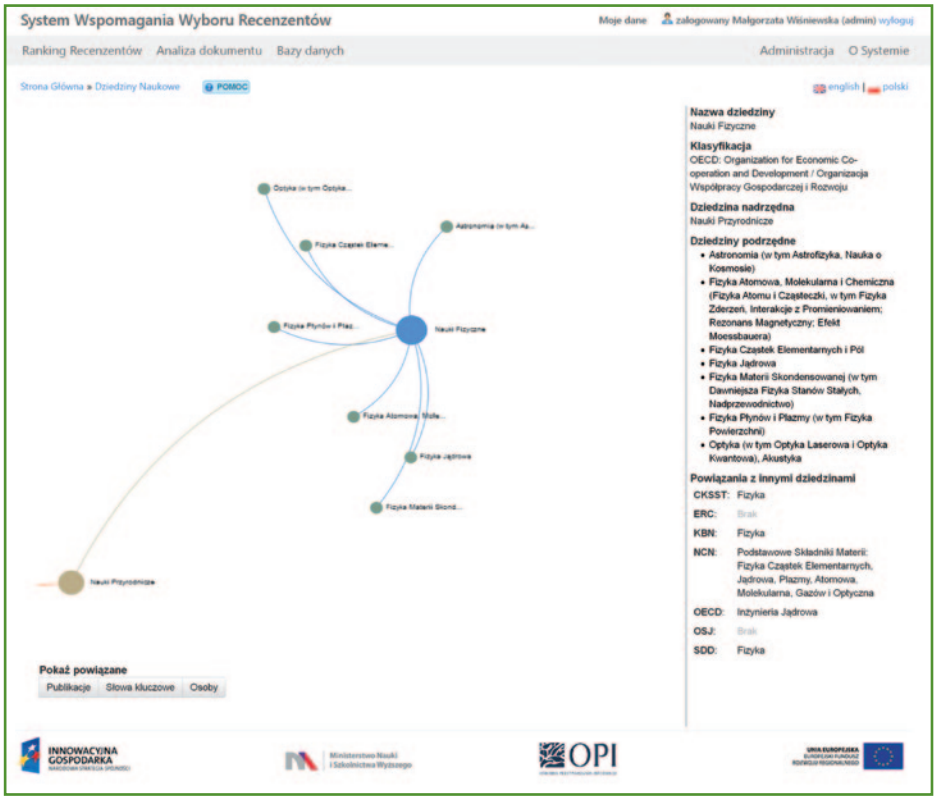
Rysunek 10. Przykład interfejsu analizy dokumentu

Źródło: System Wspomagania Wyboru Recenzentów, OPI

4. Dziedziny naukowe

Widok dla dziedzin naukowych podzielony jest na dwie sekcje odseparowane od siebie linią (rysunek 11). Lewa strona widoku przedstawia reprezentację siedmiu klasyfikacji nauki w postaci grafu lub tabeli (dla starszych przeglądarek internetowych). Sekcja ta umożliwia wybór dziedziny naukowej należącej do dowolnej z tych klasyfikacji. Po kliknięciu na któryś z węzłów grafu lub wybraniu opcji „pokaż szczegóły w przypadku tabeli", po prawej stronie ekranu pojawiają się szczegóły dotyczące dziedziny. Kliknięcie w węzeł powoduje jego wyśrodkowanie, przybliżenie i pokazanie dziedzin podrzędnych. Wszystkie połączenia hierarchiczne można wyświetlić jednocześnie poprzez zaznaczenie checkboksa „pokaż wszystkie połączenia hierarchiczne". Po wybraniu tej opcji, na grafie automatycznie pojawią się relacje dla wybranej dziedziny, które można wyłączyć odznaczając *checkbox* „pokaż relacje". Widok przedstawiający szczegóły posiada dodatkowe opcje, za pomocą których użytkownik zobaczy powiązane publikacje, słowa kluczowe oraz osoby związane z wybraną dziedziną. Dane dotyczące relacji pojawiają się pod sekcją zawierającą reprezentację siedmiu klasyfikacji.

Rysunek 11. Przykład interfejsu prezentującego klasyfikacje nauki



Źródło: System Wspomagania Wyboru Recenzentów, OPI

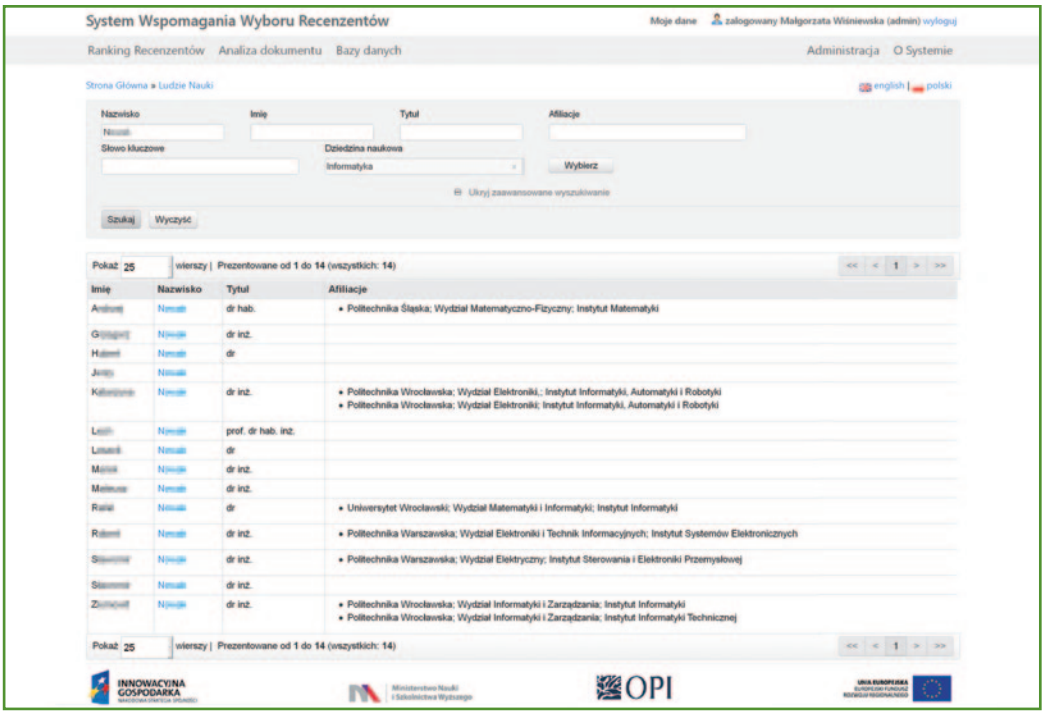
5. Ludzie

System umożliwia odnajdywanie ludzi nauki poprzez określenie ich imienia, nazwiska, stopnia lub tytułu naukowego oraz afiliacji. Dodatkowo w wyszukiwaniu zaawansowanym można ustalić słowo kluczowe opisujące osobę oraz przypisaną do niej dziedzinę (rysunek 12). Naciśnięcie przycisku „szukaj" powoduje wyświetlanie listy z wynikami wyszukiwania, zawierającej podstawowe dane. Po kliknięciu w nazwisko pojawią się dane kontaktowe oraz klasyfikacje, do których dana osoba jest przypisana, a także informacje szczegółowe:

- afiliacje – miejsca pracy, pełnione funkcje, członkostwa, powiązania z innymi instytucjami poprzez publikacje;

- promowanie i recenzje – lista promowanych prac doktorskich, recenzje prac doktorskich i habilitacyjnych;
- recenzje grantowe – statystyki dotyczące recenzji naukowca na potrzeby programów grantowych;
- słowa kluczowe – lista fraz kluczowych przypisanych do osoby wraz z ich wagami;
- publikacje – lista prac naukowych.

Rysunek 12. Przykład interfejsu prezentującego wyszukiwanie ludzi nauki



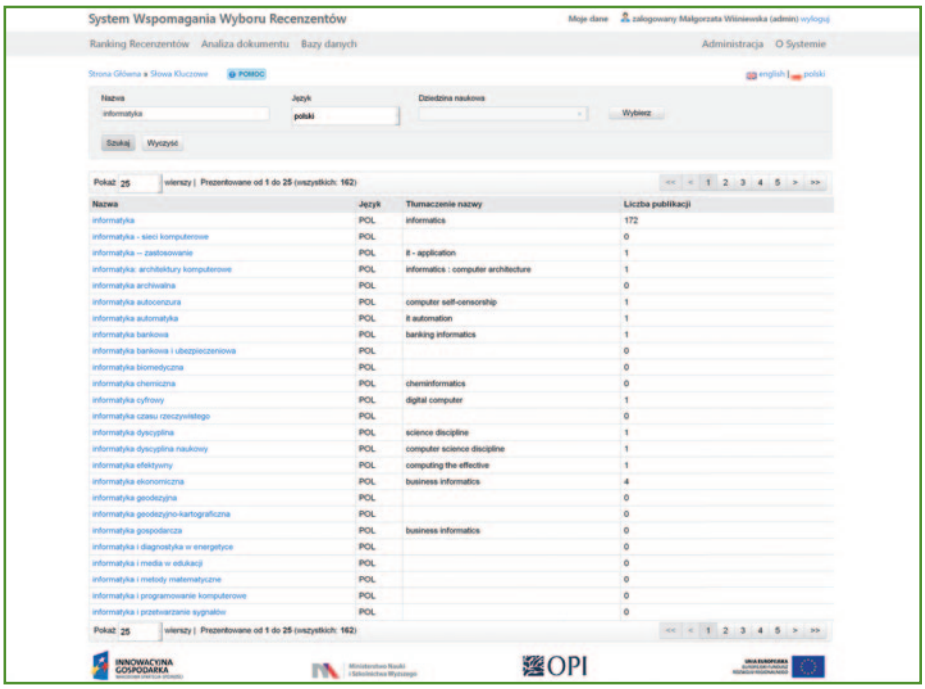
Źródło: System Wspomagania Wyboru Recenzentów, OPI

6. Słowa kluczowe

Formularz umożliwia wyszukiwanie słów kluczowych poprzez wprowadzenie nazwy, określenie języka oraz dziedziny, do której dana fraza została przypisana (rysunek 13). Wyniki prezentowane są w postaci tabelki; poza znalezionym słowem znajduje się w niej język, tłumaczenie oraz liczba publikacji i osób, do których słowo przynależy. Opcja wyszukiwania uwzględnia frazy kluczowe przypisane do osób, automatycznie wyekstraktowane z abstraktów oraz manualnie przyporządkowane do publikacji. Kliknięcie w dane słowo wyświetla ekran prezentujący szczegółowe informacje (rysunek 14):

1. wiadomości ogólne: język, typ, opis znaczenia słowa pochodzący z Wikipedii;
2. klasyfikacje słowa: w jakich dziedzinach i dyscyplinach nauki występuje dane słowo; przyporządkowanie to wynika z tego, że każde ze słów związane jest z publikacją lub osobą, a te obiekty są już sklasyfikowane według siedmiu modeli klasyfikacji nauki;
3. treści powiązane ze słowem, będące częściowo namiastką ontologii:
   - słowa kluczowe związane z danym słowem na podstawie Wikipedii – prezentowany jest prosty graf związków, dodatkowo każdy z węzłów grafu dostarcza objaśnienia słowa powiązanego;
   - słowa kluczowe związane z danym słowem na podstawie współwystąpień w publikacjach – prezentowane jest tabelaryczne oraz graficzne zestawienie częstości współwystąpień;
   - powiązane osoby, czyli osoby, które użyły danego słowa do opisu siebie lub użyły go w streszczeniach swoich publikacji;
   - powiązane publikacje, czyli lista publikacji, w których wystąpiło dane słowo.

Rysunek 13. Przykład interfejsu prezentującego listę słów kluczowych



Źródło: System Wspomagania Wyboru Recenzentów, OPI

Rysunek 14. Przykład interfejsu prezentującego szczegóły słowa

Źródło: System Wspomagania Wyboru Recenzentów, OPI

7. Publikacje

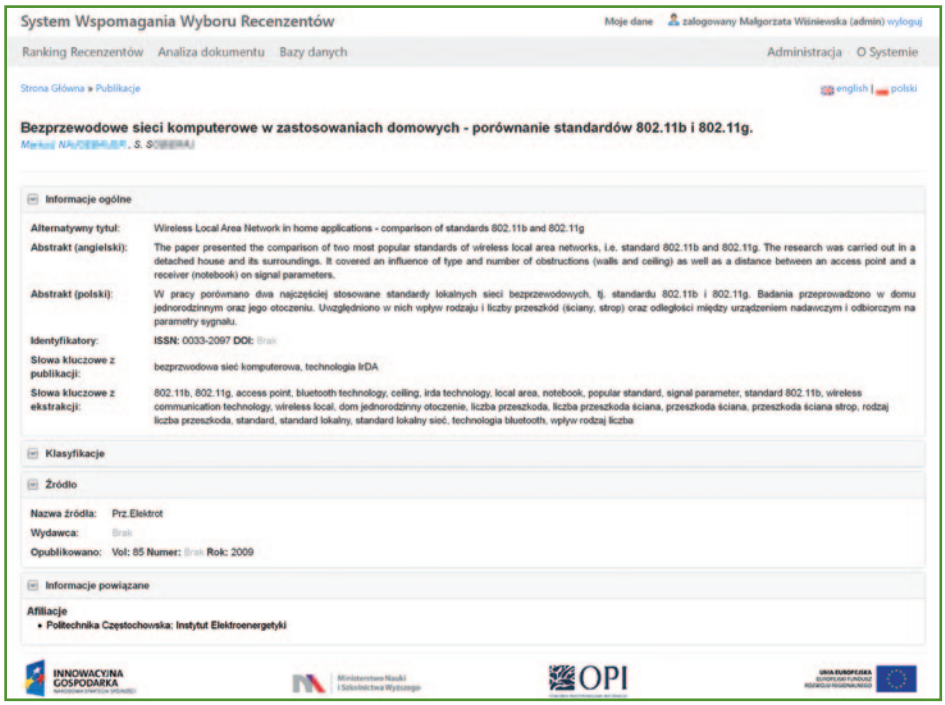
Proste wyszukiwanie publikacji możliwe jest poprzez podanie dowolnej frazy z tytułu, nazwiska autora oraz roku (lub przedziału lat) wydania. Dodatkowo, zaawansowana opcja rozszerza zakres kryteriów

wyszukiwania poprzez podanie źródła, typu publikacji oraz dziedziny, do której należy (rysunek 15). Po wprowadzeniu pierwszych trzech liter w polu ze źródłem, generowana jest lista możliwych pozycji do wyboru, każdy kolejny znak uszczegóławia zapytanie. Wyniki wyszukiwania przedstawione są w postaci listy zawierającej tytuł publikacji, autorów, rok wydania i źródło. Kliknięcie w tytuł powoduje przejście do szczegółowego widoku:

- lista autorów;
- informacje ogólne – alternatywny tytuł, abstrakty, identyfikatory ISSN i DOI, słowa kluczowe;
- klasyfikacje – przypisane dziedziny naukowe;
- źródło – nazwa źródła, wydawca, numer wydania i rok;
- informacje powiązane – afiliacje.

Kliknięcie w nazwisko autora przekierowuje na stronę opisującą daną osobę, natomiast w źródło – na stronę szczegółową źródła.

Rysunek 15. Przykład interfejsu prezentującego szczegóły publikacji



Źródło: System Wspomagania Wyboru Recenzentów, OPI

8. Źródła

W systemie rozróżniamy trzy typy źródeł: książki, czasopisma i konferencje. Ich wyszukiwanie możliwe jest poprzez wybranie jednego z dostępnych typów, określenie dziedziny, do której należy oraz wprowadzenie dowolnej liczby znaków z tytułu źródła. Lista wyników przedstawia pełny tytuł, typ oraz wydawcę (rysunek 16). Kliknięcie w tytuł przekierowuje do widoku ze szczegółami:

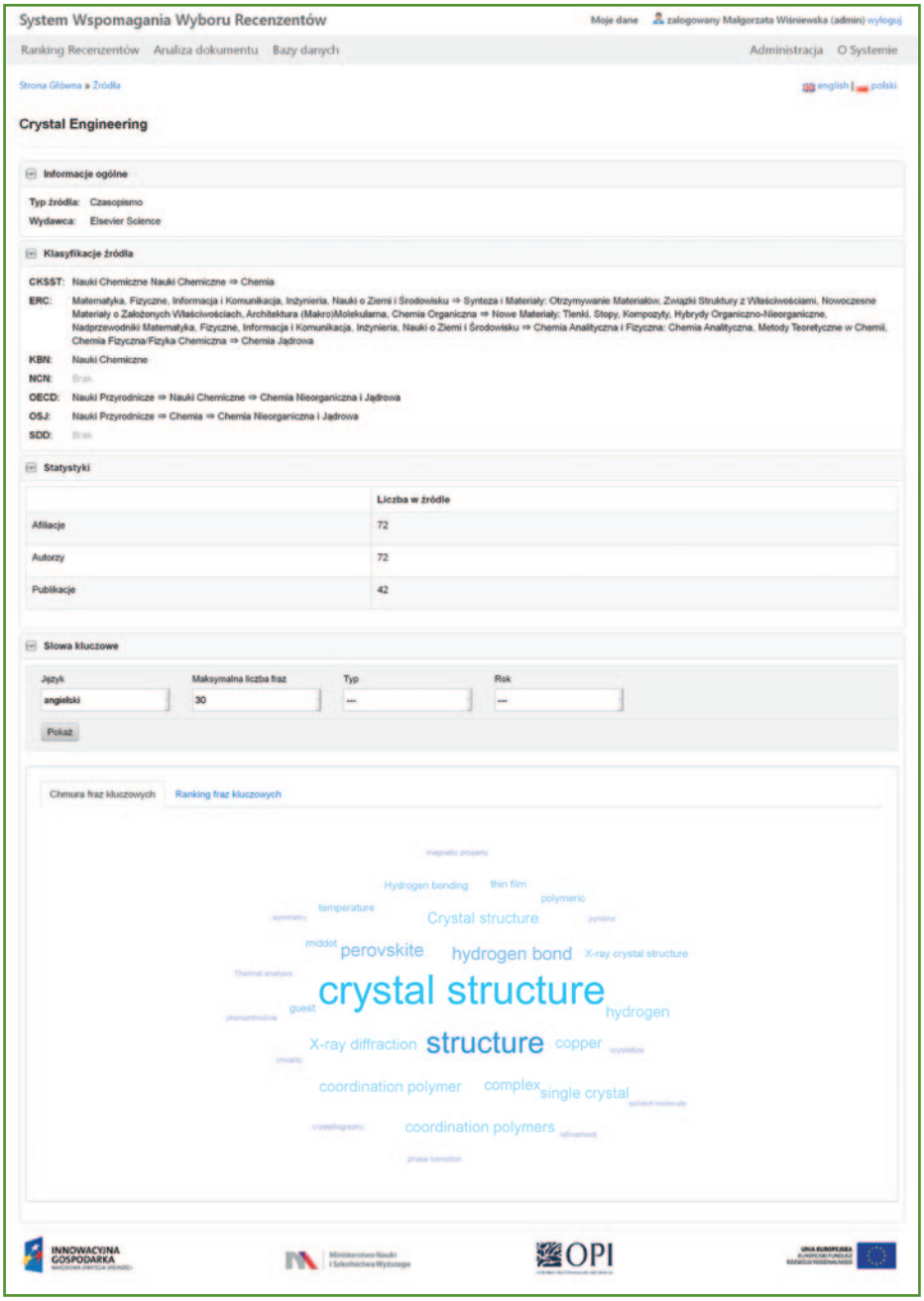
- informacje ogólne – typ źródła, wydawca;
- klasyfikacje źródła – dziedziny, do których dane źródło jest przypisane;
- statystyki – informacje o liczbie autorów, afiliacji oraz publikacjach powiązanych;
- słowa kluczowe – lista słów kluczowych związanych z danym źródłem.

Prezentacja wyników dostępna jest w wersji graficznej oraz tabelarycznej.

9. Administracja

Narzędzia systemowe dostępne są dla użytkowników z uprawnieniami administratora, po zalogowaniu do systemu. Obejmują zagadnienia związane z parametrami systemowymi, słownikami i zarządzaniem użytkownikami. Wszystkie parametry systemowe opisane są przez nazwę parametru oraz wartość. Każdy parametr systemowy może być oznaczony jako edytowalny lub nieedytowalny. System umożliwia dodawanie nowych i usuwanie już istniejących parametrów. Słowniki systemowe przedstawiają zbiór identyfikatorów używanych w systemie wraz z ich opisem (na przykład słownik typów publikacji, który identyfikuje publikację jako artykuł, książkę czy materiał konferencyjny). Narzędzia systemowe umożliwiają również dodawanie nowych użytkow-

Rysunek 16. Przykład interfejsu prezentującego szczegóły źródła



Źródło: System Wspomagania Wyboru Recenzentów, OPI

ników (rysunek 17). Takie działanie poprzedza proces weryfikacji, w którym sprawdza się, czy podane dane są odpowiedniego formatu i czy wszystkie wymagane pola są wypełnione. Każdy użytkownik musi mieć przypisaną przynajmniej jedną rolę. Proces dodania nowego klienta kończy się wysłaniem e-maila z informacją o haśle i loginie. System umożliwia edycję oraz usuwanie istniejących już użytkowników oraz resetowanie hasła.

Rysunek 17. Przykład interfejsu prezentującego listę użytkowników systemu

Źródło: System Wspomagania Wyboru Recenzentów, OPI

IV. Podsumowanie

Badania wybranych systemów wspomagających proces recenzowania wykazały, że nie ma możliwości zaadaptowania istniejących implementacji do potrzeb polskiego środowiska naukowego. W tej sytuacji rozsądnym krokiem wydaje się opracowanie i wdrożenie własnych rozwiązań.

Ośrodek Przetwarzania Informacji – Instytut Badawczy posiada pewne doświadczenia w obsłudze procesu recenzowania. Skuteczność dotychczasowych narzędzi została zweryfikowana poprzez ankietę rozesłaną do wnioskodawców i recenzentów w systemach informatycznych obsługiwanych przez OPI. Najważniejsza konkluzja brzmi: niezbędna jest praca nad systemem wspomagania wyboru recenzentów oraz polepszeniem organizacji całego procesu. Przedstawiony projekt nowego systemu zakłada, że ranking recenzentów będzie budowany na podstawie aktualnego dorobku naukowego poszczególnych osób. Zbadana zostanie zgodność słów kluczowych wydobytych z wniosku oraz słów kluczowych związanych z potencjalnym oceniającym, przy czym słowa kluczowe naukowca będą wydobyte automatycznie z jego dorobku.

Projekt systemu był implementowany przez zespół laboratorium inteligentnych systemów informatycznych OPI. Po zakończeniu prac programistycznych, przyjęte założenia poddano weryfikacji na drodze doświadczeń. Szczegółowy opis algorytmów oraz niektórych testów znajduje się w następnym rozdziale.

V. Bibliografia

Zespół laboratorium inteligentnych systemów informatycznych Ośrodka Przetwarzania Informacji – Instytutu Badawczego, Projekt systemu, niepublikowany, OPI, Warszawa 2011.
Zespół laboratorium inteligentnych systemów informatycznych Ośrodka Przetwarzania Informacji – Instytutu Badawczego, System wspomagania wyboru recenzentów (prototyp), niepublikowany, OPI, Warszawa 2011.

Źródła internetowe:

Apache, http://www.apache.org, dostęp 08.08.2012.
JBoss, http://www.jboss.org, dostęp 08.08.2012.
MongoDB, http://www.mongodb.org, dostęp 08.08.2012.
Spring, http://www.springsource.org, dostęp 08.08.2012.

Rozdział czwarty

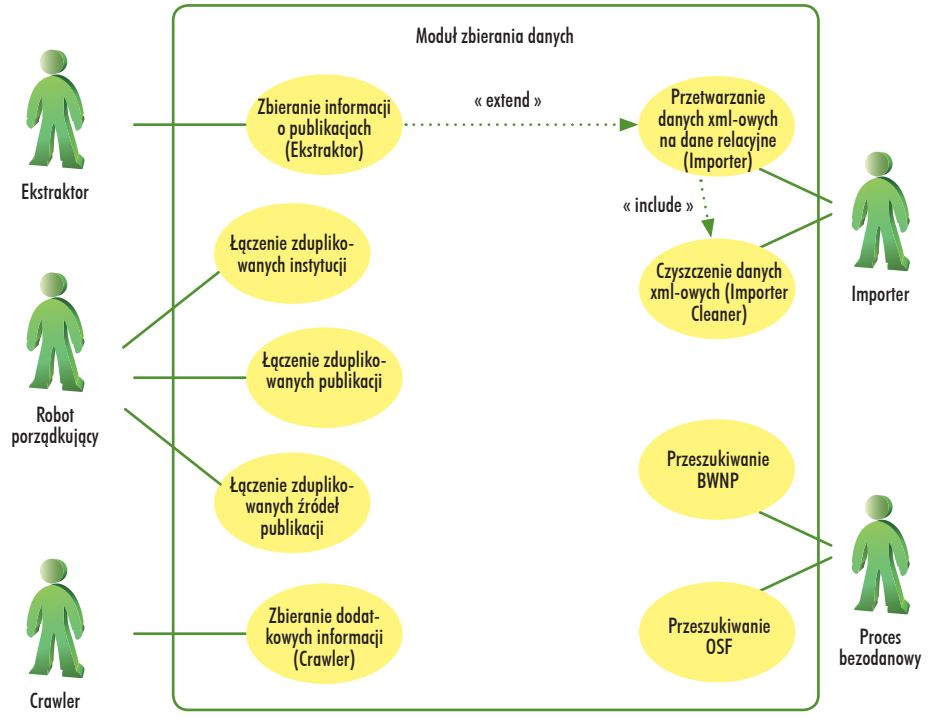
MODUŁY MERYTORYCZNE

(Jarosław Protasiewicz, Sławomir Dadas, Małgorzata Gałężewska, Tomasz Stanisławek, Marek Kozłowski, Jan Artysiewicz)

I. Moduł zbierania danych

Celem modułu jest utworzenie możliwie kompletnej bazy dorobku naukowego polskich uczonych, co pozwoli przygotować ich profile oraz ranking potencjalnych recenzentów. Moduł wyszukuje, pobiera i składuje informacje o ewentualnych ekspertach w jednorodnej formie, w bazie danych systemu wspomagania wyboru recenzentów. Ze względu na kategorie tych informacji wyróżniono pięć podstawowych funkcjonalności modułu: proces bazodanowy, ekstraktor, importer, crawler (robot internetowy analizujący strony naukowców) oraz procesy porządkujące dane, czyli instytucje, źródła i publikacje (rysunek 18).

Rysunek 18. Role i czynności użytkowników w module zbierania danych



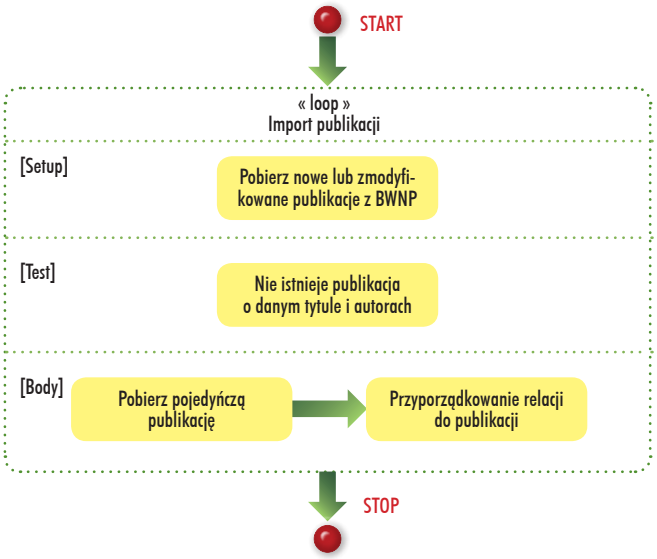
Źródło: opracowanie własne autorów

1. Procesy bazodanowe

Procesy bazodanowe mają charakter cykliczny i przyrostowy, co oznacza, że w określonych odstępach czasu aktualizują przyrostowo bazę systemu wspomagania wyboru recenzentów, pobierając dane z BWNP i bazy OSF. Wszystkie osoby z BWNP zostały pobrane do systemu wspomagania wyboru recenzentów; dane o nich są uaktualniane na bieżąco. Podobnie dzieje się z publikacjami, ich autorami i afiliacjami. Klasyfikacje osób z bazy OSF pobierane są cyklicznie lub jednorazowo, w zależności od modelu klasyfikacji (na przykład klasyfikacje SDD[64] nie są już rozwijane w OSF, więc zostały zaimportowane jednorazowo).

Procesy bazodanowe zrealizowane zostały w języku PL/SQL i są uruchamiane jako samodzielne procesy w bazie danych. Procesy są skomplikowane i nie wnoszą wartości naukowej, dlatego przedstawiono tylko wybrany diagram prezentujący algorytm pobierania publikacji (rysunek 19).

Rysunek 19. Wybrany algorytm pobierania publikacji z Bazy Wiedzy o Nauce Polskiej



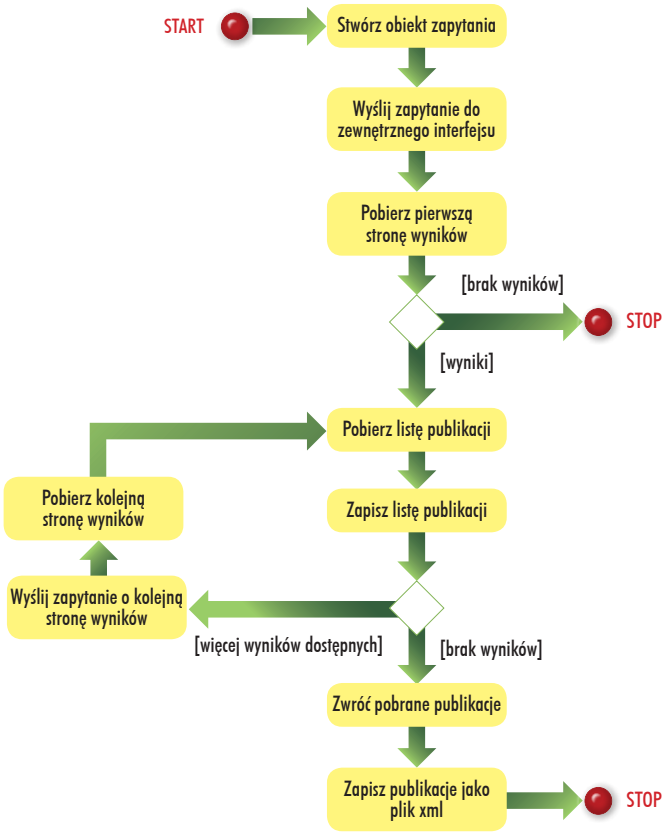
Źródło: opracowanie własne autorów

2. Ekstraktor i importer

Ekstraktor jest odpowiedzialny za zbieranie informacji na temat publikacji naukowych z zewnętrznych źródeł danych. Jego podstawową funkcjonalność wypełnia robot internetowy[65], działający jako proces na serwerze aplikacyjnym JBoss; ma za zadanie wysyłanie zapytań o publikacje do zdefiniowanych źródeł danych. Zapytania generowane przez robota zawierają nazwiska polskich naukowców pochodzące z BWNP. Źródłami są internetowe bazy publikacji, serwisy udostępniające API lub publicznie dostępne pliki będące obrazami baz danych. Robot pozyskuje ze źródeł zbiory publikacji autorów o nazwiskach wyspecyfikowanych w zapytaniu. Następnie **importer** przetwarza te dane, sprowadzając je do jednolitego formatu. Pobierane są abstrakty publikacji, informacje o źródle (np. „czasopismo"), cytowaniach, autorach, słowach kluczowych oraz afiliacjach (instytucjach powiązanych z publikacją). Przetworzone dane zapisywane są w bazie danych systemu. Algorytmy działania ekstraktora i importera obrazują rysunki 20 i 21.

[64] Słownik Dziedzin i Dyscyplin stosowany przez Ministerstwo Nauki i Szkolnictwa Wyższego.
[65] Zagadnienia teoretyczne wyjaśniono w dodatku.

Rysunek 20. Diagram aktywności ekstraktora



Źródło: opracowanie własne autorów

3. Crawler (robot analizujący strony internetowe)

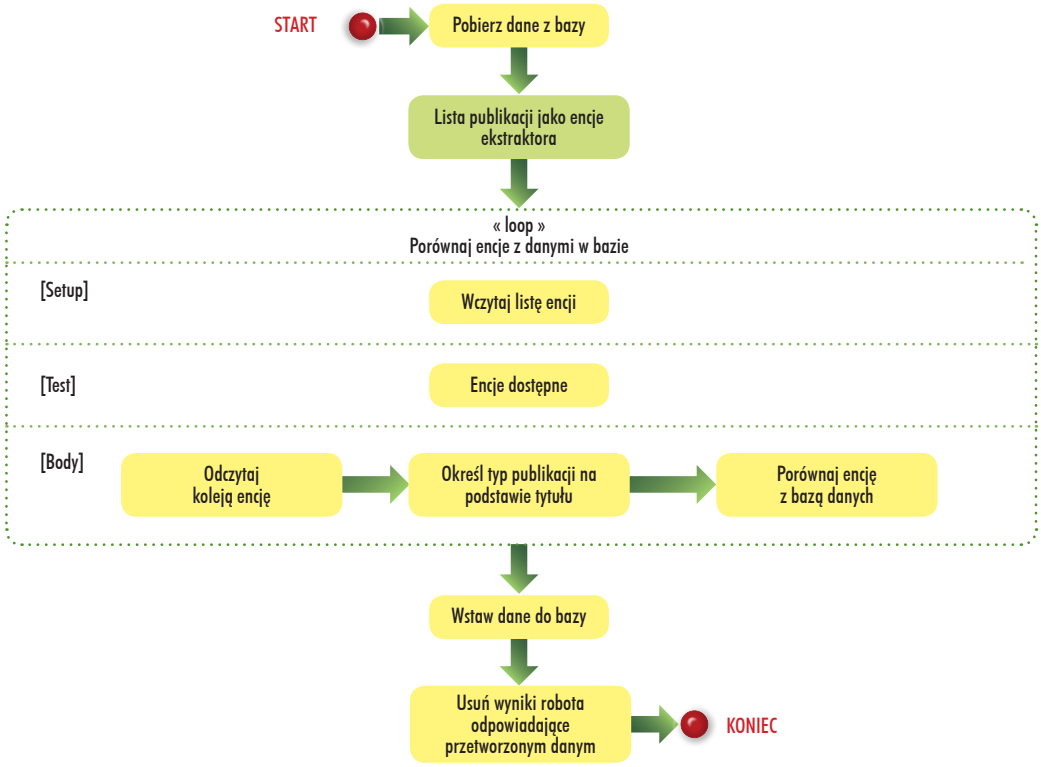
Zadaniem crawlera[66] jest uzupełnianie bazy systemu wspomagania wyboru recenzentów o dodatkowe informacje pochodzące ze stron internetowych. Podstawowa funkcjonalność polega na pobieraniu danych o publikacjach ze stron domowych naukowców. Poza tym do jego zadań należeć będzie zbieranie uzupełniających informacji o potencjalnych recenzentach: teksty publikowane w internecie, abstrakty, fragmenty publikacji *etc.*

W przeciwieństwie do ekstraktora, który pobiera dane ze zdefiniowanych baz na podstawie reguł utworzonych przez programistów, crawler jest w stanie automatycznie wyodrębniać publikacje ze stron o nieznanej strukturze. Jest to zadanie nietrywialne ze względu na różnorodność dokumentów, które można spotkać w internecie. Różnice występują zarówno na poziomie struktury strony (podział na podstrony, ilość informacji umieszczanych w poszczególnych sekcjach), jak i na poziomie struktury pojedynczych dokumentów (lokalizacja stałych elementów: nagłówka, menu, stopki, głównej treści *etc.*). Ponieważ niemożliwa jest bezpośrednia identyfikacja publikacji w tak zróżnicowanych dokumentach, należy je uprzednio przetworzyć w postać łatwiejszą do analizy. Moduł odpowiedzialny za crawlowanie podzielono na trzy odrębne procesy:

- przeszukiwanie strony naukowca wraz z jej podstronami w poszukiwaniu dokumentów zawierających publikacje;
- przetwarzanie pobranych dokumentów do postaci tekstowej;
- wykorzystanie metody CRF (*Conditional Random Fields*[67]) do sekwencyjnego etykietowania fragmentów dokumentu i wykrywania w nich publikacji.

[66] Zagadnienia teoretyczne wyjaśniono w dodatku.
[67] Bishop C.M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, New York 2006.

Rysunek 21. Diagram aktywności importera



Źródło: opracowanie własne autorów

Pierwszy z procesów dla zadanego adresu źródłowego pobiera dokument znajdujący się pod tym adresem, a następnie wyodrębnia z niego wszystkie odnośniki do innych dokumentów, niekoniecznie znajdujących się w tej samej domenie. Na podstawie reguł heurystycznych, dla każdego odnośnika dokonywana jest ocena, czy może on potencjalnie zawierać informacje o publikacjach naukowca. Reguły te opierają się na opisie linka oraz na samym adresie, poszukując w nich słów kluczowych lub fragmentów słów („publikacja", „artykuł", „badania", „science", „research", „naukowe" *etc.*). Po wybraniu zgodnych z regułami linków, są one pobierane i przetwarzane w taki sam sposób jak strona źródłowa. Crawler przeszukuje rekurencyjnie powiązane strony i – o ile są zgodne z regułami – zapisuje je. Przeszukiwanie jest przerywane dla dokumentów, w których nie znaleziono nazwiska naukowca lub gdy osiągnięta zostanie zdefiniowana głębokość przeszukiwania.

Drugi krok w procedurze wykrywania publikacji polega na wyodrębnieniu z pobranych dokumentów HTML tekstu przy zachowaniu elementów struktury oryginalnego dokumentu. Zachowanie struktury polega na wydzieleniu w tekście fragmentów, które w oryginale reprezentowały oddzielne sekcje, akapity, elementy listy lub wiersze tabeli. W pliku wynikowym takie wydzielone fragmenty zapisywane są w kolejnych liniach. Tekst jest zapisywany w takiej samej kolejności, w jakiej pojawiał się na stronie. Podział na linie dokonywany jest za pomocą prostych reguł. Na przykład każdy wiersz w tabeli (znacznik <tr> wewnątrz znacznika <table>) w pliku wynikowym zapisany zostanie w oddzielnej linii. Podobnie jest z kolejnymi elementami list, paragrafami (znacznik <p>), blokami tekstu oddzielonymi dwoma lub więcej znakami nowej linii (dwa elementy
 następujące bezpośrednio po sobie) *etc.* Jeżeli dokument HTML zawierał listę publikacji, to w przetworzonym dokumencie tekstowym informacje o kolejnych publikacjach znajdą się w oddzielnych liniach.

W wyniku działania tego procesu otrzymywane są pliki, w których każdej z linii nadać można etykietę: PUBLIKACJA (jeżeli fragment zawiera dane o pojedynczej publikacji) lub INNA (jeżeli linia zawiera inne informacje). W przedstawionym problemie etykietowania duże znaczenie ma kolejność danych wejściowych. Publikacje

zazwyczaj występują w grupach następujących po sobie linii, stąd wystąpienie jednej linii z etykietą PUBLIKACJA zwiększa prawdopodobieństwo nadania takiej samej etykiety następnym fragmentom. W algorytmie crawlera, do automatycznego etykietowania publikacji w pobieranych zbiorach danych wykorzystano metodę statystyczną CRF, rodzaj klasyfikatora używanego w problemach wykrywania wzorców oraz przewidywania sekwencji. Podczas gdy popularne klasyfikatory (np. Naive Bayes) przewidują klasę obserwacji tylko na podstawie cech samej obserwacji, CRF wykorzystuje również informację o obserwacjach sąsiadujących. Przed sklasyfikowaniem wyników działania crawlera przez algorytm CRF, poszczególne linie w plikach tekstowych są zamieniane na sekwencje cech binarnych. Dopiero w takiej postaci algorytm nadaje im jedną z dwóch wymienionych wyżej klas. Cechy zostały dobrane w taki sposób, aby dobrze rozdzielały obserwacje dwóch różnych klas. Przykładowe cechy użyte w klasyfikatorze to:

- Czy linia zawiera nazwiska? (na podstawie pół miliona polskich i zagranicznych nazwisk)
- Czy linia jest dłuższa niż 1000 znaków? Czy linia jest krótsza niż 100 znaków?
- Czy linia zawiera rok?
- Czy linia zawiera skróty takie jak „vol.", „nr", „pp", „str." *etc.*
- Czy linia zawiera nazwę miasta lub państwa?

Opisany powyżej mechanizm został zaimplementowany w systemie, jednak wymaga jeszcze wielu testów. Problem pobierania danych naukowców z internetu jest na tyle rozległy i skomplikowany, że wykracza poza obszar tego opracowania. Z całą pewnością autorzy podzielą się wynikami dalszych prac w kolejnych publikacjach.

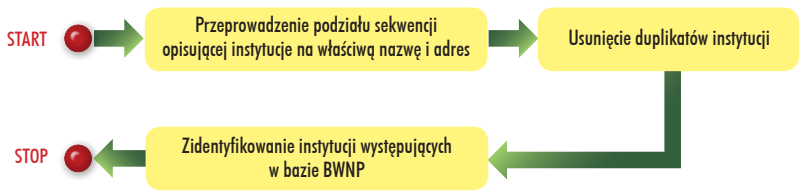
4. Procesy porządkujące

Dane pozyskane z internetu są nieraz zapisywane w różnej formie tekstowej, zatem część z nich może być powielana. Z tego powodu moduł wyposażono w narzędzia normalizujące bazę danych w zakresie usuwania duplikatów danych. Zaimplementowane zostały trzy procesy porządkujące: łączenie zduplikowanych instytucji, łączenie zduplikowanych publikacji oraz łączenie zduplikowanych źródeł.

4.1. Łączenie zduplikowanych instytucji

Polega ono na ekstrakcji właściwej nazwy instytucji, następnie pogrupowaniu instytucji według nazw w celu usunięcia duplikatów, a ostatecznie – przypisaniu instytucji do odpowiednich instancji w BWNP (rysunek 22). Sekwencje znaków opisujące instytucje do połączenia są afiliacjami publikacji. Najpierw, aby wydzielić nazwę, adres i język, w którym zapisana jest nazwa, przetwarza się je. Dodatkowo nazwa jest normalizowana, co polega na usuwaniu znaków diakrytycznych, zamianę znaków na duże litery, usunięcie znaków nie-alfanumerycznych. Następnie instytucje są grupowane według znormalizowanej nazwy. W tych grupach zachowywana jest tylko jedna instytucja, pozostałe są usuwane. Ostatecznie pozytywnie weryfikuje się każdą instytucję o nazwie odpowiadającej nazwie instytucji w BWNP w języku polskim lub angielskim.

Rysunek 22. Łączenie zduplikowanych instytucji

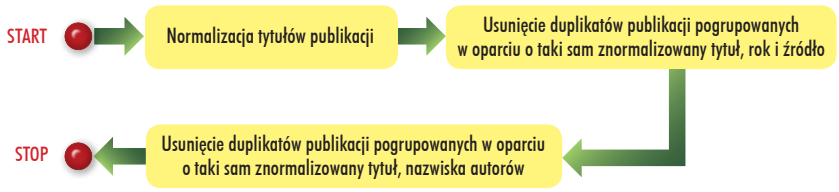


Źródło: opracowanie własne autorów

4.2. Łączenie zduplikowanych publikacji

W sposób iteracyjny przeglądane są wszystkie publikacje nieposiadające znormalizowanych tytułów, a potem dokonywana jest ich normalizacja. Doprowadza się do usunięcia znaków diakrytycznych, zmiany kodowania, zamiany znaków na *upper-cases* i usunięcia znaków nie-alfanumerycznych. Następnie tworzy się grupę duplikatów publikacji. Publikacje grupowane są dwustopniowo: najpierw według znormalizowanego tytułu, a następnie, w ramach powyższych zbiorów – na klastry o tym samym zestawie autorów. W drugim etapie opcjonalnie wykorzystywana jest informacja o roku wydania publikacji – gdy obie porównywane prace dysponują rokiem, to jest on brany pod uwagę w procesie weryfikacji zgodności. Grupy duplikatów publikacji są dzielone: na pojedynczą publikację do zachowania i pozostałe do usunięcia. Przed usunięciem duplikatów unikatowe dla nich informacje wykorzystuje się do zaktualizowania publikacji wybranej do zachowania. Proces ten pokazuje rysunek 23.

Rysunek 23. Łączenie zduplikowanych publikacji

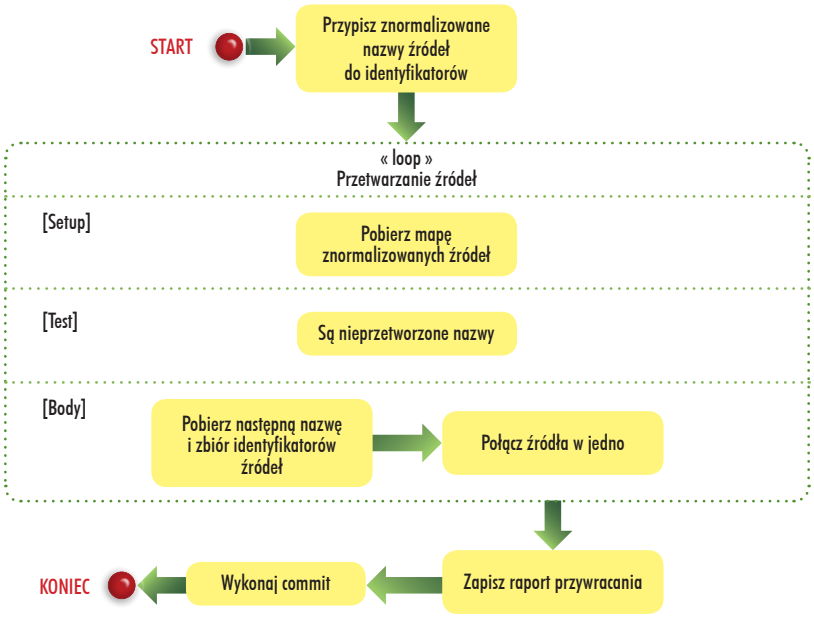


Źródło: opracowanie własne autorów

4.3. Łączenie zduplikowanych źródeł

Działanie algorytmu jest tutaj podobne do poprzednich. Nazwy źródeł publikacji są normalizowane, a następnie łączone w grupy. Zachowuje się tylko jedno źródło z grupy, a pozostałe usuwa, przy jednoczesnym uzupełnieniu informacji zachowywanego źródła, gdy któreś z usuwanych posiadało ich większy zakres. Widać to na rysunku 24.

Rysunek 24. Łączenie zduplikowanych źródeł

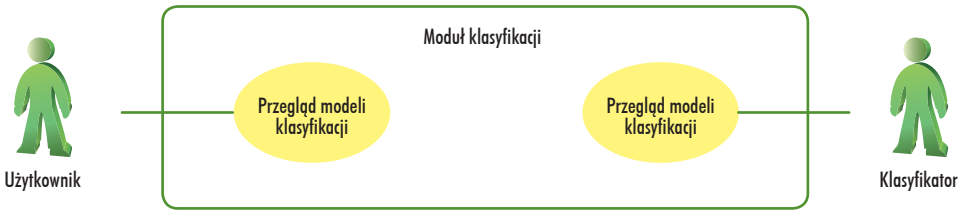


Źródło: opracowanie własne autorów

II. Moduł klasyfikacji

Moduł odpowiada za grupowanie danych według znanych kategorii nauki. Wypełnia dwie podstawowe funkcje: implementuje różne modele klasyfikacji nauki oraz dokonuje kategoryzacji dokumentów naukowych zgromadzonych w module zbierania danych według wybranego modelu klasyfikacji (rysunek 25).

Rysunek 25. Role i czynności użytkowników w module klasyfikacji



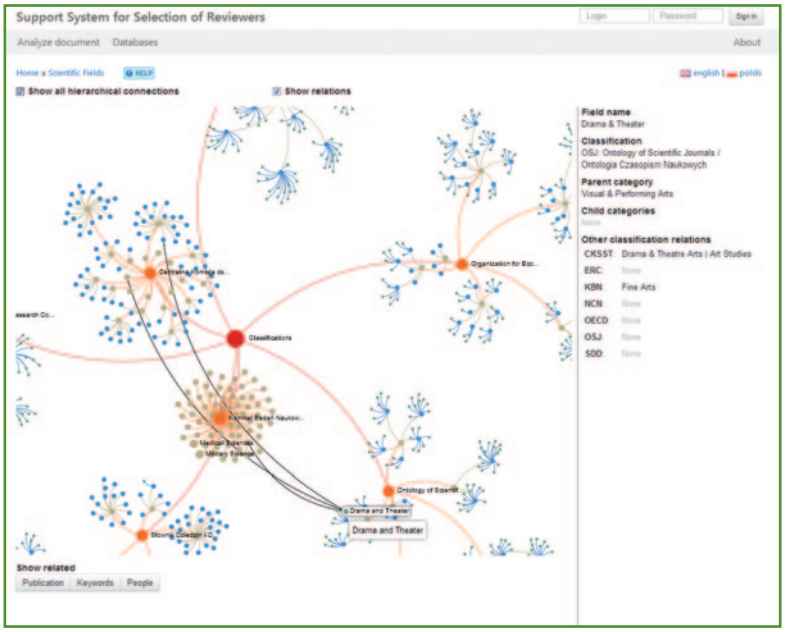
Źródło: opracowanie własne autorów

1. Modele klasyfikacji nauki

Rdzeń modułu to jedno-, dwu- lub trójpoziomowe modele klasyfikacji nauki pokazujące podział nauki na dziedziny/dyscypliny. W systemie uwzględniono siedem modeli klasyfikacji: KBN, ERC, OECD, NCN, SDD, CKSST, OSJ[68]. Wszystkie modele zostały powiązane ze sobą[69], co przekłada się na określenie, jak dziedziny w jednym modelu klasyfikacji są powiązane z dziedzinami w innych modelach. Poprzez interfejs WWW użytkownik ma dostęp do:

- przeglądania relacji hierarchicznych;
- przeglądania powiązań między modelami klasyfikacji;

Rysunek 26. Widok interfejsu przedstawiający powiązania między modelami klasyfikacji



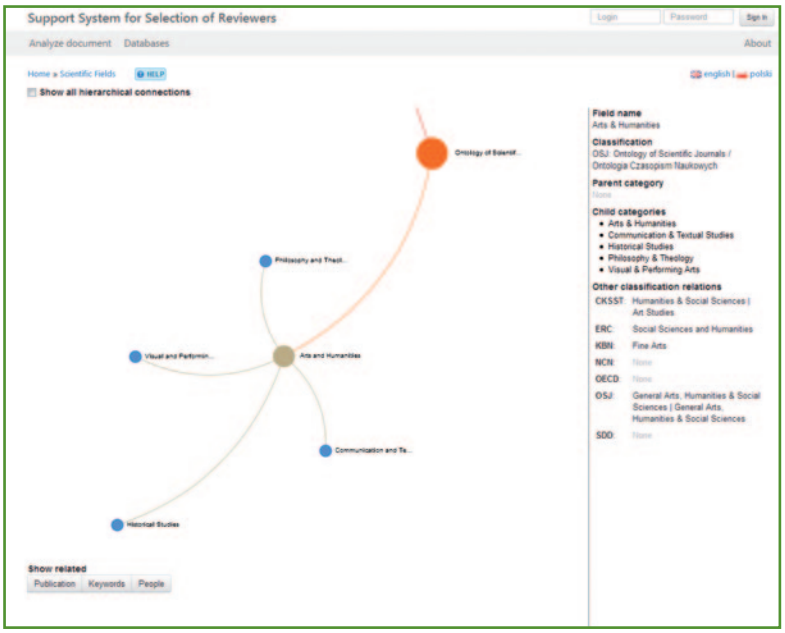
Źródło: System Wspomagania Wyboru Recenzentów, OPI

[68] Modele klasyfikacji nauki zostały przedstawione w tomie pierwszym.
[69] Powiązania modeli zostały opracowane w Ośrodku Przetwarzania Informacji – Instytucie Badawczym przez zespół laboratorium interaktywnych technologii.

- przeglądania publikacji powiązanych z wybraną dziedziną;
- przeglądania słów kluczowych powiązanych z wybraną dziedziną;
- przeglądania osób powiązanych z wybraną dziedziną.

Przykładowe widoki dostępne z poziomu użytkownika systemu poprzez interfejs WWW przedstawiają rysunki 26 i 27.

Rysunek 27. Widok interfejsu przedstawiający hierarchię wybranego modelu klasyfikacji



Źródło: System Wspomagania Wyboru Recenzentów, OPI

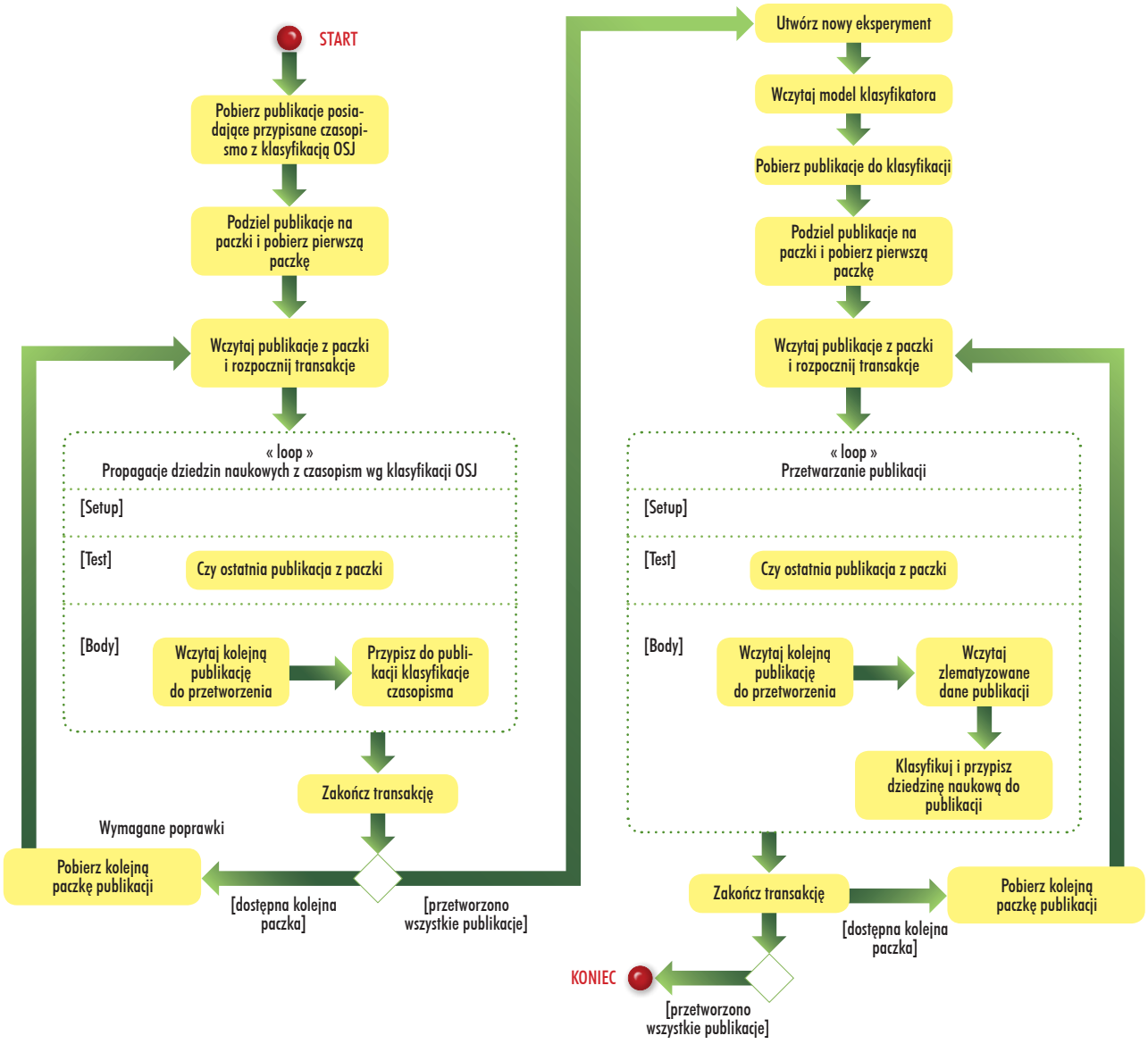
2. Klasyfikator

Drugim składnikiem tego modułu jest klasyfikator bayesowski[70]. Dane o recenzentach ograniczono obecnie do prac naukowych, zatem moduł dostosowano do danych o publikacjach. Docelowo możliwe będzie jego rozszerzenie w kontekście nowych zbiorów informacji.

Dane wejściowe dla kategoryzacji stanowi wektor zbudowany z tekstu abstraktu publikacji, tytułu oraz podanych przez autorów słów kluczowych. Kategoryzacja przypisuje publikacje (posiadające abstrakt lub słowa kluczowe) do dziedzin wybranego modelu klasyfikacji nauki. Ponieważ dla tego modelu dostępny był zbiór trenujący wymagany do zbudowania klasyfikatora, w obecnej wersji tym modelem jest klasyfikacja Ontology of Scientific Journals (OSJ). Klasyfikator pozwala na stosowanie innych modeli klasyfikacji, o ile będą dla nich dostępne zbiory uczące. Każdy z modeli zawiera różne poziomy, zaczynając od podstawowych kategorii (np. chemia) po poddziedziny (np. chemia organiczna). W ramach modułu klasyfikacji można wybrać odpowiedni poziom szczegółowości kategoryzacji. W ten sposób oznaczony dokument naukowy da się wykorzystać w kolejnych modułach, między innymi do identyfikacji osób. Ponadto każdy model klasyfikacji jest powiązany z pozostałymi, zatem w efekcie uzyskuje się przyporządkowanie publikacji i osób do każdego modelu. Dokładnie pokazuje to rysunek 28.

[70] Zagadnienia teoretyczne wyjaśniono w dodatku.

Rysunek 28. Klasyfikacja publikacji



Źródło: opracowanie własne autorów

3. Wyniki testów

Dane wejściowe dla kategoryzacji to wektor słów zbudowany poprzez połączenie tekstu obu abstraktów publikacji (polskiego i angielskiego), tytułu i podanych przez autorów słów kluczowych. Na tak zbudowanym wektorze dodatkowo usunięto słowa nieistotne (*stop words*), wykonano lematyzację oraz nadano wagi poszczególnym wyrazom przy zastosowaniu algorytmu TF-IDF. Eksperymenty przeprowadzono przy użyciu walidacji krzyżowej, natomiast do określania jakości klasyfikatora posłużono się dokładnością[71], czyli miarą mówiącą, jaki procent z klasyfikowanych publikacji otrzymał poprawną kategorię.

Wybór modelu klasyfikacji OSJ, która posiada trzy poziomy szczegółowości (nazywane później L0, L1 i L2, zaczynając od kategorii głównej), umożliwił przeprowadzenie eksperymentów na różnych jej poziomach.

[71] Zagadnienia teoretyczne wyjaśniono w dodatku.

Początkowo wykonano doświadczenia na najwyższym poziomie L0 dla różnych metod budowy zbioru uczącego. Należy dodać, iż model klasyfikacji na poziomie L0 zawiera sześć dziedzin naukowych:

- nauki ogólne;
- nauki medyczne;
- ekonomia i nauki społeczne;
- sztuka i nauki humanistyczne;
- nauki przyrodnicze;
- nauki stosowane.

Tabela 9. Skuteczność klasyfikacji w różnych modelach klasyfikatorów na poziomie kategorii głównych

Rodzaj klasyfikatora	Dokładność
1. Klasyfikator uwzględniający wszystkie kategorie główne przy proporcjonalnym rozkładzie (względem występowania w dostępnym korpusie danych) w zbiorze trenującym	61,29%
2. Klasyfikator uwzględniający wszystkie kategorie główne przy równym rozkładzie w zbiorze trenującym	63,39%
3. Klasyfikator z pominięciem kategorii „nauki ogólne" przy równym rozkładzie w zbiorze trenującym	74,05%
4. Klasyfikator z pominięciem kategorii „nauki ogólne" oraz połączonymi kategoriami „nauki przyrodnicze" i „nauki stosowane" przy równym rozkładzie w zbiorze trenującym	84,13%

Źródło: opracowanie własne autorów

Tabela 10. Skuteczność klasyfikacji w obrębie poszczególnych par kategorii głównych

Porównywane pary kategorii głównych	Dokładność
Nauki medyczne – sztuka i nauki humanistyczne	95,58%
Ekonomia i nauki społeczne – nauki przyrodnicze	94,66%
Nauki medyczne – ekonomia i nauki społeczne	94,18%
Sztuka i nauki humanistyczne – nauki stosowane	94,63%
Sztuka i nauki humanistyczne – nauki przyrodnicze	93,90%
Nauki medyczne – nauki przyrodnicze	92,19%
Ekonomia i nauki społeczne – nauki stosowane	90,88%
Nauki medyczne – nauki stosowane	89,32%
Ekonomia i nauki społeczne – sztuka i nauki humanistyczne	86,42%
Nauki przyrodnicze – nauki stosowane	70,43%

Źródło: opracowanie własne autorów

Dla dwóch pierwszych eksperymentów przedstawionych w tabeli 9 uzyskano dość niską skuteczność, z przewagą podejścia z równym rozkładem zbioru uczącego. Z tego powodu podjęto decyzję o usunięciu kategorii „nauki ogólne" (publikacje w tym zbiorze zbyt często przenikają się tematycznie z pozostałymi dziedzinami) oraz zastosowaniu w kolejnych testach zbioru uczącego z równym rozkładem dziedzinowym. Takie działanie

zapewniło przyrost jakości klasyfikatora o 10%, do 74,05% (punkt 3 w tabeli 9). Aby poprawić jakość klasyfikacji, wykonano również operację łączenia najbardziej przenikających się kategorii z poziomu L0. W celu sprawdzenia, które z nich są najściślej powiązane, zbudowano klasyfikator dla każdej pary kategorii z osobna. Wyniki skuteczności klasyfikacji znajdują się w tabeli 10; wyraźnie widać, że najbardziej przenikają się kategorie „nauki przyrodnicze" i „nauki stosowane". Po złączeniu tych dwóch dziedzin, jakość klasyfikatora na poziomie L0 zwiększyła się o kolejne 10% – do 84,13%. Na tym etapie zakończono ewaluację testów na najwyższym poziomie.

Kolejny krok to doświadczenia na poziomie L1, które opierały się na wynikach otrzymanych przez model klasyfikatora na poziomie kategorii głównych. Model zbudowany z niezależnych klasyfikatorów na poziomie L0 i L1 nazywamy klasyfikatorem hierarchicznym. Do przeprowadzenia doświadczeń na poziomie podkategorii konieczne było – z racji małej liczby publikacji – ręczne przyporządkowanie instancji trenujących w poddziedzinach kategorii głównej „sztuka i nauki humanistyczne". To działanie pozwoliło też zwiększyć zbiór trenujący dla modelu klasyfikatora na poziomie L0, co wpłynęło na poprawę skuteczności klasyfikacji.

Podobnie jak to miało miejsce w eksperymentach dla zbioru opartego o kategorie główne, również w klasyfikacji na poziomie L1 niezbędne było wykonanie połączeń między przenikającymi się tematycznie dziedzinami w ramach jednej kategorii głównej. Po testach par dla poszczególnych kategorii głównych uzyskano następujące mapowania:

- nauki medyczne – połączenie dwóch podkategorii: „nauki biomedyczne" i „medycyna kliniczna";
- nauki przyrodnicze – połączenie dwóch podkategorii: „biologia" oraz „nauki o Ziemi i środowisku";
- nauki stosowane – połączenie trzech podkategorii: „budowa i projektowanie środowiska", „technologie strategiczne i innowacyjne" oraz „inżynieria";
- ekonomia i nauki społeczne – brak połączeń;
- sztuka i nauki humanistyczne – brak połączeń.

Wyniki w tabeli 11 wyraźnie wskazują na zwiększenie jakości klasyfikacji poprzez mapowanie między dziedzinami o podobnej tematyce.

Tabela 11. Skuteczność klasyfikacji na poziomie L1 dla poszczególnych kategorii głównych

Nazwa dziedziny, na podstawie której zbudowano model klasyfikatora na poziomie jej podkategorii	Dokładność	
	Bez połączonych podkategorii	Z połączonymi podkategoriami
Nauki medyczne	74,00%	82,00%
Nauki przyrodnicze	84,00%	90,00%
Nauki stosowane	74,00%	88,00%
Ekonomia i nauki społeczne	84,00%	84,00%
Sztuka i nauki humanistyczne	82,00%	82,00%

Źródło: opracowanie własne autorów

Zbudowanie jednego klasyfikatora dla podkategorii dziedzin „nauki przyrodnicze" i „nauki stosowane" skutkuje uzyskaniem dokładności na poziomie 75%, mimo scalania podkategorii w ramach dziedzin. Jest to wynik dość znacznie odbiegający od jakości wyizolowanych klasyfikatorów dla obu wymienionych dziedzin. Biorąc ten fakt pod uwagę, podjęto próbę wprowadzenia klasyfikatora dyskryminującego (*voting classifier*), gdzie głosowanie odbywa się za pomocą wielkości prawdopodobieństwa. Ten model oparto na dwóch klasyfikatorach – oddzielnie dla podkategorii nauk przyrodniczych i nauk stosowanych. Publikacja jest ety-

kietowana przez obydwa klasyfikatory niezależnie, a finalną kategorię na poziomie L1 uzyskuje ta z większym prawdopodobieństwem. Zadbano o uczenie pojedynczych klasyfikatorów na zbiorach o porównywalnych liczbach dokumentów. Dla zastosowanego podejścia, przy wielkości zbioru uczącego równego 20 tysięcy wektorów dokumentów, wyniki oscylują wokół 40%, co dyskryminuje ten model klasyfikatora.

Ostatecznie końcową postacią klasyfikatora jest model hierarchiczny. Do procesu uczenia pobierane są w nim publikacje o równym rozkładzie dla zmapowanych kategorii głównych, a następnie dla każdej z nich buduje się osobny klasyfikator z uprzednio zdefiniowanymi mapowaniami dla podkategorii. Skuteczność klasyfikacji na poziomie kategorii głównych wynosi tutaj 87,6%, a na poziomie podkategorii 72,5%.

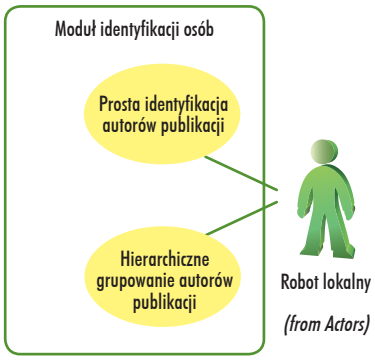
Następnie w bazie danych zapisuje się kategorie główne oraz podkategorie. W kategoriach zmapowanych zapisywane są wszystkie pierwotne dziedziny naukowe.

III. Moduł identyfikacji osób

Zgromadzone przez moduł zbierania danych informacje o publikacjach[72] pochodzą z heterogenicznych źródeł, nie istnieje zatem żadne oczywiste powiązanie między autorami występującymi w poszczególnych publikacjach a rekordami osób znajdujących się w systemie. Najczęściej źródła umieszczają bardzo ograniczone informacje o autorach, sprowadzające się do wymienienia inicjałów imion oraz nazwiska autora. Dodatkowym utrudnieniem jest również możliwość występowania kilku osób o identycznych nazwiskach i inicjałach lub imionach. Ponieważ zbiór publikacji podpisanych tym samym imieniem i nazwiskiem może być autorstwa kilku różnych osób, przyporządkowanie publikacji do rzeczywistego profilu naukowca staje się zadaniem niebanalnym.

Zadaniem modułu identyfikacji osób jest przypisanie autorstwa publikacji do naukowców znajdujących się w Bazie Wiedzy o Nauce Polskiej lub utworzenie profili naukowców niewystępujących w BWNP i przypisanie im publikacji. Identyfikacja odbywa się za pomocą dwóch algorytmów: prostej identyfikacji i grupowania hierarchicznego. Pierwszy przypisuje publikacje tylko do autorów z BWNP, drugi natomiast ma możliwość tworzenia nowych profili naukowców i dodawania do nich publikacji (rysunek 29).

Rysunek 29. Moduł identyfikacji osób



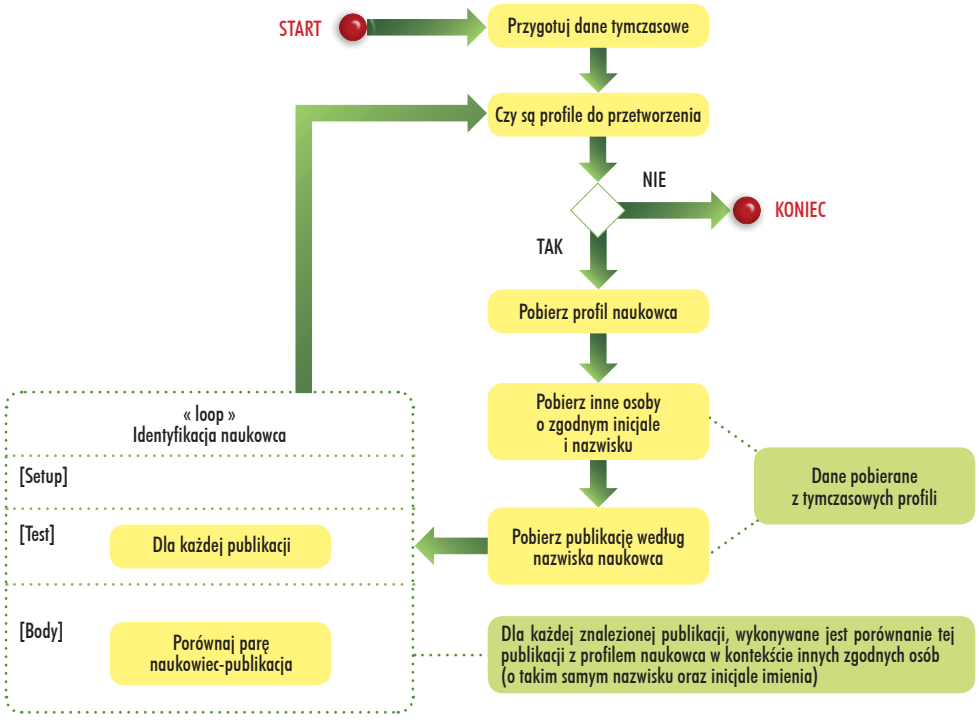
Źródło: opracowanie własne autorów

1. Prosta identyfikacja autorów publikacji

Identyfikacja prosta opiera się na danych pochodzących z Bazy Wiedzy o Nauce Polskiej. Działanie algorytmu polega na porównywaniu identyfikowanych publikacji z poszczególnymi osobami i wyszukiwaniu podobieństw na podstawie ściśle zdefiniowanych reguł. W tym momencie algorytm wykorzystuje pięć reguł

[72] Inne dane o potencjalnych recenzentach nie będą rozpatrywane na tym etapie rozwoju systemu.

Rysunek 30. Prosta identyfikacja publikacji



Źródło: opracowanie własne autorów

identyfikacji, uruchamianych sekwencyjnie (rysunek 30). Jeżeli jednej z reguł uda się zweryfikować autorstwo publikacji, proces jest przerywany i następne reguły nie wykonują się. W przeciwnym wypadku sprawdzone zostaną kolejno wszystkie reguły. Jeżeli żadna z nich nie będzie wystarczająca do wskazania autora, publikacja pozostanie nierozpoznana. Algorytm może być uruchamiany przyrostowo, a każde kolejne uruchomienie korzysta z danych przypisanych przez poprzednią identyfikację. Dzięki temu realne jest zidentyfikowanie nowych publikacji nawet wtedy, gdy nie pojawiły się żadne nowe dane w bazie. Owe reguły stwierdzające, kiedy autor publikacji zostaje uznany za zweryfikowanego to:

1. Istnienie w BWNP unikatowego imienia i nazwiska oraz zawarcie w publikacji osoby o tym samym pełnym imieniu i nazwisku.
2. Porównanie współautorów publikacji i osób powiązanych z naukowcem przez prace badawcze oraz autorów identyfikowanej publikacji. Jeżeli osoba jest powiązana z inną osobą lub kilkoma osobami, które występują na liście współautorów oraz gdy nie istnieje żaden inny naukowiec o takim inicjale i nazwisku, który jest powiązany z tymi osobami, autorstwo publikacji zostaje zweryfikowane.
3. Porównywanie afiliacji naukowca i afiliacji publikacji. Autorstwo przypisuje się osobie, która posiada najwięcej zgodnych afiliacji.
4. Sprawdzanie zgodności dziedziny naukowej określonej osoby i dziedziny przypisanej do publikacji.
5. Dopasowanie publikacji do najbliższego jej profilu naukowca. Profil naukowca to wektor ważonych słów kluczowych, pozyskanych z tytułów publikacji, których autorem jest dana osoba. Analizowana publikacja, a dokładniej jej tytuł, przekształcana jest do postaci wektora słów, który następnie porównuje się z profilami potencjalnych autorów. Porównanie polega na obliczaniu podobieństwa wektora publikacji z wektorem profilu naukowca. Dopasowanie mierzy się tylko w kontekście słów zgodnych, które są unikatowe w ramach jednego profilu oraz występują co najmniej w dwóch publikacjach badanej osoby. Miara dopasowania to suma wag zgodnych słów z badanego profilu. Ostatecznie o zgodności publikacji z analizowaną osobą decyduje również przekroczenie minimalnego poziomu wsparcia miary dopasowania, a także brak większego podobieństwa wśród alternatywnych autorów.

2. Wyniki testów prostej identyfikacji

Po zaimplementowaniu i wdrożeniu algorytmu prostej identyfikacji, przeprowadzono weryfikację wyników algorytmu, z pominięciem reguły piątej. Konieczne było ręczne sprawdzenie, czy autorzy przypisani przez algorytm identyfikacji do osób w BWNP zostali powiązani zgodnie z rzeczywistym autorstwem publikacji. Ze względu na liczbę autorów (około 4,75 mln publikacji zawierających około 19 mln wpisów autorów) oraz czasochłonność manualnej identyfikacji, niemożliwe było zweryfikowanie wszystkich danych – należało wybrać możliwie reprezentatywną próbkę. Ostatecznie zbiór testowy został wybrany w następujący sposób:

- wybrano po jednym naukowcu z każdej dziedziny KBN i zweryfikowano wszystkie publikacje tego naukowca – łącznie publikacje dla 63 naukowców z nauk ścisłych i humanistycznych;
- wybrano popularne nazwiska i inicjały, dla których mogły występować silne niejednoznaczności podczas procesu identyfikacji. Wyłoniono cztery grupy naukowców, z których każda zawierała co najmniej 12 różnych osób o identycznym nazwisku i inicjale (według BWNP); następnie zidentyfikowano wszystkie publikacje, w których pojawiał się autor o takich danych. W ten sposób zweryfikowano publikacje ponad 48 osób.

Łącznie zweryfikowano 2921 publikacji, wśród których znalazły się 34 błędnie przyporządkowane. Pomyłki zazwyczaj nie występowały pojedynczo, najczęściej zdarzało się kilka błędów dotyczących tych samych autorów. W kilku przypadkach autorem publikacji była osoba niewystępująca w BWNP, natomiast identyfikacja przypisała publikację do znanego naukowca z bazy (trzy osoby). Zdarzały się też sytuacje, że autor podpisany był w publikacji jedynie swoim drugim imieniem, co wprowadzało w błąd regułę wykorzystującą porównywanie unikatowych imion (dwie osoby). Przy jednej osobie, w BWNP istniały nieprawidłowo przypisane publikacje, a to spowodowało kolejne błędy identyfikacji w algorytmie opierającym się na tych danych. Jeden z błędów dotyczył przypadkowej zbieżności nazwisk współautorów dla dwóch różnych osób o takim samym inicjale i nazwisku.

Pomimo tych pomyłek, na zweryfikowanych danych algorytm osiągnął precyzję oraz kompletność[73] odpowiednio na poziomie 98,22% oraz 64,75%. Kompletność z pewnością udałoby się poprawić poprzez wielokrotne uruchamianie algorytmu – publikacje przypisane w jednej iteracji mogą być wykorzystane w regułach przy kolejnej. Dotyczy to przede wszystkim reguły współautorstwa, opierającej się na autorach z publikacji, które zostały przypisane do osoby. Im więcej takich publikacji, tym więcej potencjalnych współautorów można powiązać z daną osobą. Celem algorytmu regułowego było osiągnięcie jak najwyższej precyzji, aby jego wyniki mogły posłużyć do zbudowania dużych zbiorów uczących dla bardziej zaawansowanych algorytmów.

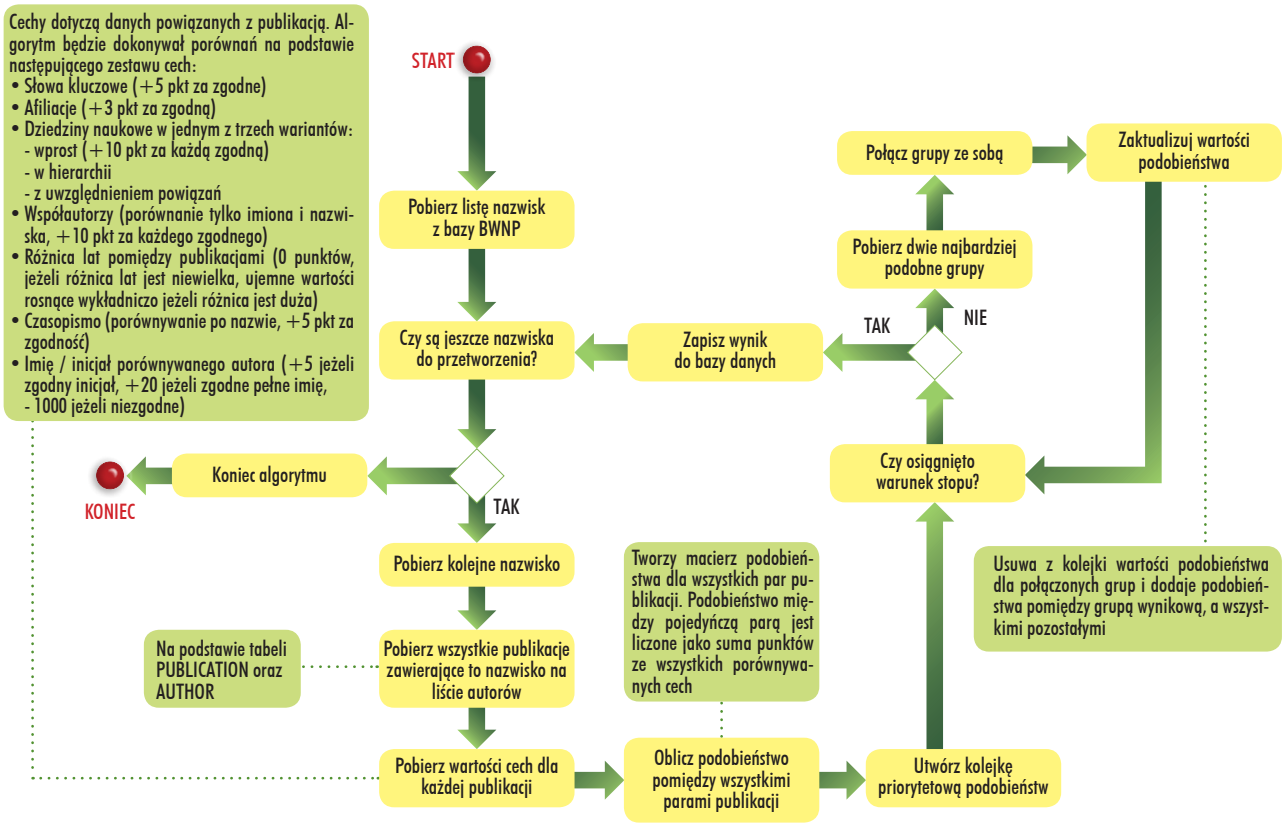
Po dodaniu reguły piątej do algorytmu, dodatkowo zweryfikowano poprawność identyfikacji tylko dla tej reguły. Wybrano sto publikacji, które zawierały autorów zidentyfikowanych przez regułę. Publikacje przypisano do ośmiu różnych osób w BWNP. Spośród tych publikacji, 18 zidentyfikowano błędnie (precyzja – 82%). Pomyłki związane były przede wszystkim z nierównomiernym rozkładem liczby publikacji między naukowcami o takim samym inicjale i nazwisku. W razie niejednoznaczności, reguła preferowała naukowców z przypisaną większą liczbą publikacji, co było równoznaczne z większą liczbą słów w profilu tej osoby. Zwiększało to prawdopodobieństwo, że w tytule porównywanej publikacji znajdą się słowa unikatowe w ramach profilu. Dodatkowe publikacje były więc błędnie przypisywane do naukowców z dużą liczbą publikacji, co pociągało za sobą kolejne błędy. Reguła przyczyniła się do zmniejszenia ogólnej precyzji algorytmu, nie została więc jeszcze wdrożona w systemie. Po wprowadzeniu poprawek, na przykład dodaniu wag rozróżniających słowa bardziej i mniej znaczące, przeprowadzone zostaną kolejne eksperymenty. Osiągnięte wyniki wpłyną na decyzję o dodaniu reguły do kolejnej wersji systemu.

3. Hierarchiczne grupowanie autorów publikacji

Drugim z algorytmów jest algorytm grupowania hierarchicznego (*Hierarchical Agglomerative Clustering*, HAC[74]), jeden z popularnych algorytmów uczenia bez nadzoru, przedstawiony na rysunku 31. Jego cel to

[73] Zagadnienia teoretyczne wyjaśniono w dodatku.
[74] Zagadnienia teoretyczne wyjaśniono w dodatku.

Rysunek 31. Algorytm grupowania hierarchicznego



Źródło: opracowanie własne autorów

wydzielenie – ze zbioru obiektów posiadających pewne cechy – grup obiektów do siebie podobnych. W module identyfikacji osób obiektami są publikacje, a utworzone grupy reprezentują profile naukowców, którzy są autorami prac znajdujących się w ich grupie. Podobieństwo między obiektami w HAC może być definiowane w dowolny sposób. Moduł identyfikacji osób wykorzystuje znajomość dziedziny problemu, podobieństwo jest więc związane z prawdopodobieństwem, że dwie publikacje zostały napisane przez tego samego badacza. Podobieństwo określa liczba punktów, która może być wartością ujemną (jeżeli występują duże różnice pomiędzy publikacjami), dodatnią (jeżeli występują podobieństwa) lub zerem (jeżeli podobieństwa i różnice się równoważą lub gdy cech nie można ze sobą porównać). Podczas liczenia podobieństwa porównywane są: słowa kluczowe, afiliacje, dziedziny naukowe, współautorzy (tylko imiona i nazwiska), różnica lat między publikacjami (0 punktów – gdy różnica jest niewielka, ujemne wartości rosnące wykładniczo – gdy różnica jest duża), czasopismo, imię lub inicjał, nazwisko porównywanego autora.

W kolejnych krokach algorytmu łączone są ze sobą najbardziej podobne publikacje. Algorytm rozpoczyna więc od najwyższych wartości podobieństwa i z każdą kolejną iteracją wartość dla łączonych publikacji jest coraz mniejsza. Jeżeli podobieństwo osiągnie ustaloną wartość minimalną, algorytm jest przerywany, a jego wynikiem są grupy publikacji. Następnie każdą grupę przekształca się we wstępny profil naukowca.

Dalej przedstawiony zostanie szczegółowy opis algorytmu identyfikacji autorów wykorzystywanego przez system wspomagania wyboru recenzentów.

3.1. Identyfikacja autorów

Zaproponowana w systemie metoda identyfikacji autorów to zmodyfikowana wersja algorytmu grupowania hierarchicznego. Obiektami grupowanymi są publikacje, natomiast utworzone w wyniku działania algorytmu grupy zostaną jednoznacznie przypisane do konkretnych autorów. Ostatecznym celem jest utworzenie w bazie danych profili pracowników naukowych; każdemu z nich przyporządkowane zostaną publikacje, których rzeczywiście jest twórcą. Ze względu na bardzo dużą liczbę publikacji, cały proces jest wieloetapowy, a podczas jego działania algorytm identyfikacji uruchamia się wielokrotnie. Pojedyncze uruchomienie działa w kontekście konkretnego nazwiska pobranego z bazy danych. Aby przetworzyć wszystkie artykuły i utworzyć profile dla wszystkich autorów znajdujących się w bazie, metoda identyfikacji musi zostać uruchomiona dokładnie tyle razy, ile unikatowych nazwisk znajduje się aktualnie w tabeli autorów. Cały proces tworzenia profili można zapisać w postaci następujących kroków:

1. Ze zbioru wszystkich naukowców pobierz nazwisko kolejnej osoby.
2. Dla wybranego nazwiska pobierz wszystkie publikacje, które zawierają autora o takim nazwisku. Pobierz wszystkie publikacje o autorach posiadających nazwisko takie jak przetwarzany naukowiec.
3. Uruchom algorytm grupowania hierarchicznego dla wybranych publikacji, gdzie autor o wybranym nazwisku będzie traktowany jako „autor główny", natomiast pozostali autorzy jako „autorzy drugorzędni":
   - oblicz podobieństwa między publikacjami;
   - w kolejnych krokach łącz najbardziej podobne do siebie publikacje bądź ich grupy;
   - zakończ w momencie, gdy utworzona zostanie pojedyncza grupa;
   - utwórz dendrogram i ogranicz go do określonego poziomu odcięcia na wybranym dla algorytmu poziomie odcięcia;
   - stwórz w bazie wiersze odpowiadające konkretnym pracownikom naukowym i przypisz im publikacje na podstawie wydzielonych przez algorytm grup;
   - jeżeli istnieją w bazie nieprzetworzone jeszcze nazwiska, powróć do punktu 1.

Jak wynika z powyższego opisu, ta sama publikacja może być wykorzystana przez algorytm identyfikacji wielokrotnie – dokładnie raz dla każdego współautora, który jest zapisany w bazie danych. Za każdym razem tylko jeden z autorów będzie oznaczony przez algorytm jako „autor główny". Pojęcia „autor główny" oraz „autorzy drugorzędni" związane są ze sposobem, w jaki traktowana jest lista autorów przez metodę identyfikacji. Autor główny to osoba, której publikacje są identyfikowane podczas aktualnego uruchomienia algorytmu. Autorzy drugorzędni są w trakcie jego działania wykorzystani podczas liczenia podobieństw między publikacjami (sposób wyliczania podobieństw zostanie opisany w dalszej części rozdziału).

W wyniku działania zaproponowanej metody powstają profile pracowników naukowych. Jeżeli do osoby przypisane zostały publikacje pochodzące z Bazy Wiedzy o Nauce Polskiej, istnieje jednoznaczna relacja pomiędzy utworzonym profilem a danymi osoby w tej bazie. W przeciwnym wypadku, gdy stworzony profil nie wskazuje na osobę pochodzącą z BWNP, co dotyczy przede wszystkim naukowców zagranicznych, którzy znaleźli się na listach współautorów lub posiadają polskie nazwisko, może on funkcjonować samodzielnie, nie wskazując na dane w BWNP.

3.2. Podobieństwo między publikacjami

W algorytmie identyfikacji osób podstawową miarą wykorzystywaną do grupowania publikacji jest podobieństwo w postaci liczby całkowitej oznaczającej liczbę punktów. Publikacje są porównywane parami, ze względu na kilka cech. Im większa jest liczba punktów dla danej pary publikacji, tym bardziej są one do siebie podobne. Podobieństwo może przyjmować zarówno wartości dodatnie, jak i ujemne. Każda cecha ma wpływ na ogólną wartość punktową podobieństwa. Jeżeli publikacje są do siebie podobne ze względu na porównywaną cechę, do całkowitego podobieństwa dodawane są punkty. Jeżeli występują różnice, podobieństwo jest zmniejszane o pewną liczbę punktów. Jeżeli natomiast publikacji nie można porównać lub wartość cechy nie jest znacząca dla ogólnego podobieństwa, punkty się nie zmieniają. W szczególnym przypadku punkty

nie ulegają zmianie także, gdy wartość cechy jest nieokreślona lub brakująca dla dowolnej z porównywanych publikacji.

Ostateczna wartość podobieństwa dwóch publikacji wyliczana jest jako suma wszystkich cech punktów otrzymanych z podobieństw składowych. Wartości punktowe przypisywane do ogólnego podobieństwa mogą być różne w zależności od tego, jakie są wartości cechy w porównywanych publikacjach. Przykładowo, podczas porównywania imion osób istotność informacji zależy od wyniku porównania. Jeżeli imiona są różne, można prawie na pewno wykluczyć, że mamy do czynienia z tą samą osobą. Z kolei gdy imiona są takie same, informacja ta nie jest już tak ważna – istnieje bowiem wiele różnych osób o identycznych imionach. W pierwszym przypadku waga informacji powinna więc być znacznie wyższa niż w drugim. W algorytmie identyfikacji wartości punktowe zostały dobrane subiektywnie, na podstawie oceny istotności informacji wynikającej z wartości cech. Punkty te zostały następnie nieznacznie skorygowane po przeprowadzeniu wstępnych eksperymentów.

W kolejnych wersjach systemu planowane są eksperymenty dotyczące doboru optymalnych wartości punktowych z wykorzystaniem algorytmów uczenia maszynowego, a dokładniej klasyfikatorów. Odbywać się to będzie poprzez wytrenowanie klasyfikatora na podstawie zbioru uczącego zbudowanego na parach publikacji, wybranych w losowy sposób. Następnie wagi przypisane konkretnym cechom w modelu klasyfikatora należy odpowiednio przekształcić, zgodnie z użytym klasyfikatorem, tak aby można było wyznaczać podobieństwa między grupami publikacji.

Poniżej zostały przedstawione cechy, które algorytm wykorzystuje podczas porównywania publikacji oraz sposoby wyliczania wartości punktowych dla tych cech.

Imiona głównego autora. Wartości punktowe wyliczane są na podstawie porównywania imion autorów oznaczonych jako „autorzy główni". Nazwisko jest oczywiście ignorowane, ponieważ w algorytmie porównywane będą tylko osoby o takim samym nazwisku. Jeżeli dowolna z publikacji nie zawiera pełnego imienia, ale inicjał autora, to wartość zostanie wyliczona poprzez porównanie inicjałów. W przypadku zgodnego pierwszego imienia cecha otrzymuje $+10$ punktów, w przypadku zgodnego inicjału $+5$ punktów. Występowanie różnych imion lub inicjałów wyklucza, że publikacje zostały napisane przez tego samego autora. Wtedy do całkowitego podobieństwa doliczana jest kara -1000 punktów.

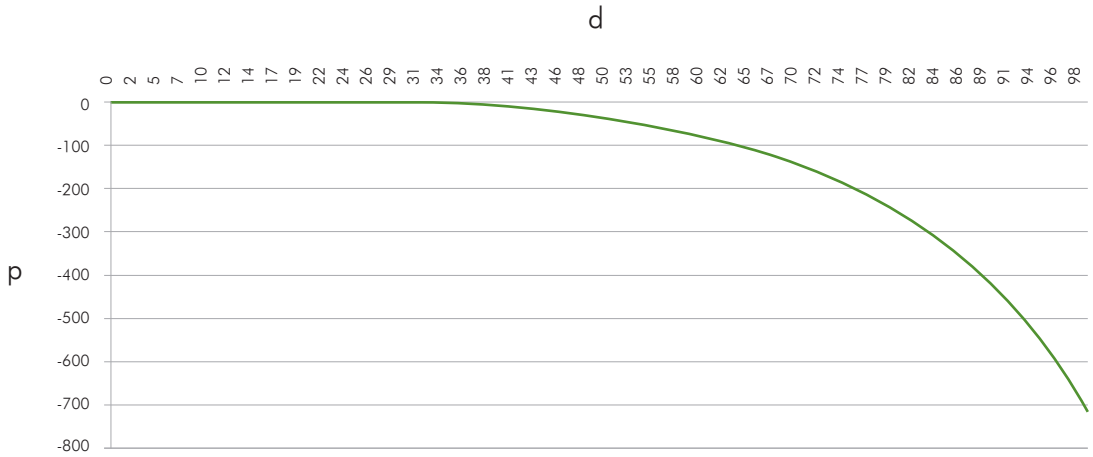
Rok publikacji. Liczba lat dzielących od siebie dwie publikacje to kolejna przesłanka, która może posłużyć do określenia, czy zostały one napisane przez tego samego autora. W algorytmie założono, że publikacje są do siebie podobne, jeżeli opublikowano je w podobnych latach. W przypadku tej cechy liczba punktów jest zależna od różnicy lat, w których opublikowane zostały artykuły. W przedziale od 0 do 32 lat cecha w żaden sposób nie wpływa na całkowite podobieństwo. Od 33 lat wzwyż od podobieństwa odejmowane są punkty. Liczba punktów wzrasta wykładniczo wraz ze zwiększającą się różnicą lat, zgodnie ze wzorem:

$$p = m_d \cdot (-e^{d \cdot w_d} + c_d)$$

gdzie:
d – różnica lat między publikacjami;
m_d, c_d i w_d – stałe, którym nadano wartości odpowiednio $m_d = 5$, $c_d = 5$, $w_d = 0{,}05$.

Wartości stałych zostały dobrane w taki sposób, aby dla różnicy lat mniejszej od 30 nie występowała żadna kara punktowa, natomiast dla wartości 100 kara ta wynosiła około -1000. Pokazuje to rysunek 32.

Rysunek 32. Wykres kary punktowej dla różnicy lat pomiędzy publikacjami



Źródło: opracowanie własne autorów

Czasopismo. Podobieństwo opiera się na porównywaniu czasopisma naukowego, w którym artykuły zostały opublikowane. Jeżeli artykuły opublikowano w tym samym czasopiśmie, do całkowitego podobieństwa doliczanych jest 5 punktów. W przeciwnym razie żadne punkty nie są doliczane ani odejmowane.

Słowa kluczowe, współautorzy, afiliacje i dziedziny naukowe. Występowanie wspólnych słów kluczowych sugeruje, że dwie publikacje mogą dotyczyć podobnej problematyki. Podobnie jest z występowaniem tych samych dziedzin. Te same nazwiska pojawiające się na liście współautorów oraz takie same afiliacje również wskazują na istnienie pewnego powiązania porównywanych publikacji. Ponieważ każde z podobieństw składowych dla tych cech jest liczone w podobny sposób, wszystkie zostaną tu opisane wspólnie. Wszystkie sprowadzają się do porównywania ze sobą dwóch list, a liczba punktów dodanych do ogólnego podobieństwa zależy od liczby zgodnych obiektów występujących w tych listach. Różnice występują jedynie w definicji równości obiektów oraz w liczbie punktów przyznawanej za każdy zgodny obiekt. Afiliacje są uznawane za równe, jeżeli mają identyczne nazwy (bez uwzględniania wielkości liter). Słowa kluczowe muszą mieć identyczne nazwy oraz taki sam język. Autorzy są porównywani według inicjału pierwszego imienia oraz pełnego nazwiska, dziedziny natomiast – według przypisanych im numerów identyfikacyjnych. Dla wszystkich wymienionych cech, do ogólnego podobieństwa dodaje się pewną liczbę punktów za każdy zgodny obiekt. Dla autorów i dziedzin naukowych jest to 10 punktów, dla słów kluczowych – 5 punktów, a dla afiliacji – 3 punkty.

3.3. Wstępnie przypisane publikacje

Część publikacji biorących udział w algorytmie identyfikacji autorów pochodzi z Bazy Wiedzy o Nauce Polskiej, która zawiera własne profile uczonych oraz wybrane publikacje im przypisane. Jest to wiedza pewna, która może być wykorzystana przez algorytm grupowania. Metoda identyfikacji została więc zmodyfikowana w taki sposób, aby jako dane wejściowe poza zbiorem publikacji mogła przyjmować informacje z BWNP. Następnie dane te są przekształcane przez algorytm we wstępne grupy, tworzone jeszcze przed obliczeniem macierzy porównań. Jeśli więc wśród danych wejściowych znajdą się publikacje pochodzące z wewnętrznych źródeł danych OPI, będą one pogrupowane przed rozpoczęciem działania algorytmu. Dodatkowo, podczas tworzenia macierzy porównań wartość podobieństwa między publikacjami jest ustalana na zero, jeśli znalazły się one w różnych tak zainicjowanych grupach. Wykorzystanie danych z BWNP wpływa również na działanie metod aglomeracyjnych. Każda grupa zawierająca te publikacje otrzymuje unikatową etykietę. Podobieństwa pomiędzy grupami posiadającymi różne etykiety jest zawsze równe 0. Oznacza to, że wstępnie zainicjowa-

ne grupy nie mogą łączyć się między sobą, a jedynie z publikacjami lub grupami publikacji pochodzącymi z zewnętrznych źródeł danych.

4. Wyniki testów grupowania hierarchicznego

Celem tego algorytmu było wydzielenie grup publikacji posiadających podobne cechy. Takie zbiory reprezentują profile naukowców, którzy są twórcami publikacji znajdujących się w grupie, przy czym grupa może zawierać publikacje tylko jednej osoby. Dane wejściowe to publikacje pobrane z BWNP, mające zidentyfikowanych autorów, a także sprawdzone ręcznie podczas weryfikacji jakości metodą prostej identyfikacji osób. Sprawdzenie skuteczności algorytmu możliwe było dzięki wykorzystaniu przyporządkowań osób do publikacji uzyskanych w prostej identyfikacji osób, z racji wysokiej precyzji tej metody (ponad 98%). Weryfikacja odbyła się na zasadzie porównania zidentyfikowanych autorów publikacji uzyskanych w metodzie prostej identyfikacji do tych otrzymanych po wykonaniu algorytmu grupowania hierarchicznego.

Eksperymenty wykonano dla 50 losowo wybranych nazwisk, dla których liczba osób w BWNP wynosi od 100 do 200. Przy takich założeniach, zbiór weryfikujący zawierał około 20 tysięcy publikacji, z czego w prostej identyfikacji udało się zidentyfikować 5121. W ten sposób dobrana próbka umożliwiła weryfikację przyporządkowań publikacji do osób, dla których mogą występować potencjalne błędy. Dla algorytmu grupowania hierarchicznego ustalono liczbę minimalnego rozmiaru grupy, wynoszącą trzy publikacje. Dodatkowo doświadczenia sprawdzały, jak parametr odcięcia (średnie podobieństwo w grupie) wpływa na poprawność identyfikacji.

Tabela 12. Wyniki eksperymentów algorytmu grupowania hierarchicznego dla różnych wartości parametru odcięcia

Parametr odcięcia	Precyzja (precision)	Kompletność (*recall*)	Miara F (*F1 Measure*)
5	97,96	54,27	69,84
10	98,87	34,04	50,64
20	99,42	9,98	18,14
30	99,35	2,99	5,81

Źródło: opracowanie własne autorów

Jak pokazuje tabela 12, w przeprowadzonych doświadczeniach, bez względu na wartość parametru odcięcia wskaźnik precyzji[75] utrzymywał się cały czas na wysokim poziomie. Spowodowane było to zapewne tym, że algorytm grupowania hierarchicznego uwzględniał częściowo te same cechy (afiliacje, współautorstwo, dziedzina), co metoda identyfikacji prostej. Najwyższa wartość wskaźnika kompletności została osiągnięta przy najniższym progu parametru odcięcia. Należy również wziąć pod uwagę fakt, że nie wszystkie publikacje przyporządkowane przez algorytm grupowania hierarchicznego mogły zostać zweryfikowane (jedynie te, które wcześniej uwzględnił algorytm identyfikacji prostej). Z tego względu, w razie ewentualnego użycia należałoby ustawić parametr odcięcia na wyższą wartość, wynoszącą 10 punktów, tak aby uzyskać precyzję możliwie najwyższą w stosunku do kompletności.

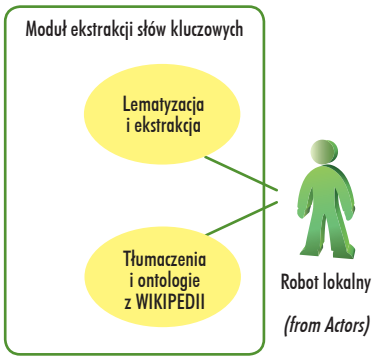
IV. Moduł ekstrakcji słów kluczowych

Zadanie modułu to wyodrębnianie słów kluczowych z zadanego tekstu (rysunek 33). W ten sposób stworzona lista słów kluczowych, wraz z opcjonalną listą słów przypisanych manualnie może być traktowana jako wektor deskryptorów dokumentu (spójna metoda reprezentacji dokumentu). Można zrealizować to zadanie dla

[75] Zagadnienia teoretyczne wyjaśniono w dodatku.

języka polskiego i angielskiego, z możliwością rozszerzenia na kolejne języki. Konfigurowalna jest też liczba otrzymywanych słów kluczowych oraz ich długość. Moduł ekstrakcji dla tekstów w języku polskim oparty jest na autorskim rozwiązaniu Polish Keyword Extractor (PKE[76]), z kolei wersja dla języka angielskiego wykorzystuje nowozelandzką otwartą bibliotekę Maui[77]. W obu przypadkach otrzymywane słowa kluczowe występują bezpośrednio w tekście dokumentu.

Rysunek 33. Role i czynności użytkowników w module ekstrakcji słów kluczowych



Źródło: opracowanie własne autorów

1. Lematyzacja i ekstrakcja

Maui to algorytm tematycznego indeksowania tekstów oparty na metodzie Keyphrases Extraction Algorithm (KEA[78, 79]). Może być używany bez zewnętrznych źródeł wiedzy lub z nimi (tezaurusy, ontologie); w projektowanym systemie wykorzystuje się Maui bez dodatkowych zewnętrznych słowników. Działanie algorytmu opiera się na dwóch etapach: selekcji kandydatów oraz ich ocenie za pomocą metod uczenia maszynowego. Kandydatami są *n*-gramy – co najwyżej o długości 3, które nie zaczynają się ani nie kończą *stop words*. Przed rozpoczęciem ekstrakcji, w celu zbudowania modelu musi nastąpić faza uczenia. Dla każdego *n*-gramu Maui oblicza następujące cechy:

- TF-IDF;
- pozycja pierwszego wystąpienia w tekście;
- odległość między pierwszym a ostatnim wystąpieniem w tekście;
- długość (liczba składowych słów);
- słowo-kluczowość (określa, jak często kandydat pojawił się jako słowo kluczowe w korpusie treningowym);
- stopień semantycznego powiązania kandydatów (opcjonalnie).

Dla tak zbudowanych wektorów cech model klasyfikatora bayesowskiego wyznacza prawdopodobieństwo, iż dany kandydat jest słowem kluczowym. Kandydaci z najwyższymi prawdopodobieństwami będą uznawani za ostateczne słowa kluczowe.

PKE jest autorskim algorytmem zainspirowanym metodami Rapid Automatic Keyword Extraction (RAKE[80]) i KEA. W przeciwieństwie do tych metod posiada polski lematyzator, tagger części mowy oraz dedykowane metody selekcji i oceny kandydatów. Ocena może być dokonywana w oparciu o metody statystyczne lub uczenia maszynowego (klasyfikator bayesowski). PKE dzieli tekst na zdania, następnie słowa są normalizowane (lematyzowane) i opisywane znacznikami części mowy. Selekcja słów kluczowych, oparta na wzorcach częstych słowo-kluczowych wyrażeń części mowy, pozwala zidentyfikować skończoną liczbę kandydujących

[76] Zagadnienia teoretyczne wyjaśniono w dodatku.
[77] Medelyan O., Frank E., Witten I.H., *Human-Competitive Tagging Using Automatic Keyphrase Extraction*, in: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, ACL, Stroudsburg 2009.
[78] Witten I.H., Paynter G.W., Frank E., Gutwin C., Nevill-Manning C.G., *KEA: Practical Automatic Keyphrase Extraction*, in: *Proceedings of the Fourth ACM Conference on Digital Libraries*, ACM, New York 1999.
[79] Zagadnienia teoretyczne wyjaśniono w dodatku.
[80] Zagadnienia teoretyczne wyjaśniono w dodatku.

słów kluczowych. Ostatecznie kandydaci są oceniani przez modele (klasyfikator lub miara statystyczna), a ci z najwyższymi ocenami zwracani jako finalne słowa kluczowe.

Wyodrębnienia słów kluczowych ze streszczeń artykułów naukowych można dokonać na dwa sposoby:

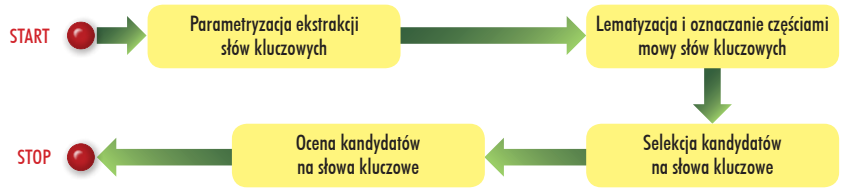
- analiza każdego abstraktu oddzielnie i następnie utworzenie pewnej wspólnej reprezentacji uzyskanych zbiorów dla pracownika naukowego lub dziedziny;
- zgrupowanie abstraktów w zbiór tekstów, a następnie analiza całego połączonego tekstu streszczeń publikacji danego pracownika naukowego lub dziedziny.

W systemie ostatecznie zastosowano pierwsze podejście, dzięki temu pozyskano słowa kluczowe dla każdej publikacji dysponującej abstraktem. Pozwala to na lepsze opisanie publikacji, zwłaszcza gdy autor jest interdyscyplinarny. Technicznie proces ekstrakcji wymaga wcześniejszego zbudowania ekstraktora. Następuje to na poziomie startu aplikacji; wtedy tworzone są instancje ekstraktorów. Odbywa się ładowanie gotowych modeli z bazy danych, albo przez proces uczenia. Ekstrakcja dla języków angielskiego i polskiego nieznacznie się różni. Kolejne etapy tych dwóch procesów przedstawiono na rysunkach 34 i 35.

Rysunek 34. Ekstrakcja słów kluczowych dla języka angielskiego

Źródło: opracowanie własne autorów

Rysunek 35. Ekstrakcja słów kluczowych dla języka polskiego



Źródło: opracowanie własne autorów

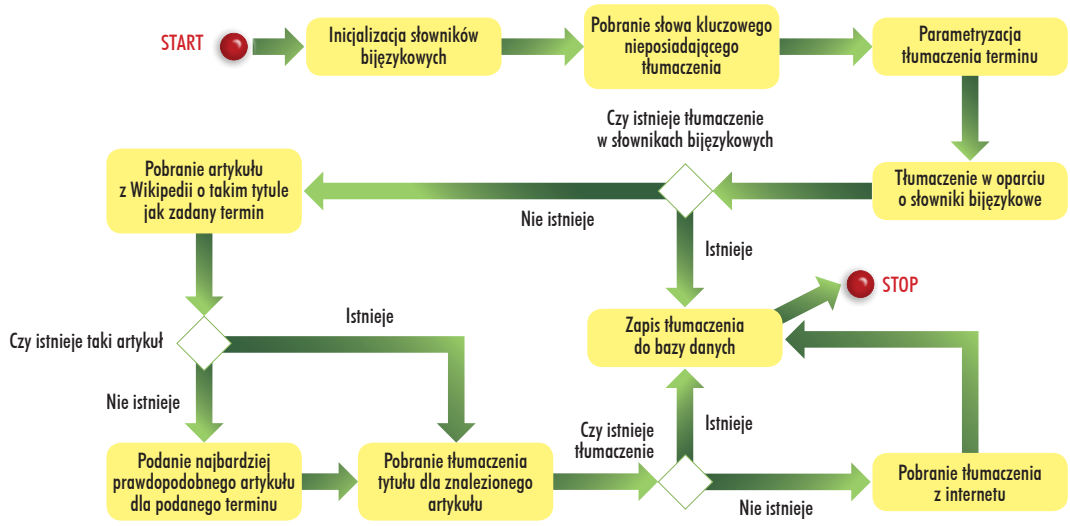
2. Tłumaczenia i „ontologiczna" wiedza z Wikipedii

System został dodatkowo wyposażony w mechanizm automatycznego tłumaczenia słów kluczowych z języka polskiego na angielski i odwrotnie. Tłumaczenia odbywają się przy wykorzystaniu publicznie dostępnych słowników ogólnego użytku, polskiej i angielskiej Wikipedii, oraz w ostateczności w oparciu o tłumaczenia dostępne w internecie. Dysponując definicjami wybranych słów kluczowych, Wikipedia służy także jako tezaurus, a umożliwiając budowanie grafu powiązań między słowami kluczowymi – jako uproszczona ontologia. Dla podanego słowa kluczowego istnieje możliwość pobrania z Wikipedii jego definicji, deskryptywnych etykiet, hierarchii kategorii czy terminów powiązanych z nim semantycznie. Pozwala to pozyskiwać wiedzę niedostępną bezpośrednio w publikacjach, a także budować relacje między słowami kluczowymi. Do urzeczywistnienia powyższych mechanizmów wykorzystano wiedzę zgromadzoną w polsko- i anglojęzycznej Wikipedii. Przebieg procesów tłumaczenia słów oraz znajdowania słów bliskoznacznych przedstawiono na rysunkach 36 i 37.

V. Moduł rankingowania

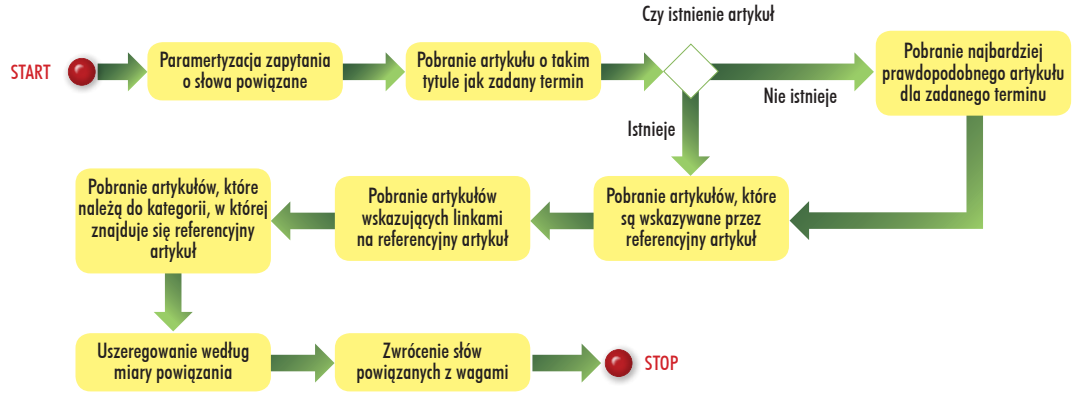
Moduł ma wygenerować propozycje recenzentów dla zadanego problemu. Składają się na to cztery procesy biznesowe: analiza dokumentu do recenzji, wprowadzanie słów kluczowych, tworzenie profilu naukowca oraz generowanie rankingu (rysunek 38). Poniżej prezentowana jest przykładowa implementacja procesów tego modułu. Ranking będzie przedmiotem szczegółowych badań i jego założenia teoretyczne będą się zmieniały, poniższą propozycję należy traktować zatem jako punkt wyjścia do dalszych badań.

Rysunek 36. Tłumaczenie słów



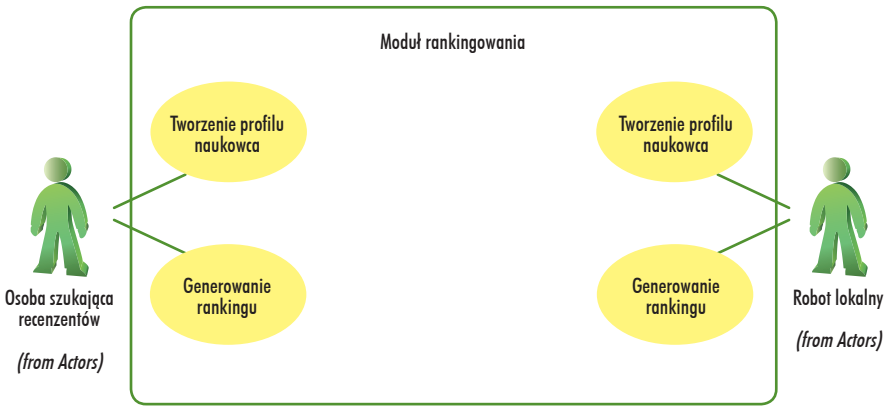
Źródło: opracowanie własne autorów

Rysunek 37. Słowa powiązane



Źródło: opracowanie własne autorów

Rysunek 38. Role i czynności użytkowników w module rankingowania



Źródło: opracowanie własne autorów

1. Analiza dokumentu i wprowadzanie słów kluczowych

Osoba szukająca recenzentów może zdefiniować problem wprowadzając tekst do recenzji lub podając zbiór słów kluczowych. Jeżeli podawany jest tekst, to wykonane zostanie automatyczne wyodrębnienie istotnych słów za pomocą algorytmów z modułu ekstrakcji słów kluczowych. Użytkownik może skorygować lub uzupełnić wyrazy wydobyte automatycznie. Istnieje także możliwość podania klasyfikacji nauki jako definicji problemu – wtedy klasyfikacje traktowane są jak słowa kluczowe. Niezależnie od sposobu wprowadzania danych, ostatecznie problem jest określony jako wektor słów:

$$p = [p_1, p_2, ..., p_n]^T$$

gdzie:
n – liczba słów definiujących problem.

2. Tworzenie profilu naukowca

Każdy potencjalny recenzent r posiada swój profil, który stanowi wektor cech, gdzie każda cecha zawiera słowo kluczowe i wagi określające stopnie przynależności słowa do osoby:

$$\boldsymbol{c_r} = [\{w_1^1, w_1^2, w_1^3, \ldots, s_1\}, \{w_2^1, w_2^2, w_2^3, \ldots, s_2\}, \ldots, \{w_m^1, w_m^2, w_m^3, \ldots, s_m\}]^T$$

gdzie:
m – liczba słów kluczowych potencjalnego recenzenta.

Wektor cech c_r potencjalnego recenzenta r jest generowany przez proces tworzenia profilu naukowca, który pobiera słowa z dostępnych źródeł i oblicza wagi słów. Słowa opisujące danego naukowca mogą pochodzić z wielu źródeł i dla każdego z nich istnieje oddzielna metoda liczenia wag. Obecnie możliwe są następujące źródła pochodzenia słów:

- słowa mogą być podane przez samego naukowca jako jego specjalizacje i klasyfikacje; dane znajdują się w bazach BWNP, OSF i zazwyczaj są podawane podczas zgłaszania doktoratu lub habilitacji oraz zgłaszania się do bazy recenzentów lub recenzowania – wagę słowa s_i oznaczymy jako w_i^1;
- słowa mogą pochodzić ze słów kluczowych publikacji, które są podawane przez autorów publikacji – wagę słowa s_i oznaczymy jako w_i^2;

- słowa mogą pochodzić z ekstrakcji słów kluczowych ze streszczeń i tytułów publikacji – wagę słowa s_i oznaczymy jako w_i^3.

Słowa podane przez naukowca uznaje się za najbardziej wiarygodne, zatem:

$$w_i^1 = 1$$

Natomiast w przypadku słów pochodzących z publikacji wagi są uzależnione wykładniczo od roku publikacji:

$$w_p = e^{-(y_n - y_p)/c}$$

gdzie:
y_p – rok publikacji, z której pochodzi dane słowo;
y_n – obecny rok;
$C = 5$ – stała[81]

oraz sigmoidalnie od liczby wystąpień słowa:

$$w^{2,3} = \frac{1}{1 + e^{-b \cdot \Sigma_{i=1}^{p}(w_p \cdot c_y \cdot w_k)}}$$

gdzie:
w_p – waga publikacji;
c_y – liczba publikacji z danego roku zawierających słowo kluczowe;
$b = 0{,}5$ – stała[82].
w_k – waga słowa, która dla słów będącymi manualnymi słowami kluczowymi wynosi 1, natomiast dla słów pochodzących z ekstrakcji słów kluczowych jest równa prawdopodobieństwu obliczanemu podczas ekstrakcji słów z abstraktu i tytułu publikacji (korzystając z twierdzenia Bayesa oblicza się, czy dane słowo jest odpowiednim kandydatem na słowo kluczowe).

3. Generowanie rankingu

Ranking recenzentów będzie zawierać listę osób posortowaną malejąco według miary podobieństwa pomiędzy wektorem słów p definiującym problem a wektorem cech c_r recenzenta r. Zostanie przyjęta cosinusowa miara podobieństwa:

$$d_r = \frac{\boldsymbol{p} \cdot \boldsymbol{c_r}}{|\boldsymbol{p}| \cdot |\boldsymbol{c_r}|}$$

przy czym:

$$\boldsymbol{p} \cdot \boldsymbol{c_r} = \sum_{i=1}^{n} Ep_i \cdot \max\{w_i^1, w_i^2, w_i^3, \ldots\} \cdot Es_i$$

$$|\boldsymbol{p}| = \sqrt{\Sigma_{i=1}^{n}(Ep_i)^2)}$$

$$|\boldsymbol{c_r}| = \sqrt{\Sigma_{i=1}^{n}(\max\{w_i^1, w_i^2, w_i^2, \ldots\} \cdot Es_i)^2)}$$

[81] Ostateczna wartość stałej zostanie dobrana doświadczalnie podczas walidacji rankingu.
[82] Ostateczna wartość stałej zostanie dobrana doświadczalnie podczas walidacji rankingu.

gdzie:
$Ep_i = 1$ dla każdego;
$Es_i = 1$ jeżeli $p_i = s_i$ i $Es_i = 0$ w przeciwnym wypadku.

W finalnej wersji systemu ranking będzie uwzględniał indeks Hirscha[83], w celu uwzględnienia elementu częstości cytowań prac poszczególnych autorów.

VI. Podsumowanie

W wyniku działania algorytmów zawartych w zaprojektowanych modułach, powstanie nie tylko baza danych o potencjalnych recenzentach, ale także system zawierający relacje pomiędzy tymi danymi. Na rysunku 39 pokazano związki między potencjalnym recenzentem, słowami, publikacjami, afiliacjami, klasyfikacjami, źródłami. Potencjalny recenzent należy do odpowiednich dziedzin i dyscyplin, które są reprezentowane przez model siedmiu powiązanych ze sobą taksonomii nauki. Klasyfikacje nauki są rozszerzane przez słowa kluczowe. Oczywiście pracownik naukowy posiada także wektor ważonych słów kluczowych tworzący jego profil, jest autorem publikacji oraz jest związany z różnymi instytucjami, co nazywamy afiliacjami. Obiekty publikacji posiadają źródła oraz – podobnie jak osoby – należą do odpowiednich klasyfikacji nauki i posiadają afiliacje.

Rysunek 39. Związki pomiędzy danymi

Źródło: opracowanie własne autorów

Z całą pewnością nadal rozwijane będą algorytmy systemu i testowane różne nowe, obiecujące metody. W najbliższym czasie planowane jest uruchomienie crawlera analizującego strony internetowe pracowników naukowych. Prawdopodobnie mechanizmy ekstrakcji słów kluczowych będą wymagały ciągłego doskonalenia, a także budowy bardziej wyrafinowanych modeli, być może oddzielnych dla każdej dziedziny nauki i języka. Ponadto, w algorytmie grupowania hierarchicznego arbitralnie przyjęto wartości parametrów podobieństwa. Planowane jest zastosowanie technik optymalizacyjnych do wyznaczania tych parametrów i ponowne testy algorytmu. Ostatecznie, metodę generowania rankingu potencjalnych recenzentów należy traktować jako pierwszą propozycję, która będzie weryfikowana i modyfikowana już podczas używania systemu przez końcowych użytkowników. Autorzy nie zakładają, że przedstawiony systemem jest dziełem kompletnym i zamkniętym. Jest to raczej przyczynek do dalszej pracy, której efekty być może pozwolą lepiej dobierać recenzentów.

[83] Zagadnienia teoretyczne wyjaśniono w dodatku.

VII. Bibliografia

Bishop C.M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, New York 2006.

Medelyan O., Frank E., Witten I.H., *Human-Competitive Tagging Using Automatic Keyphrase Extraction*, in: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 3, ACL, Stroudsburg 2009.

Witten I.H., Paynter G.W., Frank E., Gutwin C., Nevill-Manning C.G., *KEA: Practical Automatic Keyphrase Extraction*, in: *Proceedings of the Fourth ACM Conference on Digital Libraries*, ACM, New York 1999.

Dodatek

WYBRANE PODSTAWY TEORETYCZNE

(Sławomir Dadas, Tomasz Stanisławek, Marek Kozłowski, Jarosław Protasiewicz, Małgorzata Gałężewska)

Autorzy dziękują Kamilowi Główczyńskiemu za współudział w wykonaniu analizy metod ekstrakcji słów kluczowych.

I. Roboty internetowe

Roboty internetowe są programami służącymi do gromadzenia danych na temat struktury i zawartości dokumentów znajdujących się w sieci internet. Innymi stosowanymi zamiennie pojęciami o podobnym znaczeniu są: *bot*, *web spider* (pająk sieciowy), *web crawler* (pełzacz sieciowy), *web wanderer* (podróżnik sieciowy), *ant* (mrówka). Ich działanie opiera się na przeszukiwaniu stron na podstawie zdefiniowanych list, które mogą być rozszerzane o nowe pozycje w trakcie działań.

W potocznym rozumieniu robot internetowy kojarzony jest często ze specyficznym i najpopularniejszym rodzajem robotów indeksujących wyszukiwarek internetowych. Ich sposób działania polega na przeszukiwaniu sieci w uporządkowany sposób, w celu budowy indeksów wyszukiwarek. Na każdej odwiedzonej stronie robot poszukuje odnośników do innych dokumentów znajdujących się w internecie, a następnie podąża za tymi odnośnikami. W ten sposób powiększa się ilość stron zaindeksowanych przez wyszukiwarki. Dodatkowo, poza rozszerzaniem indeksu o nowe wpisy roboty indeksujące odwiedzają już wcześniej zaindeksowane strony w celu aktualizacji danych.

Istnieją jednak również roboty przeznaczone do zupełnie innych zadań. Popularne jest wykorzystywanie aplikacji służących do tworzenia pełnych kopii (*mirrors*) oraz archiwów wersji stron internetowych. Wiele serwisów gromadzących duże ilości danych korzysta też z automatycznych procesów weryfikujących poprawność i spójność tych danych (np. walidacja dostępności linków lub sprawdzanie zgodności dokumentów z określonym formatem). Innym typem robotów są aplikacje specjalnie zaprojektowane do gromadzenia specyficznych rodzajów danych (np. spamboty wyszukujące w internecie adresy e-mail i dane osobowe)[84].

Zazwyczaj roboty indeksujące wykorzystywane przez wyszukiwarki internetowe nie filtrują dokumentów, które pobierają. W pewnych specyficznych zastosowaniach korzysta się jednak z robotów gromadzących dane dotyczące konkretnego, zdefiniowanego tematu. Roboty takie są określone nazwą *focused crawlers* lub *topical crawlers* [85, 86, 87]. Lista stron, których zawartość będzie pobierana przez robota, może być zdefiniowana manualnie przez użytkownika lub odkrywana w trakcie działania robota dzięki odpowiednim algorytmom *text mining*, które pozwalają na określenie, czy aktualnie przeszukiwany dokument jest zgodny ze zdefiniowanym tematem. Podobnie jak u robotów indeksujących, roboty tematyczne potrzebują pewnego początkowego zbioru adresów stron relewantnych. Często spotykanym przykładem tego typu aplikacji są roboty wykorzystywane przez porównywarki cenowe. Serwisy takie udostępniają swoim klientom interfejs pozwalający na zesta-

[84] Liu B., *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*, Springer, New York 2010.

[85] Hsu C.C., Wu F., *Topic-specific crawling on the web with the measurements of the relevancy context graph*, „Information Systems", 31(4), 232–246, 2006.

[86] Almpanidis G., Kotropoulos C., Pitas I., *Combining text and link analysis for focused crawling – An application for vertical search engines*, „Information Systems", 32(6), 886–908, 2007.

[87] Diligenti M., Coetzee F., Lawrence S., Giles C.L., Gori M., *Focused Crawling Using Context Graphs*, in: El Abbadi A. et al., ed., *Proceedings of the 26th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., San Francisco 2000.

wienie cen tego samego produktu w różnych sklepach internetowych, tak aby mogli wybrać najkorzystniejszą ofertę. Aby zaprezentować dane w wygodnej dla klienta formie, robot musi zebrać dane o produktach na stronach poszczególnych sklepów, odczytując z treści dokumentów jak najwięcej informacji o produkcie – poza ceną i nazwą również kategorię, model, nazwę producenta *etc.* Im więcej danych może być prawidłowo odczytanych przez robota, tym większe możliwości daje użytkownikom interfejs porównywarki.

1. Architektura robotów internetowych

Nie istnieje uniwersalny algorytm robota. Poszczególne implementacje mogą różnić się od siebie w zależności od zadań i sposobu gromadzenia danych. Roboty zbierające dane z dowolnych dokumentów znajdujących się w internecie działają w inny sposób niż roboty, których źródła danych zostały ściśle zdefiniowane. Część robotów indeksuje pełną treść dokumentów, inne skupiają się tylko na wyodrębnianiu konkretnych danych, ignorując pozostałą część. W podobny sposób implementowane są roboty tworzące listę stron do przeszukiwania w trakcie działania, do których należą przede wszystkim boty indeksujące. Rozpoczynają one swoje działanie, mając do dyspozycji pewną początkową liczbę adresów URL, a następnie rozbudowują ją o kolejne adresy. Na każdej nowo odwiedzanej stronie wyszukują linki do innych dokumentów. Jeżeli linki prowadzą do dokumentów jeszcze nieodwiedzonych, są one dodawane do listy. W ten sposób zbiór znanych przez robota stron jest rozszerzany. Proces ten trwa do momentu, gdy osiągnięty zostanie warunek zatrzymania algorytmu. Warunki takie mogą być zdefiniowane w różny sposób, w zależności od celu działania robota. Aplikacja może zostać zatrzymana po ściągnięciu określonej liczby stron, gdy:

- wyczerpią się przydzielone robotowi zasoby;
- pobrane będą wszystkie strony oddalone o nie więcej niż *k* linków od listy stron początkowych;
- minie określony przez programistę czas od uruchomienia procesu
- lub spełniony zostanie inny warunek odpowiedni dla danego typu robota[88].

Schemat działania robota internetowego przedstawia rysunek 40.

Określona lista początkowych adresów, z której korzysta robot, nosi nazwę granicy (*frontier*)[89]. Dla każdego adresu URL znajdującego się na liście robot pobiera dokument znajdujący się pod tym adresem, wyodrębnia z niego wszystkie linki, a następnie dla każdego linku sprawdza, czy został on już wcześniej dodany do granicy. Jeżeli warunek nie jest spełniony, granica rozszerza się o nowy adres. Początkowa lista stron może być utworzona manualnie lub na podstawie już istniejącej bazy stron (np. z wyszukiwarki internetowej). Dokumenty pobierane przez robota są na ogół przechowywane w bazie danych. Zapisywany może być pełen tekst dokumentu lub dane z niego wyodrębnione. Niektóre roboty zapisują też dodatkowe metainformacje o dokumencie: miary istotności, klasyfikację dokumentu, czas ostatniego pobrania, adres strony źródłowej, na której znaleziony został URL do dokumentu.

Sieć stron internetowych może być traktowana jak graf, w którym wierzchołki są reprezentowane przez dokumenty, natomiast krawędzie przez linki między dokumentami. W takim kontekście algorytm robota można rozpatrywać jak klasyczny algorytm przeszukiwania grafu, który rozpoczyna swoją pracę od określonego zestawu wierzchołków i rozbudowuje go o kolejne wierzchołki, podążając za krawędziami. W związku z tym, można tutaj zastosować popularne strategie przeszukiwania grafu. Ze względu na ilość dokumentów w internecie oraz gęstość połączeń między nimi, do najbardziej efektywnych i najczęściej wykorzystywanych podczas implementacji strategii należą przeszukiwania wszerz i priorytetowe.

2. Metody wyszukiwania

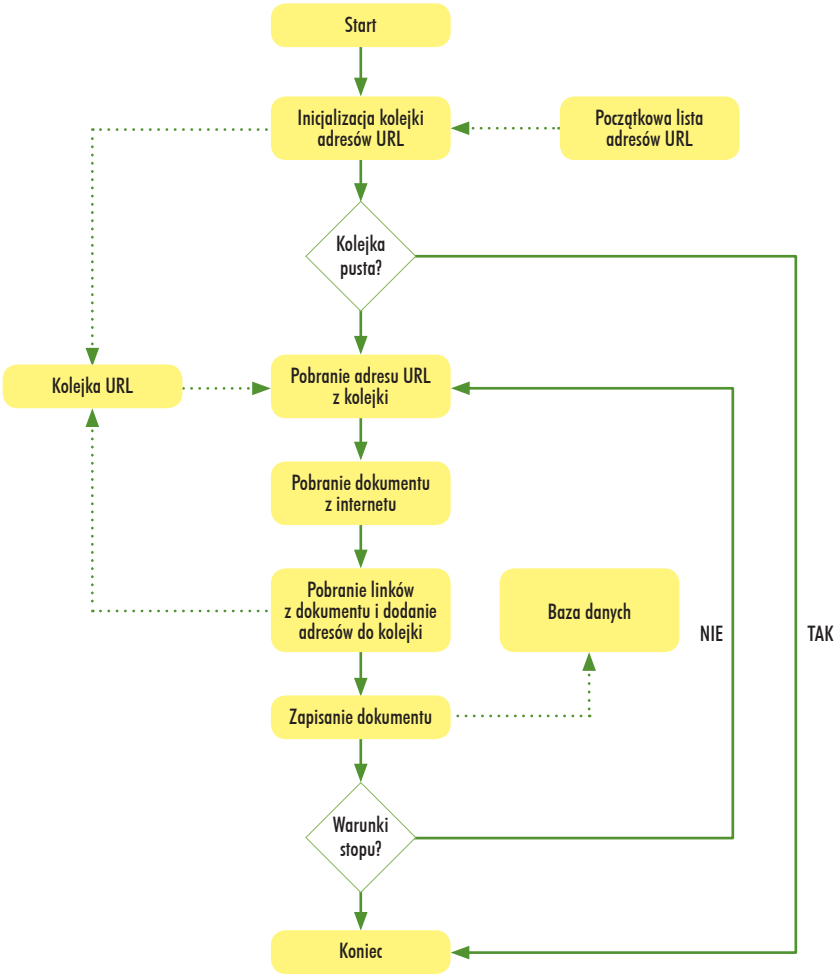
2.1. Przeszukiwanie wszerz

W tej metodzie robot rozpoczyna poszukiwanie od adresów znajdujących się początkowo w granicy, a dopiero po ich odczytaniu pobiera kolejne dokumenty. W praktyce strategię tę najłatwiej zaimplementować na

[88] Liu B., op.cit.
[89] Ibidem.

kolejce FIFO. W każdym nowo odwiedzonym dokumencie wszystkie odnalezione adresy URL są dodawane do końca kolejki. Oznacza to, że odwiedzone zostaną dopiero wtedy, kiedy robot odczyta wszystkie wcześniejsze adresy[90].

Rysunek 40. Schemat działania robota internetowego



Źródło: opracowanie własne autorów na podstawie: Liu B., *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*, Springer, New York 2010.

2.2. Przeszukiwanie priorytetowe

Strategia ta pozwala na nadanie każdemu odnalezionemu przez robota adresowi URL priorytetu. Istotność może być definiowana na postawie istotności strony źródłowej, na której znajdował się link lub na podstawie analizy kontekstu, w jakim się pojawił. W metodzie tej zamiast kolejki FIFO stosowana jest kolejka priorytetowa. Adresy o wyższym priorytecie trafiają na początek kolejki i są odczytywane szybciej niż adresy o niższych priorytetach[91].

3. Problemy

Z powodu dynamicznego rozwoju internetu i rozpowszechnienia technologii umożliwiających tworzenie interaktywnych aplikacji internetowych (AJAX, Flash, Silverlight *etc.*) pojawia się coraz więcej problemów

[90] Ibidem.
[91] Ibidem.

utrudniających implementację robotów. Współczesne roboty muszą uwzględniać wciąż więcej przypadków, na które można natknąć się podczas przeszukiwania sieci. Wdrożenie efektywnego robota wymaga od programisty przewidywania sytuacji problematycznych i opracowania sposobów ich uniknięcia lub zminimalizowania negatywnego efektu. Poniżej zaprezentowane zostaną najczęściej spotykane trudności związane z projektowaniem automatycznych procesów przeszukujących internet.

3.1. Pułapki *(spider traps)*

Coraz więcej serwisów internetowych zawiera treści generowane dynamicznie. Czasem prowadzi to do sytuacji, kiedy możliwe jest wygenerowanie teoretycznie nieskończonej liczby stron, każdej z innym adresem URL. Strony internetowe, które w zamierzony lub niezamierzony sposób powodują zapętlenie się robota poprzez generowanie unikatowych linków do dynamicznie tworzonych dokumentów, noszą nazwę *spider traps*. W większości przypadków działanie takie nie jest zamierzone i wynika jedynie z udostępnionej funkcjonalności serwisu oraz treści w nim zawartych. Przykłady takiego niezamierzonego działania strony internetowej to[92]:

1. **Dynamicznie generowane kalendarze,** zawierające odnośniki do kolejnych miesięcy bądź lat. Podążając za tymi linkami, robot może wygenerować nieskończenie wiele stron. Dobrym rozwiązaniem, które pozwoliłoby serwisowi na uniknięcie tego problemu, jest wprowadzenie ograniczenia na możliwą do wyświetlenia datę.
2. **Strony generujące losowe treści** (np. tekst *lorem ipsum*) z niepowtarzalanymi adresami URL.
3. **Serwisy internetowe zapisujące w adresie sekwencję ostatnio wykonanych przez użytkownika czynności.** Może być to sklep internetowy, który dla kolejnych odwiedzanych przez określoną osobę stron z informacjami produktach umieszcza w adresie URL listę numerów ID poprzednio przeglądanych artykułów. Dzięki temu sklep może analizować, które towary są najczęściej oglądane wspólnie. Negatywnym skutkiem takiej funkcjonalności jest jednak generowanie wielu adresów linkujących do tej samej strony. Roboty internetowe mogą traktować je jako odwołania do różnych dokumentów, co potencjalnie grozi zapętleniem się procesu.
4. **Strony internetowe umieszczające w swoich adresach parametry,** których wartości są obliczane na postawie unikatowych numerów powiązanych z sesją lub „ciasteczkami" (*cookies*) użytkownika.
5. **Wyszukiwarki,** będące przede wszystkim częścią większych serwisów, zawierające odnośniki do ostatnio wykonywanych zapytań.

Opisane pułapki działają nie tylko na szkodę samego robota, który odczytuje wiele niepotrzebnych lub zduplikowanych dokumentów, ale również serwerów hostujących daną stronę. Długotrwałe przeszukiwanie serwisu przez robota obciąża serwer dużą ilością niepotrzebnych żądań HTTP, co spowalnia jego działanie. Czasami zdarza się też, że wielokrotnie wykonywana przez robota akcja powoduje tworzenie nowych rekordów w bazie danych po stronie serwera, a to skutkuje wypełnieniem bazy znaczną ilością niepotrzebnych informacji.

Ponieważ istnieje wiele sposobów na utworzenie wyżej opisanych pułapek, nie ma szansy opracowania uniwersalnej metody ich rozpoznania. Nawet gdy obsłużone zostaną najczęściej występujące typy zasadzek, to wciąż robot narażony będzie na pułapki nietypowe, których programista nie mógł przewidzieć. Aby uniknąć utknięcia w nieskończonych pętlach, przy projektowaniu robota należy uwzględnić pewne limity; ich przekroczenie powinno przerwać znajdywanie danych w aktualnie przeszukiwanym serwisie. Jedno z takich ograniczeń to długość adresu URL – jeżeli przekroczy ona zdefiniowaną ilość znaków, dokument znajdujący się pod takim adresem nie powinien zostać ściągnięty. Inny sposób polega na ograniczeniu liczby dokumentów sekwencyjnie pobranych w ramach tej samej domeny. Na przykład wyszukiwanie może zostać przerwane, jeżeli liczba kolejnych linków odwołujących się do tej samej domeny przekroczy sto. Od tego momentu robot powinien ignorować wszystkie adresy URL prowadzące do tej domeny, a podążać tylko za linkami wskazującymi na strony zewnętrzne. Czasami zdarza się, że serwis sam informuje o możliwości wpadnięcia w pułapkę i umieszcza w pliku robots.txt odpowiedni wpis zabraniający robotom pobierania stron dynamicznie generowanych. Każdy robot przestrzegający tych zasad może uniknąć wpadnięcia w *spider trap*[93].

[92] Ibidem.
[93] Ibidem.

Jeżeli robot internetowy stosuje strategię przeszukiwania wszerz, wpływ pułapek na jego ogólną wydajność nie jest duży. Dzięki tej strategii robot przeszukuje zazwyczaj wiele stron w tym samym czasie, więc pomimo utknięcia w pułapce na jednej nich jest w stanie skutecznie pobierać dane z pozostałych.

3.2. Normalizacja adresów URL

W trakcie ekstrakcji linków z dokumentów znalezionych w internecie robot może spotkać się z różnymi formatami zapisu adresu URL odwołującego się do tego samego zasobu. Jeżeli nie będzie przeprowadzał normalizacji odczytywanych adresów, istnieje ryzyko, że różnie zapisane adresy będą traktowane jak odwołania do dwóch różnych dokumentów, podczas gdy oba wskazują na ten sam zasób. Jeżeli zasoby znajdują się w tej samej domenie co dokument, który się do nich odwołuje, adres może być zapisany za pomocą ścieżki absolutnej bądź relatywnej. Jest to podstawowa kwestia, którą musi uwzględniać robot podczas odczytywania adresów. Do innych aspektów związanych z zapisem URL należą[94]:

- wielkość liter w nazwie domeny – adres tej samej strony może być zapisany za pomocą wielkich i małych liter;
- adres może opcjonalnie zawierać numer portu – domyślny numer portu (80) nie jest wymagany, w związku z tym podanie go zmienia jedynie zapis adresu, nadal wskazuje on jednak na ten sam zasób;
- domyślna nazwa dokumentu – często odwoływanie się do zasobu o nazwie index.html zwraca dokładnie ten sam dokument, co odwołanie do samego katalogu, bez podania nazwy dokumentu;
- nieprawidłowe znaki w adresie – jeżeli adres URL zawiera znaki, które są niedozwolone, powinny one zostać zastąpione odpowiadającymi im sekwencjami znakowymi (*escape sequences*);
- sekwencje znakowe mogą zastępować również prawidłowe i dopuszczalne znaki w adresie – w razie takiej sytuacji robot powinien przywrócić oryginalny znak odpowiadający sekwencji;
- w adresie może znajdować się odwołanie do dokumentu wraz ze wskazaniem fragmentu tego dokumentu – normalizacja powinna usunąć wszystkie takie wskazania.

Prawidłowo zaimplementowana normalizacja zwiększa efektywność działania robota poprzez unikanie pobierania duplikatów dokumentów. Poza prostą normalizacją adresów, niektóre roboty wykorzystują również bardziej zaawansowane techniki oparte na algorytmach heurystycznych. Techniki te pozwalają eliminować z adresów nadmiarowe parametry żądań HTTP, na przykład powiązane z sesją lub „ciasteczkami" użytkownika[95].

3.3. Odczyt struktury dokumentu

Poza pobieraniem dokumentów z sieci, do zadań robotów należy też zazwyczaj przetworzenie tych dokumentów w taki sposób, aby wyodrębnić dane istotne dla systemu, w kontekście którego robot działa. Wyodrębniane dane mogą być łatwe do zlokalizowania (np. adresy e-mail lub metadane z nagłówka dokumentu), ale w większości przypadków do ich wyodrębnienia potrzebna jest znajomość struktury strony HTML. W celu odczytania takiej struktury, roboty muszą skorzystać z mechanizmu analizy dokumentu.

Zaimplementowanie takiego mechanizmu od podstaw jest bardzo złożonym zadaniem. Rozwój internetu i pojawianie się nowych standardów oraz technologii spowodowały, że współczesne dokumenty składają się nie tylko z tagów i atrybutów, ale również ze skryptów JavaScript, CSS, a także różnych typów zagnieżdżonych obiektów. Powszechny dostęp do internetu oraz coraz większe możliwości publikacji treści przyciągają osoby niezajmujące się informatyką profesjonalnie. W efekcie wiele dostępnych stron nie spełnia standardów prawidłowego dokumentu HTML. Do najczęściej popełnianych błędów należy: pomijanie wymaganych dla dokumentu tagów, nieprawidłowe ich zagnieżdżanie, umieszczanie tagów otwierających bez zamykających, pomyłki w nazwach i wartościach atrybutów, używanie znaków specjalnych bez sekwencji ucieczki *etc.*

Przeglądarki internetowe korzystają z zaawansowanych algorytmów pozwalających na poprawne wyświetlanie skonstruowanych dokumentów, jednak konieczność obsługi tego typu błędów w znacznym stopniu utrudnia możliwość zaimplementowania dedykowanego analizatora dokumentu (*parsera*) dla robota inter-

[94] Pant G., Srinivasan P., Menczer F., *Crawling the Web*, in: Levene M., Poulovassilis A., eds., *Web Dynamics: Adapting to Change in Content, Size, Topology and Use*, Springer, Berlin – Heidelberg 2004.
[95] Liu B., op.cit.

netowego. Zazwyczaj w robotach wykorzystywane są rozwiązania udostępniane przez biblioteki specjalnie w tym celu zaprojektowane (np. Jsoup dla języka Java lub BeautifulSoup dla języka Python) oraz przeglądarki internetowe o otwartych źródłach (np. silniki Gecko i Webkit). W większości przypadków analiza stron HTML polega na utworzeniu reprezentacji dokumentu w postaci drzewa DOM (*Document Object Model*). Struktura taka jest łatwa do przeszukiwania i ułatwia robotom lokalizację oraz wyodrębnianie treści[96].

3.4. Sieć ukryta

Tylko niewielka część dokumentów znajdujących się w internecie jest zaindeksowana przez roboty wyszukiwarek internetowych. Pozostała część z różnych powodów nie jest osiągalna przez klasyczne roboty indeksujące – wszystkie strony niedostępne dla takich robotów określa się pojęciem **sieci ukrytej** (*invisible web*) lub **sieci głębokiej** (*deep web*). Szacuje się, że rozmiar sieci ukrytej jest od kilkuset do kilku tysięcy razy większy od zaindeksowanej sieci [97, 98]. Większość znajdujących się w niej zasobów może zostać pobrana przez roboty specjalnego przeznaczenia, zaimplementowane w celu przeszukiwania konkretnych fragmentów. Szczegółową analizę sieci ukrytej przeprowadził Shestakov[99]. Strony znajdujące się w głębokiej sieci są niedostępne z różnych powodów. Główne przeszkody, które utrudniają robotom dostęp do wszystkich lub części treści na tych stronach to:

- udostępnianie treści w formatach, które nie mogą zostać odczytane przez roboty indeksujące;
- dostępność treści jedynie poprzez interfejs zapytań opartych na formularzach;
- strony wyświetlające dane w czasie rzeczywistym;
- serwisy udostępniające treści na podstawie subskrypcji lub opłaty za dostęp;
- serwisy wymagające rejestracji i logowania;
- strony zabraniające robotom dostępu do treści;
- zasoby, do których nie istnieją żadne odwołania w zindeksowanym internecie.

3.5. Dynamiczne strony internetowe

Popularyzacja języka JavaScript oraz technologii AJAX spowodowały, że coraz więcej stron internetowych korzysta z dynamicznych interfejsów z treściami pobieranymi asynchronicznie. Jest to duże wyzwanie dla typowych robotów, które traktują pobrane dokumenty jako treści statyczne i w taki sposób je analizują. Tymczasem strony korzystające z JavaScript mogą zmieniać swoją strukturę w odpowiedzi na akcje użytkownika oraz doczytywać pewne treści z użyciem zapytań AJAX już po załadowaniu się strony. Ignorowanie tego typu funkcjonalności powoduje, że robot może nie mieć dostępu do części informacji widocznych dla odwiedzających stronę.

Nowoczesne roboty na ogół zawierają już pewne mechanizmy umożliwiające ekstrakcję danych z dokumentów korzystających z asynchronicznych wywołań JavaScript. Stosuje się tutaj dwa podejścia. Pierwsze polega na zintegrowaniu mechanizmu robota z silnikami przeglądarek internetowych o otwartych źródłach, które zawierają interpretery JavaScript pozwalające na wykonywanie dowolnych funkcji i symulowanie rzeczywistego działania strony w przeglądarce. Podejście drugie opiera się na analizie treści stron oraz dołączonych skryptów – automatycznie wypełniane są żądania, które powinny zostać wykonane przez JavaScript, pobierając dynamicznie doczytywane dane[100].

3.6. Zasady dostępu do serwerów *(politeness policy)*

Automatyczne zapytania wysyłane przez robota internetowego są znacznym obciążeniem dla serwerów WWW, dlatego ich wpływ na prawidłowe i wydajne funkcjonowanie serwerów jest o wiele większy niż oddziaływanie pojedynczego użytkownika wysyłającego zapytania manualnie. Podczas implementacji robota należy pilnować, aby nie wysyłał on wielu zapytań do tego samego serwera jednocześnie. Stosowanie strategii

[96] Ibidem.
[97] Devine J., Egger-Sider F., *Beyond Google: The invisible web in the academic library*. „The Journal of Academic Librarianship", 30(4), 265–269, 2004.
[98] Shestakov D., *Search Interfaces on the Web: Querying and Characterizing*, University of Turku, 2008.
[99] Ibidem.
[100] Duda C., Frey G., Kossmann D., Zhou Ch., *AJAX Search: Crawling, indexing and searching web 2.0 applications*, „Proceeding VLDB Endowment", 1, 1440–1443, 2008.

przeszukiwania wszerz zazwyczaj pomaga rozkładać obciążenie na wiele serwerów, poszczególne serwisy mogą jednak wprowadzać dodatkowe ograniczenia, do których roboty powinny się stosować. Standard *Robot Exclusion Protocol*[101] pozwala autorom stron na umieszczanie w pliku robots.txt parametru *crawl-delay*, oznaczającego minimalny czas w sekundach pomiędzy kolejnymi żądaniami wysyłanymi przez robota.

4. Roboty tematyczne *(topical crawlers, focused crawlers)*

Klasyczne roboty indeksujące pobierają wszystkie dokumenty możliwe do zaindeksowania, bez analizowania ich treści. Dla pewnych specyficznych zastosowań wykorzystuje się roboty tematyczne, których zadanie to pobieranie dokumentów dotyczących konkretnej kwestii lub dziedziny wiedzy. Angielska nazwa *topical crawlers* jest pojęciem ogólniejszym i określa roboty zbierające dane powiązane z wybranym tematem. Termin *focused crawlers* wiąże się z robotami, które gromadzą dane tematyczne dysponując pewną liczbą dokumentów relewantnych (na ich podstawie wyszukiwane są kolejne informacje zgodne z dziedziną[102]). Podczas przeszukiwania sieci, dla każdego pobranego dokumentu robot określa miarę jego dopasowania do tematu. Jeżeli wartość obliczonej miary nie przekroczy pewnego progu, treść strony nie jest zapisywana.

5. Ekstrakcja danych a roboty internetowe

Do tej pory opisywane były przede wszystkim roboty indeksujące, zapisujące pełną zawartość pobieranych przez siebie dokumentów. Zazwyczaj jednak, poza treścią w postaci oryginalnego dokumentu oraz jego reprezentacji tekstowej takie roboty zapisują też inne dane przydatne podczas indeksowania: słowa kluczowe, tytuł strony, tytuły nagłówków występujących w treści dokumentu. Wymaga to od robota zastosowania prostych metod analizy treści stron internetowych. Odrębnym rodzajem robotów są aplikacje zaprojektowane specjalnie do wyodrębniania konkretnych danych z pobieranych dokumentów.

Większość kwestii poruszonych w tym rozdziale ma zastosowanie również dla robotów wyodrębniających dane, niemniej pewne problemy związane z ich implementacją wymagają uzupełnienia. W większości przypadków zadanie ekstrakcji wypełniane przez te roboty polega na wyodrębnieniu z dokumentów zawierających wymieszane dane encji (*entity*) o określonej strukturze. Dokumenty pobierane przez robota zawierają informacje zarówno istotne, jak i nieistotne z punktu widzenia ekstrakcji. Celem jest znalezienie i wydobycie tych fragmentów dokumentu, które dotyczą wyodrębnianej encji danych.

Popularną i stosunkowo prostą egzemplifikacją takich aplikacji są spamboty, wyszukujące w internecie adresy oraz inne dane kontaktowe osób. Spamboty zapisują tylko interesujące informacje, ignorując pozostałą część dokumentu. Inny przykład to serwis internetowy agregujący informacje pochodzące z innych serwisów. Udostępnia on użytkownikom jednolity interfejs, pozwalający na wysyłanie zapytań. Zapytania trafiają do serwisów zewnętrznych, ich wyniki są odczytywane przez dedykowanego robota i trafiają z powrotem do serwisu głównego, który zwraca je użytkownikowi. Tak jest chociażby ze stroną ułatwiającą śledzenie paczek wysyłanych przez kilka różnych firm kurierskich. Klienci wysyłający dużą liczbę paczek chcieliby mieć możliwość sprawdzania na bieżąco, co dzieje się z ich przesyłkami, bez konieczności logowania się na stronę każdej firmy z osobna. Serwis agregujący pozwala wyszukiwać paczki po ich numerach i nazwie dostawcy, a następnie zwraca informacje o stanie przesyłki. Każdorazowe zapytanie użytkownika powoduje odczytanie i wyodrębnienie tych danych z oficjalnej strony odpowiedniej firmy kurierskiej. Wyodrębnione informacje o przesyłce prezentowane są w jednolitej formie, bez względu na źródło, z którego pochodzą.

Zazwyczaj do celów ekstrakcji danych wykorzystuje się dedykowane rozwiązania, mające osiągać cele założone przez programistę. W prostej ekstrakcji danych, opartej na łatwych do zdefiniowania regułach, można zastosować rozwiązania ogólnego przeznaczenia. Jednym z takich rozwiązań jest robot Web-Harvest, aplikacja kliencka pozwalająca na definiowanie zadań w języku opartym na XML, udostępniona na licencji *open source*. Za pomocą tagów i atrybutów użytkownik aplikacji definiuje polecenia dla robota – wskazuje źródła danych, określa położenie interesujących danych wewnątrz dokumentów (z wykorzystaniem składni XPath)

[101] The Robot Exclusion Protocol, http://www.robotstxt.org/robotstxt.html, dostęp 13.08.2012.
[102] Liu B., op.cit.

i zleca, w jaki sposób wyodrębnione dane powinny zostać przetworzone i zapisane. Następnie skrypt ten jest interpretowany przez aplikację, a robot wykonuje zadania w nim zawarte. W tym prostym typie ekstrakcji danych wszystkie zadania wypełniane są na podstawie reguł ściśle zdefiniowanych przez użytkownika. Literatura naukowa opisuje również – poza podejściem manualnym – metody automatyczne i półautomatyczne. Ze względu na sposób budowy reguł ekstrakcji, Liu dzieli metody wyodrębniania danych na trzy grupy[103]:

1. **Podejście manualne.** Wszystkie reguły ekstrakcji danych użytkownik definiuje ręcznie. Wymagana jest analiza kodu źródłowego stron, jednak wyniki otrzymywane w tym podejściu są najbardziej pewne i poprawne. Reguły mogą być zapisane bezpośrednio w kodzie źródłowym robota lub w postaci pośredniej: języku skryptowym lub szablonie, który jest odczytywany podczas działania aplikacji. Podejście to sprawdza się najlepiej przy niewielkiej liczbie stron, gdy nie ma potrzeby czasochłonnej analizy źródeł danych, nie jest natomiast efektywne dla wielu źródeł.
2. **Uczenie nadzorowane.** Uczy się robota wyodrębniania danych na podstawie wcześniej przygotowanej próbki uczącej. Próbka składa się z dokumentów i ręcznie wyodrębnionych encji danych. Tak przygotowany zbiór uczący służy do automatycznej budowy reguł, wykorzystywanych potem do ekstrakcji. Metoda jest skuteczna dla dużych zbiorów dokumentów o podobnej strukturze.
3. **Ekstrakcja automatyczna.** Reguły konstruowane są na podstawie dużej próbki dokumentów bez wyodrębnionych danych. Budowa reguł ekstrakcji polega na analizie dokumentów i sprawdzaniu, które ich fragmenty ulegają zmianie. Fragmenty te traktuje się jak atrybuty wyodrębnianej encji. Ten sposób jest wydajny przy dużych zbiorach danych, jednak jego wyniki narażone są na występowanie poważnych błędów.

Metody wyodrębniania danych mogą być również podzielone ze względu na sposób, w jaki reprezentowane są w nich dokumenty HTML:

1. **Reprezentacja za pomocą struktury DOM.** Tę najpopularniejszą metodę reprezentacji dokumentów HTML wszystkie przeglądarki internetowe wykorzystują do analizy i generowania strony. Z punktu widzenia ekstrakcji danych, taka reprezentacja sprawdza się najlepiej wtedy, gdy wyodrębniane dane znajdują się w osobnych wierszach tabeli lub elementach blokowych (np. każda wartość atrybutu zawiera się w odrębnym elemencie typu). Reguły wyodrębniania danych w takiej reprezentacji mogą być zapisane na różne sposoby, na przykład przy użyciu języków opracowanych do lokalizacji danych w dokumentach XML takich jak XPath czy XQuery, albo za pomocą selektorów CSS. Metoda ta jest jednak problematyczna dla danych, których nie da się zlokalizować tylko na podstawie struktury tagów, np. jeżeli wartości atrybutów są zapisane tekstem znajdującym się wewnątrz pojedynczego elementu blokowego lub wartość jednego atrybutu jest podzielona na kilka części znajdujących się w różnych fragmentach dokumentu.
2. **Reprezentacja tekstowa.** Źródło dokumentu jest tutaj traktowane jak zwykły tekst. Struktury tagów nie bierze się pod uwagę, natomiast podczas konstrukcji reguł korzysta się z sekwencji wyrażeń regularnych, które określają lokalizację danych w dokumencie. W metodzie tej można zlokalizować wartości atrybutów, które nie mogłyby być łatwo wyodrębnione z użyciem reprezentacji drzewa DOM. Reprezentacja ta jest znacznie bardziej elastyczna, ale też bardziej podatna na błędy. Wyrażenia regularne tworzone manualnie nie muszą być jednoznaczne, mogą wskazywać na kilka różnych fragmentów dokumentu, których autor wyrażenia nie przewidział. Jeżeli struktura dokumentu ulega zmianie, w niektórych dokumentach wartości atrybutów mogą nie zostać znalezione w ogóle. Dodatkowo, wyszukiwanie danych za pomocą wyrażeń regularnych jest mniej efektywne od wyszukiwania w drzewie DOM. Dla niewielkich stron nie stanowi to problemu, jednak długie dokumenty będą przeszukiwane istotnie wolniej.

Roboty służące do ekstrakcji danych są ściśle powiązane z opisaną wcześniej siecią ukrytą. Część problemów z dostępem do zasobów z sieci ukrytej, z którymi nie mogą sobie poradzić roboty indeksujące, może zostać rozwiązana za pomocą mechanizmów analizy struktury dokumentów i ekstrakcji. Duże zbiory ustrukturyzowanych danych najczęściej udostępnia się przez internetowe bazy danych. Rekordy w tych bazach dostępne są zazwyczaj tylko poprzez wyszukiwarkę umieszczoną na stronie. W nielicznych przypadkach serwis udostępnia pełną listę rekordów, co pozwala na zaindeksowanie ich wszystkich przez roboty ogólnego przeznaczenia.

[103] Ibidem.

Jest to możliwe jedynie w sytuacjach, gdy danych nie jest dużo lub podzielone są na kategorie. W pozostałych przypadkach rekordy zwracane są na podstawie zapytań – często dostęp do samej wyszukiwarki jest też ograniczony tylko do użytkowników posiadających konto w serwisie i zalogowanych. Jedynym sposobem na automatyczne przeszukiwanie takich serwisów jest zaimplementowanie dedykowanego robota zawierającego mechanizmy rejestracji i logowania oraz automatycznego formułowania zapytań, które będą sensowne w kontekście przeszukiwanej bazy danych i pobiorą niepustą listę rekordów[104].

Zazwyczaj interfejs wyświetlający wyniki wyszukiwania w internetowej bazie danych składa się z dwóch typów stron:

1. **Listy rekordów.** Każda ze stron zawiera listę obiektów określonego typu. Najczęściej lista ta przedstawiana jest w formie tabeli, w której obiekty umieszczone są w kolejnych wierszach. Taka reprezentacja w łatwy sposób umożliwia wydzielenie danych interesujących dla robota. W każdym z wierszy obiekt zawiera podstawowe informacje, a dalsze szczegóły dostępne są pod linkiem prowadzącym do strony z detalami danego rekordu. Wystąpienie w wierszu kompletnych danych bez konieczności przechodzenia do odnośników nazywane jest płaską strukturą danych.
2. **Strony szczegółowe.** Są to dokumenty szczegółowo opisujące dany rekord. Adres URL do tego typu strony najczęściej zawiera parametr HTTP GET, który w sposób unikatowy identyfikuje dany rekord w bazie. Może on przedstawiać klucz główny w rzeczywistym modelu danych lub też inną niepowtarzalną wartość.

II. Klasyfikatory tekstu

Klasyfikatory bayesowskie oraz Support Vector Machines to klasa algorytmów zaliczających się do metod uczenia z nadzorem (*supervised learning*). W metodach tych, przed procesem klasyfikacji tworzy się zbiory uczące dla klas, które chcemy uzyskać po przeanalizowaniu tekstu przez klasyfikator. Skonstruowany korpus uczący służy do zbudowania odpowiedniego modelu klasyfikacji. Wytrenowany klasyfikator przyporządkowuje odpowiednie klasy do nowych tekstów, uprzednio nieskategoryzowanych[105].

W klasyfikacji dokumentów tekstowych przypisywanie nowego tekstu do uprzednio zdefiniowanych kategorii nazywane jest kategoryzacją (*categorization*). Zadanie to mieści się w dziedzinach wydobywania informacji (*information retrieval*) i uczenia maszynowego (*machine learning*). Przez ostatnie kilkanaście lat tego rodzaju metody stały się bardzo popularne, ze względu na coraz większą ilość informacji dostępnych w formie elektronicznej[106].

1. Klasyfikatory Bayesa

Wszystkie klasyfikatory bayesowskie (naiwne i nienaiwne) wykorzystują do uczenia metody probabilistyczne oparte na twierdzeniu o prawdopodobieństwie warunkowym, które opisuje następujący wzór:

$$P(C|X) = \frac{P(C)\,P(X|C)}{P(X)}$$

gdzie:
C – klasa decyzyjna;
X – zbiór danych wejściowych;
$P(C)$ – prawdopodobieństwo *a priori* występowania klasy C, zdarzenie to nie jest zależne od danych wejściowych;
$P(C|X)$ – prawdopodobieństwo *a posteriori* wystąpienia klasy C przy posiadanych danych X;
$P(X)$ – prawdopodobieństwo pojawienia się danych X na wejściu.

[104] Ibidem.
[105] Hastie T., Tibshirani R., Friedman J., *The Elements of Statistical Learning*, Springer, New York 2009.
[106] Sebastiani F., *Text Categorization*, in: Sirmakessis S., ed., *Text Mining and Its Applications*, Springer, Berlin – Heidelberg 2004.

Istotą rzeczy jest wyznaczenie najbardziej prawdopodobnego zdarzenia $P(C|X)$, dla każdej z klas decyzyjnych. Prawdopodobieństwo *a posteriori* nie jest zależne od mianownika wyżej podanego wzoru, gdyż dla każdej badanej hipotezy wartość $P(X)$ będzie taka sama[107].

W klasyfikacji tekstów danymi wejściowymi będzie wektor X składający się z pewnych atrybutów $[x_1, x_2, x_3, ..., x_n]$. Każdy z tych atrybutów odpowiadać będzie słowu z tekstu do sklasyfikowania. Przyjmując takie założenie, można podstawić w miejsce zdarzenia losowego X wektor reprezentujący to zdarzenie $(x_1, x_2, ..., x_n)$[108]:

$$P(C|x_1, x_2, \ldots x_n) = \frac{P(C)\,P(x_1, x_2, \ldots x_n|C)}{P(x_1, x_2, \ldots x_n)}$$

1.1. Podstawowy klasyfikator Naive Bayes

Naiwne klasyfikatory bayesowskie traktują dokumenty jako tzw. worek słów, przyjmując następujące założenie jako podstawę swojego działania: prawdopodobieństwo występowania każdego słowa x_i w dokumencie d_i jest niezależne wobec występowania każdego z pozostałych przy założeniu, że klasa dokumentu jest znana[109]. Założenie o niezależności słów umożliwia w prosty sposób obliczenie prawdopodobieństwa $P(X|A)$, poprzez oddzielenie obliczania prawdopodobieństwa dla każdego słowa z osobna, co łatwo zaobserwować rozpisując:

$P(x_1, x_2, ..., x_n|C)$

$P(x_1, x_2, ..., x_n|C) =$

$P(x_1|C)\ P(x_2, x_3, ..., x_n|C) =$

$P(x_1|C)\ P(x_2|C)\ P(x_3, x_4, ..., x_n|C) =$

$P(x_1|C)\ P(x_2|C)\ P(x_3|C)\ P(x_4, x_5, ..., x_n|C)\ ...$

Zależność tę przedstawimy w postaci[110]:

$$P(x|C) = \prod_{i=1}^{n} P(x_i|C)$$

Istnieje możliwość, że wartość któregoś z atrybutów w zbiorze testującym nie wystąpi razem z klasą w zbiorze uczącym. Taka sytuacja staje się problematyczna, ponieważ prawdopodobieństwo wystąpienia tego atrybutu w klasie c_i będzie równe zero $P(X = x_1|C = c_j) = 0$. Oznacza to wyzerowanie prawdopodobieństwa wystąpienia dla danej klasy. Istnieje kilka sposobów rozwiązania tego problemu. Najczęściej stosuje się ustalenie małej wartości prawdopodobieństwa skorelowanej dla wszystkich pozostałych prawdopodobieństw[111].

[107] Liu B., op.cit.
[108] Ibidem.
[109] Fragoudis D., Meretakis D., Likothanassis S., *Best terms: An efficient feature-selection algorithm for text categorization*, „Knowledge and Information Systems", 8(1), 16–33, 2005.
[110] Hoare Z., *Landscapes of Naive Bayes classifiers*, „Pattern Analysis and Application", 11(1), 59–72, 2008.
[111] Liu B., op.cit.

1.2. Klasyfikator Multinomial Naive Bayes

Wykorzystuje się go głównie do klasyfikacji dokumentów tekstowych. Model wielomianowy używa wektorów cech liczbowych (liczba wystąpień słów) do przedstawienia dokumentu. Podobnie jak zwykły klasyfikator NB, przyjmuje następujące założenia[112, 113]:

- słowa są niezależne wobec pozostałych wyrazów występujących w tym samym dokumencie;
- długość dokumentu jest niezależna od klasy;
- prawdopodobieństwo słowa jest niezależne względem jego występowania na konkretnej pozycji w tekście.

Biorąc pod uwagę powyższe warunki, każdy dokument może być opisany za pomocą rozkładu wielomianowego. Mówiąc dokładniej zakłada się, że dla każdej stale ustalonej uprzednio liczby klas $C \in \{1, 2, m\}$ istnieje niezmienny zbiór parametrów wielomianowych. Funkcja prawdopodobieństwa klasyfikatora dla dokumentu D_i klasy C_j przedstawiona jest za pomocą poniższego wzoru:

$$P(D_i|C_j;\theta) = P(|D_i|)|D_i|!\prod_{t=1}^{|S|}\left(\frac{P(x_t|C_j;\theta)^{N_{ti}}}{N_{ti}}\right)$$

gdzie:
N_{ti} – liczba określająca liczbę wystąpień słowa x_t w dokumencie D_i;
S – słownik słów, które występują we wszystkich dokumentach;
$P(x_t \mid C_j; \theta)$ – prawdopodobieństwo wystąpienia słowa x_t w kategorii C_j;
$|D_i|$ – długość dokumentu;
$\Sigma_{t=1}^{|S|} N_{ti}$ – suma liczby wystąpień słów w dokumencie (jest równa długości dokumentu);
θ – parametr, który przyjmuje różne wartości dla poszczególnych klas[114].

Przyjmuje się, że długość dokumentu dla każdej z klas jest jednakowa. Wykorzystując podany wyżej wzór oblicza się prawdopodobieństwa wszystkich dokumentów w zbiorze trenującym, dla każdej z klas w zbiorze C. W oparciu o uzyskane prawdopodobieństwa dokumentów pod warunkiem klas możemy przejść do wyznaczenia finalnego prawdopodobieństwa klasy pod warunkiem zadanego dokumentu. Wykorzystując regułę na prawdopodobieństwo warunkowe uzyskujemy taką postać:

$$P(C_j|D_i;\hat{\theta}) = \frac{P(C_j|\hat{\theta}) * P(D_i|C_j;\hat{\theta})}{P(D_i|\hat{\theta})}$$

Podobnie jak w podstawowej formie naiwnego klasyfikatora bayesowskiego, przynależność do klasy opiera się na największej wartości prawdopodobieństwa *a posteriori* wyznaczanej dla każdej klasy osobno[115]:

$$\arg max_{C_j \epsilon C} P(C_j|D_i;\hat{\theta})$$

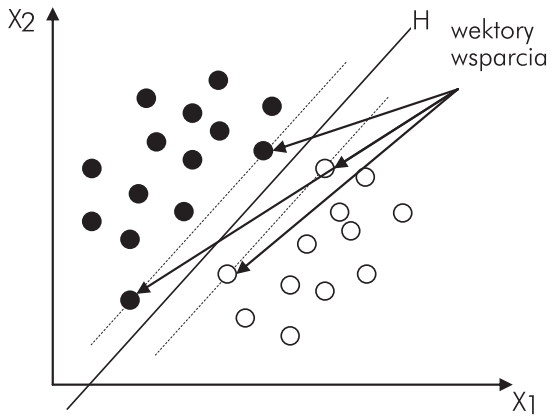
Podstawą działania różnych odmian klasyfikatorów NB jest założenie o niezależności słów względem pozostałych, występujących w tym samym tekście. Takie podejście dyskryminuje klasy posiadające silne zależności pomiędzy słowami. Większość z obecnie stosowanych algorytmów klasyfikacji odrzuca to założenie. Pomimo tego klasyfikacja oparta na modelu naiwnego klasyfikatora bayesowskiego wykazuje się wysoką poprawnością. Dużą zaletą jest również szybkość działania. Proces uczenia modelu klasyfikatora na zbiorze trenującym odbywa się tylko raz, w celu wyznaczenia estymowanych prawdopodobieństw wymaganych do klasyfikacji[116].

[112] Juan A., Ney H., *Reversing and Smoothing the Multinomial Naive Bayes Text Classifier*, in: Iñesta Quereda J.M., Micó L., eds., *Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems*, ICEIS Press, Ciudad Real 2002.
[113] Kibriya A.M., Frank E., Pfahringer B., Holmes G., *Multinomial Naive Bayes for Text Categorization Revisited*, in: Webb G.I., Yu X., eds., *Proceedings of 17th Australian Joint Conference on Artificial Intelligence*, Springer, Berlin 2004.
[114] Liu B., op.cit.
[115] Ibidem.
[116] McCallum A., Nigam K., *A Comparison of Event Models for Naive Bayes Text Classification*, in: *AAAI/ICML-98 Workshop on Learning for Text Categorization*, AAAI Press, Palo Alto 1998.

2. Klasyfikator Support Vector Machine

Jest to stosunkowo nowy algorytm, po raz pierwszy zaprezentowany w 1995 roku przez rosyjskiego matematyka Vladimira Vapnika. Stosuje się go między innymi w problemie klasyfikacji tekstu. Działanie klasyfikatora opiera się na statystycznych metodach uczenia, a dokładniej – na zasadzie strukturalnej minimalizacji ryzyka (*structural risk minimization*). Dla każdej klasy zbioru uczącego algorytm wyróżnia przypadki pozytywne i negatywne, czyli należące i nienależące do badanej klasy. Główna idea polega na znalezieniu takiej powierzchni decyzyjnej, która umożliwi optymalne oddzielenie przypadków negatywnych od pozytywnych. Wektory leżące najbliżej hiperpłaszczyzny nazywane są wektorami wsparcia (rysunek 41), ponieważ uzależniony jest od nich kształt oddzielającej powierzchni. Klasyfikacja nowych tekstów odbywa się w bardzo łatwy sposób, poprzez odwzorowanie ich na wektory wag oraz sprawdzenie, po której stronie hiperpłaszczyzny się znajdują[117].

Rysunek 41. Optymalna hiperpłaszczyzna dzieląca przypadki pozytywne od negatywnych



Źródło: opracowanie własne autorów

Aby poprawić szybkość uczenia się klasyfikatora SVM, często wykorzystywany jest algorytm Sequential Minimal Optimization (SMO). Jego cel to efektywne rozwiązanie problemu optymalizacji podczas trenowania klasyfikatora. SMO dzieli problem na serię mniejszych podproblemów, które rozwiązywane są analitycznie[118].

Podczas nauki może okazać się, że nie istnieje hiperpłaszczyzna jednoznacznie oddzielająca przypadki należące i nienależące do klasy. W związku z tym, wyróżnia się dwie klasy algorytmów opartych na SVM: **separowalne liniowo** (*linear SVM*) i **nieseparowalne liniowo** (*non-linear SVM*).

2.1. Problem separowalny liniowo

Polega na wyznaczeniu takiej przestrzeni podziału, aby margines między wektorami pozytywnymi i negatywnymi był jak największy. Na rysunku 42 zaprezentowany jest przykład punktów separowalnych liniowo przez dwie przestrzenie podziału, z czego większym marginesem dysponuje hiperpłaszczyzna H_2[119].

Powierzchnię decyzyjną w problemie SVM liniowo separowalnym opisuje się za pomocą hiperpłaszczyzny, która reprezentowana jest jako iloczyn skalarny dwóch wektorów wraz z wyrazem wolnym[120]:

$$\vec{w} * \vec{x} - b = 0$$

[117] Silva C., Ribeiro B., *On text-based mining with active learning and background knowledge using SVM*, „Journal of Soft Computing – A Fusion of Foundations, Methodologies and Applications", 11(6), 519–530, 2007.
[118] Joachims T., *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, in: Nedellec C., Rouveirol C., *Proceedings of the 10th European Conference on Machine Learning*, Springer, London 1998.
[119] Liu B., op.cit.
[120] Yang Y., Liu X., *A Re-Examination of Text Categorization Methods*, in: *Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York 1999.

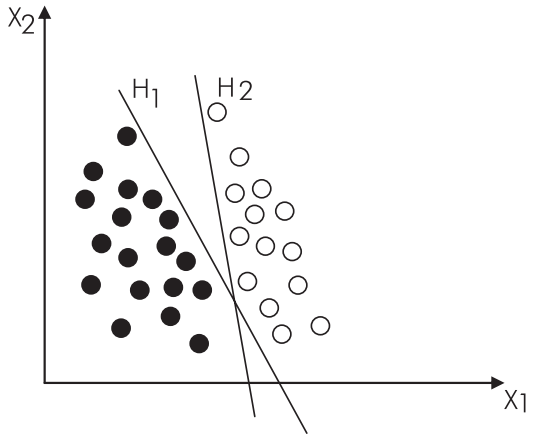
gdzie:
$\vec{w}$ – wektor wag $(w_1, w_2, ..., w_n)$, które zostają wyznaczone w procesie uczenia;
$\vec{x}$ – punkt do klasyfikacji, wektor atrybutów wejściowych $(x_1, x_2, ..., x_n)$;
b – stała wyznaczana w procesie uczenia.

Oznaczając zbiór uczący jako $D = \{(y_i, \vec{x}_i)\}$ oraz zakładając, że $y_i \in \{\pm 1\}$ jest klasyfikacją dla wektora $\vec{x}$ (+1 dla przykładu pozytywnego oraz -1 dla przykładu negatywnego), klasyfikator SVM musi rozwiązać problem znalezienia odpowiedniego $\vec{w}$ oraz wyrazu wolnego b dla poniższych ograniczeń:

$$\vec{w} * \vec{x}_i - b \geq +1 \; dla \; y_i = +1$$
$$\vec{w} * \vec{x}_i - b \leq -1 \; dla \; y_i = -1$$

Problem ten można rozwiązać stosując techniki programowania kwadratowego[121].

Rysunek 42. Zbiór punktów separowalnych liniowo



Źródło: opracowanie własne autorów

2.2. Problem nieseparowalny liniowo

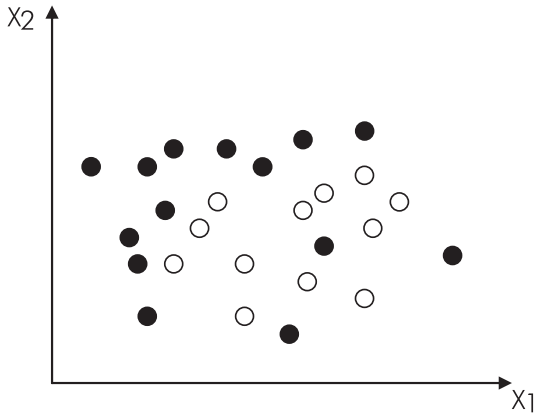
Charakteryzuje się brakiem możliwości poprowadzenia takiej hiperpłaszczyzny, aby oddzielić przykłady negatywne od pozytywnych (rysunek 43).

Algorytm rozwiązujący SVM dla problemu liniowo separowalnego można w łatwy sposób rozszerzyć na rozwiązywanie problemu nieseparowalnego. Stosuje się w tym celu dwie metody[122]: metodę elastycznego marginesu (*soft margin*) oraz zastosowanie funkcji jądra (*kernel function*).

Metoda elastycznego marginesu zakłada możliwość przekroczenia niektórych przykładów pozytywnych na drugą stronę powierzchni decyzyjnej, lub odwrotnie. Nie ma to wpływu na ostateczny wynik, jednak trzeba pamiętać, by liczba takich wyjątków nie była zbyt duża (mogłoby to spowodować zakłócenia w procesie klasyfikacji). Z tego względu wprowadza się parametr określający maksymalną liczbę punktów, które mogą błędnie przekroczyć hiperpłaszczyznę. Na rysunku 44 przedstawiono hiperpłaszczyznę *H*, oddzielającą przypadki negatywne od pozytywnych. W tym przykładzie jeden z punktów znajduje się po przeciwnej stronie przestrzeni decyzyjnej i nie jest uwzględniany przy konstrukcji hiperpłaszczyzny[123].

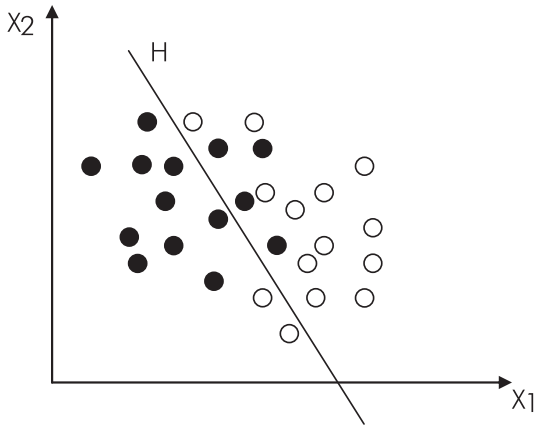
[121] Ibidem.
[122] Noble W., *What is a support vector machine?*, „Nature Biotechnology", 24, 1565–1567, 2006.
[123] Ibidem.

Rysunek 43. Zbiór punktów nieseparowalnych liniowo



Źródło: opracowanie własne autorów

Rysunek 44. Zbiór punktów nieseparowalnych liniowo z marginesem błędu



Źródło: opracowanie własne autorów

Zastosowanie funkcji jądra polega na zamianie oryginalnych wektorów wejściowych na wektory o większym wymiarze. Takie przekształcenie umożliwia konstrukcję w nowej przestrzeni hiperpłaszczyzny, oddzielającej przypadki pozytywne od negatywnych. Należy pamiętać o umiarkowanym zwiększaniu wymiarów, gdyż wraz ze wzrostem liczby atrybutów wzrasta również wykładniczo liczba możliwych rozwiązań oraz następuje zbyt duże dopasowanie rozwiązania do zbioru uczącego. Dlatego też po zwiększeniu wymiaru najczęściej stosuje się metodę *soft margin*[124]. Wyróżnia się kilka podstawowych funkcji jądra[125]:

- wielomianowa jednorodna:

$$k(x_i, x_j) = (x_i * x_j)^d$$

- wielomianowa niejednorodna:

$$k(x_i, x_j) = (x_i * x_j + 1)^d$$

[124] Noble W., op.cit.
[125] Wikipedia, *Support vector machine*, http://en.wikipedia.org/wiki/Support_vector_machine, dostęp 13.08.2012.

- gaussowska radialna funkcja bazowa:

$$k(x_i, x_j) = exp(-\gamma|| x_i - x_j||^2)^d, \qquad dla\ \gamma > 0$$

- hiperboliczny tangens:

$$k(x_i, x_j) = tanh(\kappa x_i * x_j + z)^d, przeważnie\ dla\ \kappa > 0\ i\ z < 0$$

gdzie:
d – stopień równania.

3. Optymalizacja parametrów klasyfikatora SVM (algorytm Differential Evolution)

Ustawienia poszczególnych funkcji jądra klasyfikatora SVM mają wpływ na proces uczenia, a to przekłada się na jakość klasyfikacji. Problem znalezienia optymalnych parametrów jest problemem nieliniowym, wielowymiarowym, posiadającym dużą przestrzeń potencjalnych rozwiązań, najczęściej wiążącym się z dużym nakładem obliczeniowym. Dlatego do celów optymalizacji ustawień funkcji jądra wykorzystuje się metody ewolucyjne[126].

Differential Evolution (DE) jest algorytmem należącym do grupy metod ewolucyjnych. Wykorzystuje się go jako standardowe podejście do optymalizacji ustawień zadanego problemu, gdzie danymi wejściowymi są parametry reprezentowane najczęściej jako *NP D*-wymiarowy wektor[127]:

$$x_{D,G}, i = 1,2,\ldots,NP$$

gdzie:
G – numer generacji;
NP – liczba wektorów w każdej z generacji;
i – numer wektora w populacji;
D – numer parametru w wektorze.

Optymalizacja parametrów odbywa się według następującego schematu[128]:

1. **Inicjalizacja.** Utworzenie losowej populacji o rozmiarze NP potencjalnych rozwiązań, przy określeniu progowych wartości dla każdego parametru.
2. **Mutacja.** Dla każdego wektora $x_{D,G}, i = 1,2,\ldots,NP$ tworzony jest wektor zmutowany:

$$v_{d,G+1} = x_{r1,G} + F * (x_{r2,G} - x_{r3,G})$$

z losowymi indeksami $r1, r2, r3 \in \{1, 2, \ldots, NP\}$, gdzie każdy z nich musi być różny od siebie, stąd minimalna liczba wektorów w populacji wynosi 4. F jest stałą, która musi spełniać następujące warunki: $F > 0$, $F \in [0, 2]$.

3. **Rekombinacja.** W celu zwiększenia różnorodności buduje się wektor:

$$u_{i,G+1} = [x_{D1,G+1}, x_{D2,G+1}, \ldots, x_{Di,G+1}]$$

gdzie:

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{,jeżeli } (rand\ (j) \le CR) \text{ lub } j = rnbr\ (i) \\ x_{ji,G} & \text{,jeżeli } (rand\ (j) > CR) \text{ i } j \ne rnbr\ (i) \end{cases}$$

[126] Huang C.L., Wang C.J., *A GA-based feature selection and parameters optimization for support vector machines*, „Expert Systems With Applications", 31(2), 231–240, 2006.
[127] Storn R., Price K., *Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces*, „Journal of Global Optimization", 11, 341–359, 1997.
[128] Ibidem.

dla $j = 1, 2, ..., D$, $rand(j)$ określającą losową liczbę z przedziału [0,1], $rnbr(i)$ określa losowy indeks $\in 1,2, \ldots, D$ – co zapewnia, że nowo powstały wektor $u_{i, G+1}$ będzie miał przynajmniej jeden parametr ze zmutowanego wektora $v_{d, G+1}$. CR jest stałą rekombinacji z przedziału [0,1], definiowaną przez użytkownika.

4. **Selekcja.** Decyzja o pozostawieniu nowo utworzonego kandydata $u_{i, G+1}$ odbywa się poprzez wyznaczenie funkcji kosztu. Jeśli kandydat ma mniejszą funkcję celu, to jest uwzględniony w nowym pokoleniu, w przeciwnym przypadku do nowego pokolenia przechodzi stary wektor $x_{i, G}$.
5. **Warunek stopu.** Ustalana jest stała liczba generacji, po osiągnięciu której algorytm kończy swoje działanie i zwraca najlepsze dotychczasowe rozwiązanie.

III. Metody analizy tekstu

1. Ogólny algorytm analizy

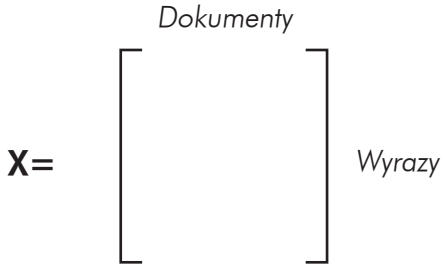
Większość metod stosowanych do wykrywania terminów wykorzystuje statystykę, przetwarzanie języka naturalnego oraz techniki eksploracji danych. Powszechnie stosowane ujęcia najpierw identyfikują potencjalnych kandydatów, a następnie określają ich poziom istotności w celu selekcji grupy kluczowych terminów.

Etapy analizy przebiegają następująco[129]:

1. Przekształcenie dokumentów źródłowych do zunifikowanej postaci:
   a) wstępne przygotowanie dokumentów (transformacja do postaci tekstowej, usunięcie znaków formatujących, ujednolicenie sposobu kodowania);
   b) wybór sposobu reprezentacji informacji zawartej w dokumentach:
      - *bag-of-words* (wektor słów opisanych częstościami);
      - reprezentacja w postaci ustrukturyzowanej.
2. Wstępne przetwarzanie zunifikowanych dokumentów:
   a) czyszczenie, normalizacja, uzupełnianie informacji;
   b) budowanie modelu na podstawie zadanej metody reprezentacji.
3. Identyfikacja potencjalnych kandydatów.
4. Selekcja ważnych terminów.

W przypadku reprezentacji *bag-of-words* wyrazy pochodzące z wektorów odpowiadających poszczególnym dokumentom są łączone w jedną, wspólną macierz. Zebrane dane tworzą macierz częstości *X*, która przyjmuje postać jak na rysunku 45.

Rysunek 45. Macierz częstości



Źródło: opracowanie własne autorów

Liczbę wystąpień *i*-tego słowa w *j*-tym dokumencie określa *i*, *j*-ty element macierzy. Analiza macierzy pozwala na badanie podobieństwa słów oraz dokumentów. Wyznaczone częstości wystąpień można poddać transformacji. Przykładowe transformacje to reprezentacja binarna (istotny jest sam fakt występowania słowa, a nie jego liczba wystąpień) oraz ważona reprezentacja logarytmiczna (uwzględnia liczbę dokumentów zawierających dane słowo).

[129] Konchady M., *Text Mining Application Programming*, Charles River Media, Rockland 2006.

Reprezentacja ustrukturyzowana[130] wzbogaca analizę informacji zawartych w poszczególnych słowach o informacje ukryte w strukturach językowych. Stosowane są struktury danych odpowiednie do przechowywania związków wynikających z kolejności wyrazów, charakterystyk obiektów, relacji między nimi oraz zależności przyczynowo-skutkowych. Przykładowe struktury to łańcuchy znaków, drzewa i grafy. Wadami tej reprezentacji są trudności z przekształceniem dokumentu tekstowego w postać ustrukturyzowaną oraz mały wachlarz metod analitycznych do operowania na informacjach przechowywanych w złożonych strukturach danych. Pierwsze prace dotyczące strukturyzacji tekstów próbowały reprezentować dokumenty w relacyjnych tabelach baz danych. Inne podejścia opierały się na budowaniu skomplikowanych plików XML. Jeszcze inne zakładały przynależność dokumentów do jednej konkretnej dziedziny i wykorzystywały modelowanie przy pomocy ontologii. Definicja ontologii obejmuje opis obiektów występujących w rzeczywistości oraz opis zależności między nimi. Obiekty są reprezentowane przez struktury danych zwane encjami. Mają one zwykle strukturę hierarchiczną, co powoduje, że reprezentowane są w postaci struktur drzewiastych. W ogólnym przypadku przekształcanie informacji zawartych w tekstach do postaci ustrukturyzowanej nawet na poziomie prostych tabel nie jest zadaniem łatwym. Proces ten najczęściej wykonuje się manualnie, co z definicji zabiera dużo czasu.

Szczególnie perspektywiczne wydaje się łączenie koncepcji reprezentacji ontologicznej z odkrywaniem wiedzy z tekstów. Nie chodzi tu o samo reprezentowanie tekstów za pomocą identyfikacji obiektów i ich relacji opisanych w zadanej ontologii. Zagadnieniem o dużym potencjale jest uczenie ontologii za pomocą eksploracji danych tekstowych. Analiza statystyczna i *data mining* tekstów może doprowadzić do identyfikacji najistotniejszych obiektów i związków między nimi[131].

Trudno zbudować reprezentację w postaci ustrukturyzowanej, dlatego najczęściej używany jest klasyczny model wektorowy (*bag-of-words*) z miarami częstości (istotności) terminów – TF_t, df_t, tdf_t, $tf\text{-}idf$[132]:

- TF_t – częstość globalna terminu t (liczba wystąpień terminu w korpusie), jest to rozszerzenie częstości terminu w ramach dokumentu;
- df_t – liczba dokumentów, w których wystąpił termin t;
- $tdf_t = avg(tf_t)\ x\ df_t$ – znormalizowana częstością dokumentową średnia częstość terminu w dokumentach;
- $tf\text{-}idf = tf_{t,d}\ x\ log(N/df_t)$ – częstość terminu t w dokumencie d znormalizowana logarytmem odwróconej częstości dokumentowej.

Etap identyfikacji potencjalnych terminów poprzedza faza wstępnego przetwarzania dokumentu. Pierwszy krok to dokonanie podziału tekstu na akapity, zdania oraz słowa, czyli tzw. tokenizacja. Kolejnymi fazami są czyszczenie i normalizowanie danych. Czyszczenie sprowadza się do usunięcia wszystkich nieistotnych słów (przyimków, zaimków *etc.*), czyli *stop words*. Normalizacja danych ogranicza się do zagregowania wyrazów – tak, aby te same słowa w różnych formach zostały zaliczone do tej samej grupy. Najczęściej stosuje się takie techniki, jak stemming, lematyzacja i analiza morfologiczna. **Stemming** polega na sprowadzeniu słów do wspólnego rdzenia, który nie musi być poprawnym słowem, **lematyzacja** – na sprowadzeniu każdego słowa do jego podstawowej formy, a **analiza morfologiczna** – na odnajdywaniu odpowiednich form słów w ustalonym słowniku. W wyniku tej analizy dostajemy – oprócz bazowej formy słownej – także informacje o części mowy czy przypadku[133].

2. Metody wykrywania terminów

Można je podzielić ze względu na rodzaje technik używanych do identyfikacji istotnych terminów – lingwistyczne i statystyczne.

Metody lingwistyczne opierają się na wykorzystaniu kryteriów syntaktycznych. Większość naukowców zgadza się z poglądem, iż terminy są głównie frazami rzeczownikowymi. Justeson i Katz[134] badali korpusy teksto-

[130] Manning C., Schutze H., Prabhakar R., *Introduction to Information Retrieval*, Cambridge University Press, 2008.
[131] Cimiano P., *Ontology Learning and Population from Text*, Springer, Berlin – Heidelberg 2010.
[132] Salton G., McGill M.J., *Introduction to Modern Information Retrieval*, McGraw-Hill, New York 1983.
[133] Charniak E., *Statistical techniques for natural language parsing*, „AI Magazine", 18(4), 33–43, 1997.
[134] Justeson J., Katz S., *Technical terminology: Some linguistic properties and an algorithm for identification in text*, „Natural Language Engineering", 1, 9–27, 1995.

we głównie za pomocą fraz rzeczownikowych, zawierających przymiotniki i czasowniki, aczkolwiek w bardzo małych proporcjach. Do wyszukiwania frazy rzeczownikowej używają oni poniższego wyrażenia regularnego:

$$((Adj|Noun)^{+}|((Adj|Noun)^{*}(NounPrep)^{?})(Adj|Noun)^{*})Noun$$

gdzie:
Adj – przymiotnik;
Noun – rzeczownik;
NounPrep – przyimek odrzeczownikowy.

Także Daille[135] koncentrował się na frazach rzeczownikowych. Wzorce pojęć dla złożonych struktur składają się głównie z dwóch elementów – rzeczowników i przymiotników. Wzorce dla języka angielskiego to przymiotnik – rzeczownik i rzeczownik – rzeczownik.

Metody statystyczne polegają na liczeniu częstości wystąpień potencjalnych terminów, gdyż najbardziej istotne pojęcia powinny się powtarzać w zbiorze dokumentów. Wczesne podejścia do automatycznego wykrywania pojęć były skupione na obliczeniach statystyk pojedynczych słów w kontekście całego korpusu. Jones[136] opisał badania nad identyfikacją statystycznie dyskryminujących słów w korpusie dokumentów. Andrade i Valencia[137] bazowali na porównywaniu rozkładu słowa w ramach zadanego dokumentu do rozkładu tego słowa w korpusie referencyjnym. Niestety, łańcuchy tekstowe pojawiające się często mogą nie być rzeczywistymi terminami, a istotne terminy niekoniecznie muszą przekroczyć zadany poziom istotności. Poleganie na statystycznym poziomie wsparcia nie pozwala również na wykrycie rzadkich pojęć.

Aktualnie powszechne jest scalanie podejścia statystycznego z informacjami lingwistycznymi. Przykładem może być składająca się z dwóch części metoda *C-Value/NC-Value*[138]. *C-value* ma wykrywać terminy zagnieżdżone wielosłowne. Kandydujące terminy są porządkowane według *C-value* – większa wartość oznacza większe prawdopodobieństwo bycia istotnym terminem. *C-value* wykorzystuje wiedzę na temat części mowy (*POS tagging*), filtrowanie nieistotnych słów, a także rozkład statystyczny. *NC-value* jest rozszerzeniem *C-value*, które wstrzykuje kontekstową informację w celu poprawy identyfikacji istotnych terminów. Idea tego usprawnienia polega na uwzględnieniu słów sąsiadujących z terminem. Słowo jest identyfikowane jako kontekstowe za pomocą liczby istotnych pojęć w sąsiedztwie których ono występuje. Większa liczba oznacza wyższe prawdopodobieństwo, iż mamy do czynienia ze słowem silnie skorelowanym z ważnymi pojęciami. Frantzi, Ananadiou i Mima[139] pokazali, że słowa kontekstowe są dziedzinowo zależne i przypisywanie wysokich wag do słów, które towarzyszą pojęciom, poprawia proces ekstrakcji kluczowych terminów.

Metody wykrywania terminów można podzielić też według kryterium zorientowania na pojedynczy dokument lub korpus. **Metody zorientowane na pojedynczy dokument** to na przykład przypisanie słów kluczowych do dokumentu lub automatyczna sumaryzacja (tworzenie streszczeń). **Metody oparte na przetwarzaniu korpusu tekstowego** w celu identyfikacji istotnych terminów będą służyły choćby do konstrukcji dziedzinowej ontologii. Automatyczne streszczanie oraz budowanie ontologii są szczególnymi przypadkami pośredniego wykorzystania ekstrakcji istotnych terminów z tekstów.

Automatyczne tworzenie streszczeń ma istotne znaczenie praktyczne w obszarze wyszukiwania informacji. Jednym z podejść do sumaryzacji jest stosowanie ekstrakcji tekstu, które polega na wybieraniu zdań lub akapitów z oryginalnego dokumentu. Jest to tzw. podejście otwarte: nie ma żadnych założeń odnośnie do treści streszczanego tekstu. Oceny istotności dokonuje się dynamicznie na podstawie analizy zawartości dokumentu. W systemach ekstrakcji tekstów podstawowy etap to wykrycie istotnych terminów. W klasycznej pracy Luhna[140] selekcja istotnych zdań opiera się na danych statystycznych. Waga zdania zależy od wagi skła-

[135] Daille B., Gaussier E., Lange J., *Towards Automatic Extraction of Monolingual and Bilingual Terminology*, in: *Proceedings of COLING 94*, COLING, Kyoto 1994.
[136] Jones K., *A statistical interpretation of term specificity and its application in retrieval*, „Journal of Documentation", 28(1), 11–21, 1972.
[137] Andrade M., Valencia A., *Automatic extraction of keywords from scientific text: Application to the knowledge domain of protein families*, „BioInformatics", 14(7), 600–607, 1998.
[138] Frantzi K., Ananiadou S., Mima H., *Automatic recognition of multiword terms*, „International Journal of Digital Libraries", 3(2), 117–132, 2000.
[139] Ibidem.
[140] Luhn H., *The automatic creation of literature abstracts*, „IBM Journal of Research and Development", 156–165, 1958.

dowych słów, obliczanej na podstawie częstości ich występowania w tekście. U Kupca, Pedersena i Chena[141] każde zdanie jest opisywane wektorem cech (m.in. długość zdania i słowa kluczowe). Następnie stosuje się naiwny klasyfikator bayesowski, który po fazie uczenia kategoryzuje podane zdanie jako istotne lub nieistotne. Występowanie istotnych pojęć w sentencjach jest jedną z głównych składowych branych pod uwagę przy decydowaniu, czy zdanie ma wchodzić w skład streszczenia dokumentu.

Umiejętność automatycznej budowy ontologii jest konieczna do powstania nowej generacji inteligentnych rozwiązań informacyjnych. Tworzenie reprezentacji wiedzy z tekstów jest zadaniem trudnym, składającym się z wielu etapów. Jednym z nich jest proces odkrywania istotnych pojęć ontologii w ramach zadanego, dziedzinowego korpusu tekstowego. W tym celu w systemie TextOntoMiner[142, 143] najpierw przeprowadza się analizę morfologiczną (opisanie części mowy), a potem wyszukuje kandydujące pojęcia na podstawie wiedzy eksperckiej (zapisanych ręcznie reguł charakteryzujących pojęcia w oparciu o związki rzeczowników z innymi częściami mowy). Ostatecznie, jako istotne pojęcia wybierani są kandydaci o odpowiednio wysokim wsparciu w dokumentach i zdaniach. W przypadku odkrywania konceptów ontologicznych analizowane są wszystkie dokumenty zadanego zbioru.

3. Metody ekstrakcji słów kluczowych

Celem przypisywania słów kluczowych jest znalezienie małego zbioru terminów, które opisują wybrany dokument, niezależnie od dziedziny, do której przynależy. Słowa kluczowe to sekwencje jednego lub wielu wyrazów, które dostarczają spójnej i zwartej reprezentacji zawartości dokumentu. Mimo ich użyteczności w analizie, indeksowaniu i wyszukiwaniu, większość dokumentów nie posiada przypisanych słów kluczowych.

3.1. Analiza *n*-gramów

U Hulth[144] automatyczna ekstrakcja słów kluczowych traktowana jest jak uczenie maszynowe z nadzorem. Pojęcia stanowiące potencjalne słowa kluczowe w dokumencie badane są na kilku płaszczyznach tokenizacji. Pierwsza, najbardziej popularna to analiza *n*-gramów. Kolejna, alternatywna to poszukiwanie fraz rzeczownikowych (rzeczownik – rzeczownik, rzeczownik – przymiotnik). Ostatnim z badanych podejść jest selekcja pojęć przy użyciu wzorców morfologicznych (niosą one informacje o częściach mowy i liczbie pojedynczej lub mnogiej). Używane są cztery deskryptywne cechy:

- częstość kandydujących pojęć w ramach dokumentu;
- częstość kandydujących pojęć w ramach całego zbioru dokumentów;
- względna pozycja pierwszego wystąpienia kandydującego pojęcia;
- morfologiczne właściwości kandydata.

Klasyfikator trenowany jest przy użyciu zbioru dokumentów z przypisanymi słowami kluczowymi. Każdy potencjalnie istotny termin zawarty w nowym dokumencie jest klasyfikowany jako kluczowy lub nie. Potencjalne terminy oznaczają w tym wypadku znormalizowane unigramy, bigramy i trigramy. Mamy do czynienia z klasyfikacją binarną opartą na indukcji reguł. Strategia użyta do konstrukcji reguł to „dziel i rządź", której celem jest maksymalizacja odseparowania między kategoriami opisanymi przez zestaw reguł.

3.2. Rapid Automatic Keyword Extraction

RAKE[145] opiera się na obserwacji, iż słowa kluczowe często zawierają wiele składowych, natomiast rzadko znaki interpunkcyjne, przyimki, zaimki czy inne słowa o minimalnym leksykalnym znaczeniu. Do podziału

[141] Kupiec J., Pedersen J., Chen F., *A Trainable Document Summarizer*, in: Fox E., Ingwersen P., Fidel R., *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Seattle 1995.
[142] Rybiński H. i inni, *Text mining tools for ontology maintenance*, reports from phase 2–5, Wydział Elektroniki i Technik Informacyjnych Politechniki Warszawskiej, 2006–2007.
[143] Gawrysiak P., Rybiński H., Protaziuk G., *Text Onto Miner – A Semi Automated Ontology Building System*, in: *Proceedings of 17th International Symposium on Intelligent Systems*, 563–573, Springer, Berlin – Heidelberg 2008.
[144] Hulth A., *Improved Automatic Keyword Extraction Given More Linguistic Knowledge*, in: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, 216–223, ACL, Stroudsburg 2003.
[145] Rose S., Engel D., Cramer N., Cowley W., *Automatic Keyword Extraction from Individual Documents*, in: Berry M.W., Kogan J., *Text Mining: Applications and Theory*, 19–37, Wiley, New York 2010.

dokumentu na zbiór kandydujących słów kluczowych metoda używa *stop words* i ograniczników sentencji (kropka, wykrzyknik *etc.*). Zidentyfikowane słowa mogą składać się z jednego lub wielu wyrazów. Kolejnym krokiem jest obliczanie macierzy współwystępowania słów zawartych w kandydujących słowach kluczowych.

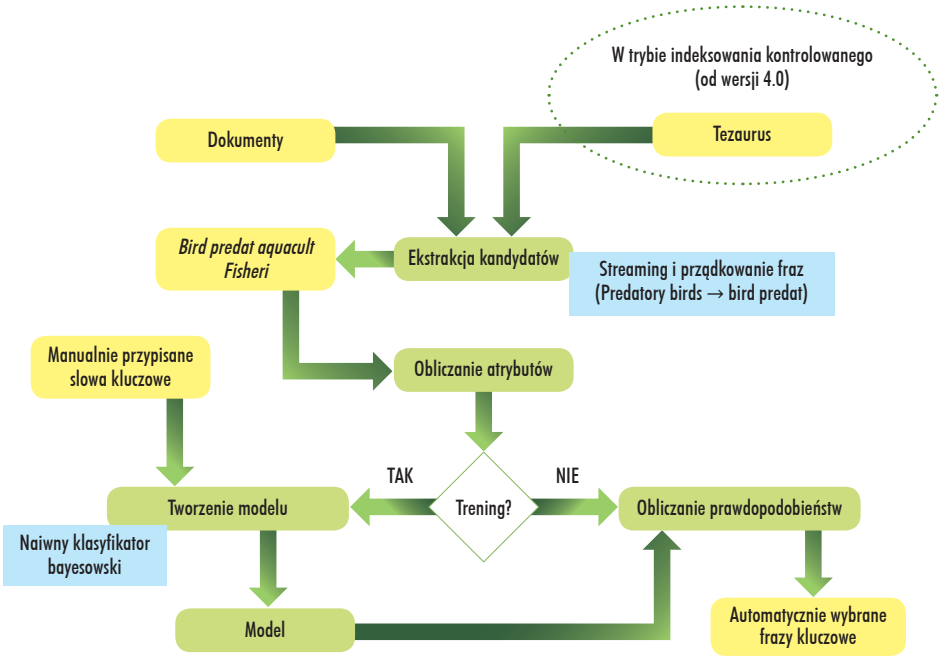
Ostatnie zadanie polega na obliczeniu współczynnika stopnia względem częstości dla każdego słowa w macierzy. Częstość słowa *freq(w)* oznacza ilość wystąpień w całym dokumencie. Stopień słowa *deg(w)* dodatkowo uwzględnia ilość wystąpień słowa *w* w dłuższych kandydujących słowach kluczowych. Finalnym wskaźnikiem jakości słowa jest współczynnik stopnia względem częstości: *deg(w)/freq(w)*.

Wagę kandydującego słowa kluczowego liczy się przez zsumowanie współczynników stopnia względem częstości kolejnych składowych słowa kluczowego. Kandydaci o *T* najwyższych wagach wybierani są jako istotne słowa kluczowe dla zadanego dokumentu. Najczęściej liczba *T* odpowiada od jednej czwartej do jednej trzeciej wszystkich kandydujących słów kluczowych.

3.3. Keyphrases Extraction Algorithm

KEA[146] jest metodą do ekstrakcji kluczowych fraz z dokumentów tekstowych, opracowaną na Uniwersytecie Waikato w Nowej Zelandii. Może być używana zarówno do swobodnego indeksowania tekstów, jak i do kontrolowanego indeksowania (przy założeniu posiadania słowników dziedzinowych). Metoda ta została zaimplementowana w Javie w postaci otwartego projektu dystrybuowanego z licencją GNU GPL. Parametry wejściowe, które trzeba dostarczyć to stop-lista oraz algorytm normalizacji słów (stemmer bądź lematyzator).

Rysunek 46. Schemat działania Keyphrases Extraction Algorithm



Źródło: opracowanie własne autorów na podstawie: http://www.nzdl.org/Kea/description.html, dostęp 14.08.2012.

[146] Witten I.H., Paynter G.W., Frank E., Gutwin C., Nevill-Manning C.G., *Automatic Keyphrase Extraction*, working paper 00/5, Department of Computer Science, The University of Waikato, 2000.

Algorytm przegląda tekst i dokonuje selekcji *n*-gramów o ustalonej długości, które nie zaczynają się i nie kończą elementami *stop words*. Dla każdego kandydata algorytm oblicza cztery wskaźniki:

- TF-IDF;
- *first occurence* – preferuje kandydatów pojawiających się na początku lub końcu dokumentu;
- *length* – liczba składowych słów, najczęściej preferowane przez ludzi są bigramy;
- *node degree* – liczba kandydatów semantycznie powiązanych z zadaną frazą; tylko do użytku z dziedzinowymi słownikami.

Charakterystyczny jest etap budowania modelu słowa kluczowego. Dla zbioru dokumentów z manualnie przypisanymi słowami kluczowymi przeprowadza się fazę uczenia, która skutkuje powstaniem probabilistycznego modelu (dokładniej – klasyfikatora *Naive Bayes*). Jest to metoda uczenia maszynowego z nadzorem. Ostatecznie, dla każdego kandydata, przy użyciu klasyfikatora i wskaźników oblicza się prawdopodobieństwo bycia słowem kluczowym. Kandydaci z najwyższymi prawdopodobieństwami wybierani są jako automatycznie wygenerowane słowa kluczowe. Cały proces został przedstawiony na rysunku 46.

3.4. Text Rank

Jest to grafowy model rang używany do przetwarzania tekstów, w tym też do ekstrakcji słów kluczowych; metoda uczenia bez nadzoru. Słowa kluczowe składają się z od jednego do *n* słów składowych. Każde składowe słowo jest modelowane jako węzeł w grafie, a każda relacja między unigramami może być zamodelowana krawędzią. Text Rank używa relacji współwystępowania ograniczonej przez odległość między unigramami. Dwa wierzchołki są połączone, jeśli odpowiadające im unigramy współwystępują w tekście w ustalonym odgórnie oknie *n* słów. W algorytmie zastosowano potencjalne ograniczenia, iż węzłami mogą być tylko rzeczowniki i przymiotniki. Po zbudowaniu grafu rangi każdego węzła są ustawiane na 1. Formalizując, niech $G = (V, E)$ będzie grafem o wierzchołkach V i krawędziach E, gdzie $E = V \times V$. Dla danego wierzchołka V_i niech $In(V_i)$ będzie zbiorem krawędzi, które wskazują na wierzchołek i, analogicznie $Out(V_i)$ będzie zbiorem krawędzi wychodzących, a w_{ij} wagą krawędzi łączącej wierzchołki V_i i V_j. Wartość oceny wierzchołka V_i wyniesie:

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

W powyższym wzorze d to parametr tłumiący z przedziału między 0 a 1, zazwyczaj przyjmuje on wartość 0,85. Uruchamiany jest algorytm liczący rangi iteracyjne, aż zostanie osiągnięta konwergencja poniżej zadanego poziomu. Ostatecznie, wierzchołki są sortowane według rang i zwracana jest lista wynikowa.

3.5. Polish Keywords Extractor

PKE jest metodą ekstrakcji słów kluczowych dedykowaną polskojęzycznym tekstom, opracowaną w projekcie „System wspomagania wyboru recenzentów". Wyodrębnianie fraz kluczowych rozpoczyna się od wstępnego przetworzenia tekstu. Na tym etapie tekst dokumentu powinien zostać podzielony na zdania i słowa. Dla każdego ze słów przeprowadza się lematyzację; określone zostają charakterystyki słowa: liczba, część mowy, rodzaj *etc.* W kolejnych fazach najpierw odbywa się identyfikacja potencjalnych kandydatów na słowa kluczowe, a następnie – ich ocena i prezentacja zadanej liczby finalnych słów kluczowych.

Na podstawie algorytmu RAKE podjęto próby wypracowania lepszych rozwiązań identyfikacji słów kluczowych. Pierwszą wprowadzoną modyfikacją było filtrowanie części mowy. Domyślnie RAKE nie bierze pod uwagę części mowy składających się na potencjalne słowo kluczowe. Przykładowo: czasownik + czasownik jest równie dobrym kandydatem co przymiotnik + rzeczownik, natomiast czasowniki bardzo rzadko są słowami kluczowymi. Początkowo przyjęto założenie, iż słowo kluczowe musi mieć postać: przymiotnik/przysłówek/rzeczownik + dowolna liczba rzeczowników. Kolejnym ulepszeniem było założe-

nie, że wyrazy obcego pochodzenia można traktować w powyższym szablonie na równi z rzeczownikiem. Następnie wykorzystano obserwację, że wyrazy zaczynające się od dużej litery i niebędące początkiem zdania są istotne i bez względu na część mowy powinny być traktowane na równi z rzeczownikiem. Ostatnie usprawnienie polegało na wprowadzeniu listy wyjątków w filtrze części mowy. Listą wyjątków jest zbiór sekwencji części mowy, które są statystycznie istotne wśród manualnie przypisanych przez autorów słów kluczowych. Problemem tego podejścia jest nadmiarowość nieistotnych kandydatów, charakteryzujących się dużą długością.

Alternatywnie dla powyższej metody identyfikacji kandydatów opracowano metodę rekurencyjną, która oddaje mniejszy, dokładniejszy zbiór kandydatów. Opiera się ona na obserwacji, że w polskojęzycznych dokumentach słowa kluczowe składają się przeważnie z następujących części mowy: rzeczownik, rzeczownik + przymiotnik, rzeczownik + rzeczownik oraz rzeczownik + symbol. Pod pojęciem „symbol" rozumie się wyrazy, dla których nie udało się poprawnie określić części mowy, na przykład: „sektor MSP", „stop AZ91". Aby zastosować powyższą zasadę, badany tekst dzieli się na ciągi słów, gdzie elementy rozdzielające to:

- części mowy inne niż: rzeczownik, przymiotnik, symbol;
- elementy listy *stop words*;
- elementy niespełniające następującego wyrażenia regularnego: *[\\p{Lu}\\p{Ll}0-9-]+$* (elementy zawierające znaki inne niż: litery, cyfry i „-").

Niestety, powyższe kryterium jest niewystarczające. W języku polskim dość często pojawiają się długie ciągi rzeczowników, przymiotników i symboli. Na przykład, w badanych tekstach abstraktów znaleziono następujący ciąg wyrazów: *istotne zróżnicowanie wartości względnego modułu sprężystości rdzenia kolb badanych mieszańców kukurydzy*. Nawet w razie wyboru jedno- i dwuelementowych podciągów otrzymano by 11 potencjalnych słów kluczowych zawierających jedno słowo i 10 zawierających dwa słowa. Ten problem uwzględniono poprzez stworzenie rekurencyjnego algorytmu selekcji istotnych fraz.

Dla badanego ciągu słów iteracyjnie sprawdza się, czy można znaleźć jeden z poniższych wzorców (kolejność jest istotna):

1. rzeczownik + przymiotnik;
2. rzeczownik + symbol;
3. rzeczownik + rzeczownik;
4. rzeczownik.

Jeżeli uda się dopasować wzorzec, wybrane słowa zostają dodane do wynikowej listy potencjalnych słów kluczowych. Następnie algorytm rekurencyjnie przeszukuje ciąg słów poprzedzający i następujący po wybranym słowie kluczowym.

Ocena kandydatów może być dokonana na podstawie podejścia statystycznego lub metod uczenia maszynowego.

W algorytmie RAKE zaproponowano następujące statystyczne miary oceny[147]:
$freq(w)$ – liczba wystąpień;
$deg(w)$ – liczba wystąpień i współwystąpień;
$\frac{deg(w)}{freq(w)}$ – iloraz dwóch powyższych.

Podczas autorskich badań wprowadzono miarę $(freq(w))^2$. Niezależnie od przyjętej miary, całkowita miara istotności słów kluczowych była sumą wag jego słów składowych. Miara $freq(w)$ jest liczbą wystąpień słowa w badanym tekście i preferuje słowa występujące w tekście często, bez uwzględnienia słów, z którymi one współwystępują. W przypadku $deg(w)$ liczy się zarówno liczbę wystąpień słowa, jak i liczbę słów, z którymi badane słowo współwystępuje; preferowane są słowa występujące często w długich frazach. Miara $\frac{deg(w)}{freq(w)}$

[147] Rose S., Engel D., Cramer N., Cowley W., op.cit.

wysoko oceni wyrazy występujące w długich słowach kluczowych, ale rzadko występujące samotnie. W przypadku $(freq(w))^2$ preferowane są zarówno frazy, jak i pojedyncze słowa.

Metody uczenia maszynowego stosowane do ewaluacji kandydatów sprowadzają się do zbudowania odpowiednich klasyfikatorów. W PKE wykorzystano naiwny klasyfikator bayesowski zbudowany na poniżej opisanych atrybutach, które mają wpływ na przydatność potencjalnego słowa kluczowego[148]:

- częstość słowa – zastosowano $(freq(w))^2$;
- pozycja słowa w zdaniu – słowa występujące na początku i na końcu zdania są przeważnie częściej kluczowe niż pozostałe wyrazy w zdaniu;
- pozycja słowa w tekście – słowa występujące w tekście wcześniej są przeważnie ważniejsze od tych występujących później;
- długość frazy – najczęściej słowa kluczowe mają jedno słowo składowe albo dwa takie słowa.

Pozycje słowa zapisano jako liczbę rzeczywistą z przedziału [0,1]; do jej wyznaczenia użyto wzoru:

$$pozycja = \frac{pozycja\ pierwszego\ wystąpienia\ badanej\ frazy}{liczba\ znalezionych\ potencjalnych\ słów\ kluczowych}$$

Badana jest przynależność do dwóch klas: „jest słowem kluczowym" oraz „nie jest słowem kluczowym".

IV. Algorytm identyfikacji autorów

System wspomagania wyboru recenzentów wykorzystuje algorytm grupowania hierarchicznego do identyfikacji autorów publikacji pobieranych przez robota internetowego. W wyniku działania robota dane są sprowadzane do jednolitego formatu i gromadzone w lokalnej bazie. Dane zapisane bezpośrednio przez robota nie mogą jednak zostać użyte do tworzenia rankingu recenzentów – konieczne jest ich przetworzenie do postaci wiedzy użytecznej dla systemu. Aby możliwy był wybór recenzentów dla wprowadzonego przez użytkownika wniosku, publikacje naukowe powinny zostać zaklasyfikowane do odpowiadających im dziedzin naukowych, natomiast autorzy publikacji zidentyfikowani jako konkretne osoby znajdujące się w BWNP. Pożądane jest również wyodrębnienie z danych słów kluczowych, które posłużą do porównywania wniosków z publikacjami znajdującymi się w bazie danych. Wszystkie te zadania wykonywane są jako autonomiczne procesy i służą wydobyciu z nieprzetworzonych danych wiedzy możliwej do wykorzystania przez system. Ten dodatek skupi się na problemie identyfikacji autorów publikacji. Rozwiązanie tego problemu okazało się niezbędne do prawidłowego korzystania z danych pobieranych przez robota, dlatego też opracowany został algorytm identyfikacji bazujący na metodzie grupowania hierarchicznego – Hierarchical Agglomerative Clustering[149]. Niżej przedstawiony zostanie problem niejednoznaczności nazwisk autorów publikacji, a dalej – teoria związana z algorytmem grupowania hierarchicznego.

1. Problem identyfikacji autorów

Dane na temat publikacji agregowane przez system pochodzą z różnych źródeł, z których każde zawiera informacje zapisane w specyficznym dla siebie formacie. O ile sprowadzenie prostych atrybutów do jednolitego formatu i ich porównywanie nie stanowi większego problemu, o tyle porównania obiektów pochodzących z różnych baz danych (w systemie wspomagania wyboru recenzentów będą to między innymi publikacje, autorzy i instytucje naukowe) jest zagadnieniem złożonym i wymagającym opracowania specyficznych algorytmów ujednolicających te dane.

System będzie korzystał przede wszystkim z wiedzy o pracownikach naukowych, kluczowym problemem staje się więc ich identyfikacja. Może się zdarzyć, że jedna i ta sama osoba figuruje w kilku bazach danych, z których każda zawiera podzbiór jej publikacji. Zadanie systemu to zatem rozpoznanie tej osoby i przypisanie do

[148] Pianta E., Tonelli S., *Association for Computational Linguistics*, in: *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala University, 2010.
[149] Manning Ch., Schutze H., Prabhakar R., op.cit.

niej wszystkich publikacji, których rzeczywiście jest twórcą. W momencie ekstrakcji danych wiedza na temat osób jest bardzo ograniczona, ponieważ zewnętrzne interfejsy często nie rozdzielają ludzi o takim samym imieniu i nazwisku oraz nie udostępniają publicznie unikatowych identyfikatorów przypisanych poszczególnym autorom. Zadanie staje się tym trudniejsze, im więcej osób o danym imieniu i nazwisku należy rozdzielić.

Sam format zapisu autora może różnić się w zależności od źródła danych. Większość źródeł nie podaje pełnego imienia autora przy jego publikacji. Podawany jest zazwyczaj inicjał imienia oraz pełne nazwisko. Niektóre źródła podają pełne nazwisko oraz pierwsze imię, wraz z inicjałami imion kolejnych. Na przykład, ta sama osoba *Jan Henryk Kowalski* może być zapisana w następujący sposób:

Kowalski, J.
Kowalski, J.H.
Jan H. Kowalski
Jan Henryk Kowalski
Kowalski, Jan H.
Kowalski, Jan
dr J.H. Kowalski
...

Oczywiście nazwisko i inicjał imienia nie wystarczają do zidentyfikowania określonego twórcy publikacji. Konieczne jest porównywanie w kontekście innych obiektów z nim powiązanych – publikacji, słów kluczowych, instytucji, współautorów. W rzeczywistości problem identyfikacji autorów można sprowadzić do problemu grupowania publikacji. Mając *n* publikacji zawierających autora o określonym nazwisku, chcemy podzielić je na *k* grup tak, aby wszystkie publikacje w pojedynczej grupie należały do jednego i tego samego autora, a jednocześnie żadna publikacja tego autora nie znalazła się w innej grupie. Algorytm identyfikacji autorów publikacji opracowany w ramach projektu zasadza się na takim właśnie podejściu. Do stworzenia algorytmu identyfikacji wykorzystano metodę grupowania hierarchicznego, opisaną niżej.

2. Hierarchical Agglomerative Clustering

HAC to jedna z metod grupowania hierarchicznego, które polegają na tworzeniu skupień na podstawie łączenia lub dzielenia obiektów i ich grup. Cały proces grupowania hierarchicznego można przedstawić w postaci drzewa lub dendrogramu. Istnieją dwa typy metod hierarchicznych[150, 151]:

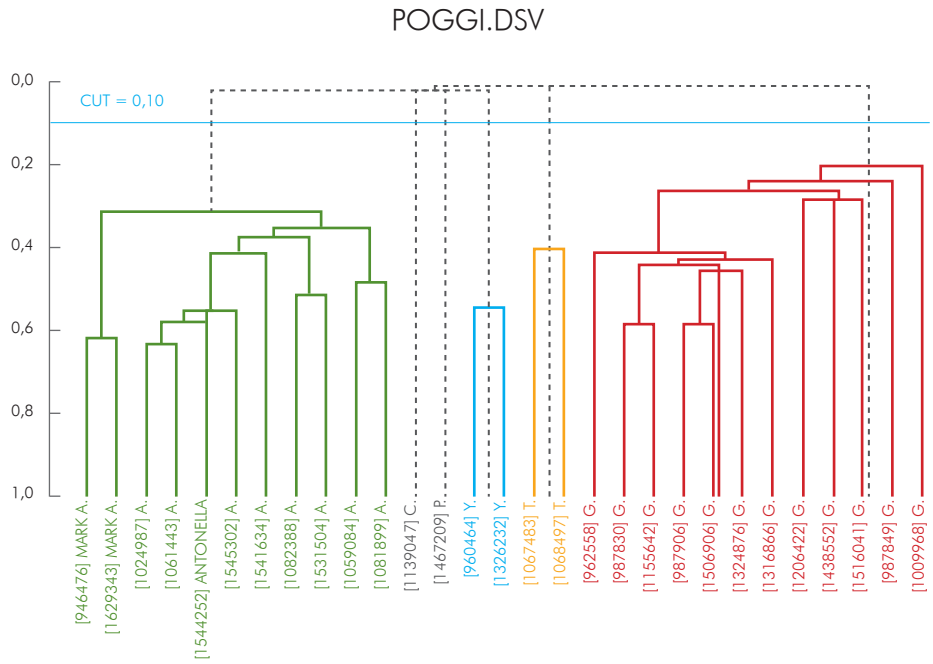
- **metody aglomeracyjne** (*bottom-up*) – na początku wszystkie obiekty traktuje się jak oddzielne skupienia; w kolejnych iteracjach algorytmu skupienia łączone są ze sobą do momentu powstania pojedynczego skupienia grupującego wszystkie obiekty;
- **metody podziałowe** (*top-down*) – najpierw istnieje tylko jedno skupienie złożone ze wszystkich obiektów; w kolejnych iteracjach dokonuje się podziału tego skupienia do momentu, aż wszystkie otrzymane skupienia będą składać się z pojedynczych obiektów.

W przeciwieństwie do innych popularnych metod grupowania, metody hierarchiczne nie wymagają określania liczby grup, które powinny zostać utworzone. Umożliwia to ich wykorzystywanie do rozwiązywania problemów grupowania tam, gdzie ta liczba nie jest znana. Są to metody iteracyjne, gdzie podczas każdej z iteracji wykonywany jest pojedynczy podział lub połączenie. Algorytm najczęściej kończy się wtedy, kiedy nie można wykonać już żadnej akcji (tzn. powstało jedno skupienie lub wszystkie skupienia są reprezentowane przez pojedyncze obiekty). Możliwe jest jednak określenie dodatkowego warunku stopu, który pozwoliłby na zakończenie algorytmu wcześniej, gdy określone przez użytkownika założenia zostały spełnione. Metody grupowania hierarchicznego dają szanse określenia liczby skupień już po zakończeniu działania algorytmu, na podstawie utworzonego podczas tego procesu dendrogramu. Dendrogram to struktura drzewiasta reprezentująca podziały lub połączenia następujące na kolejnych etapach procesu[152]. Z graficznej reprezentacji dendrogramu można odczytać, które skupienia i na jakim poziomie podobieństwa lub odmienności zostały ze sobą połączone.

[150] T. Hastie, R. Tibshirani, J. Friedman, op.cit.
[151] Manning Ch., Schutze H., Prabhakar R., op.cit.
[152] Ibidem.

Dla uproszczenia przyjmiemy, że opisywana w tym rozdziale metoda opiera się na podobieństwie między obiektami. Rysunek 47 przedstawia przykładowy dendrogram opisujący grupowanie pewnej liczby dokumentów. Pionowa oś informuje o miarach podobieństwa między łączonymi skupieniami. Z dendrogramu odczytać można, że do poziomu podobieństwa wynoszącego około 0,3 wszystkie obiekty zaznaczone kolorem zielonym zostały połączone w jedną grupę. Na osi poziomej znajdują się kolejne obiekty. Graficzna reprezentacja dendrogramu może być czytelna tylko wtedy, gdy obiekty podobne znajdują się blisko siebie. W przeciwnym razie łączące je linie przecinałyby się, co utrudniłoby identyfikację grup.

Rysunek 47. Przykładowy dendrogram



Źródło: opracowanie własne autorów

Dendrogram ułatwia śledzenie całego procesu tworzenia skupień w algorytmie grupowania. W większości zastosowań nie jest jednak informacją wystarczającą. Istotne jest również określenie preferowanej dla zadanego zbioru obiektów liczby skupień. Znalezienie takiego podziału polega na odcięciu części dendrogramu znajdującej się poniżej pewnego poziomu podobieństwa. Odcięcie jest równoznaczne z usunięciem wszystkich połączeń między skupieniami, które dokonały się na poziomach podobieństwa niższych niż określona przez odcięcie wartość. Aby określić poziom odcięcia, można posłużyć się jednym z poniższych kryteriów[153]:

- odcięcie na predefiniowanym poziomie; na przykładowym dendrogramie odcięto wszystkie skupienia utworzone dla poziomu podobieństwa 0,4 lub mniejszego;
- odcięcie dendrogramu w miejscu, gdzie różnica między miarami podobieństwa dla dwóch kolejnych połączeń jest największa;
- zdefiniowanie konkretnej liczby skupień k; dendrogram jest odcinany na poziomie, na którym istnieje dokładnie k skupień.

Metody grupowania hierarchicznego mogą korzystać z różnych miar podobieństwa obiektów lub odległości (odmienności) między obiektami. Algorytm grupowania hierarchicznego nie wymaga wprowadzania wektorów cech obiektów jako danych wejściowych, wystarczy macierz podobieństwa lub odmienności między tymi obiektami. Jest to kolejna z zalet tej metody, pozwala bowiem na grupowanie obserwacji o wektorach nienależących do przestrzeni euklidesowej, a także obserwacji o wektorach mieszanych, zawierających cechy numeryczne, nominalne i binarne. W razie podobieństwa, jako pierwsze łączone są ze sobą te skupienia,

[153] Ibidem.

dla których wartość podobieństwa jest największa. W przypadku odmienności łączone są te grupy, których odmienność jest najmniejsza. Algorytm *agglomerative hierarchical clustering* w ogólnej postaci składa się z następujących kroków[154]:

1. dla obiektów $d1, d2 \ldots d_N$ utwórz N skupień $\omega_1, \omega_2 \ldots \omega_N$, z których każde zawiera pojedynczy obiekt;
2. określ wartości podobieństwa dla wszystkich par skupień na podstawie funkcji podobieństwa między skupieniami $SIM(\omega_i, \omega_j)$;
3. wybierz parę skupień, dla których wartość podobieństwa jest największa i połącz je ze sobą;
4. oblicz na nowo podobieństwa między nowo utworzonym skupieniem a wszystkimi pozostałymi skupieniami;
5. jeżeli istnieje więcej niż jedno skupienie, wróć do punktu 3.

W zależności od użytej funkcji *SIM*, metoda grupowania aglomeracyjnego może dawać różne wyniki. Najpopularniejszymi funkcjami wykorzystywanymi do mierzenia podobieństwa między skupieniami są[155]:

- **miara pojedynczego połączenia** (*single link*) – podobieństwo między dwoma skupieniami jest liczone jako podobieństwo między dwoma najbardziej podobnymi do siebie obiektami;
- **miara całkowitego połączenia** (*complete link*) – podobieństwo między dwoma skupieniami jest liczone jako podobieństwo między dwoma najmniej podobnymi do siebie obiektami z obu skupień;
- **miara średniego podobieństwa** (*group-average*) – podobieństwo między dwoma skupieniami jest określane jako średnie podobieństwo między wszystkimi parami obiektów, włączając w to również obiekty należące do tych samych skupień:

$$SIM\ -GA\ (\omega_i, \omega_j) = \frac{1}{(N_i + N_j)(N_i + N_j - 1)} \sum_{d_m \in \omega_i \cup \omega_j} \sum_{d_n \in \omega_i \cup \omega_j, d_n \neq d_m} sim(d_m, d_n)$$

gdzie:
N_i, N_j – liczby obiektów w skupieniach ω_i, ω_j;
$sim(d_m, d_n)$ – funkcja mierząca podobieństwo między dwoma obiektami;

- **miara centroidów** – podobieństwo między skupieniami jest liczone jako podobieństwo między środkami (centroidami) skupień:

$$SIM\ -CENT\ (\omega_i, \omega_j) = \vec{\mu}(\omega_i) \cdot \vec{\mu}(\omega_j) =$$
$$= \left(\frac{1}{N_i} \sum_{d_m \in \omega_i} \vec{d}_m \right) \cdot \left(\frac{1}{N_j} \sum_{d_n \in \omega_j} \vec{d}_n \right) = \frac{1}{N_i N_j} \sum_{d_m \in \omega_i} \sum_{d_n \in \omega_j} \vec{d}_m \cdot \vec{d}_n$$

Proste implementacje grupowania aglomeracyjnego wykorzystują macierze do zapisywania odległości między skupieniami. Ze względu na konieczność przeszukiwania macierzy w każdej iteracji algorytmu, aby znaleźć najlepszą wartość podobieństwa (odległości), złożoność obliczeniowa takich rozwiązań jest rzędu $O(n^3)$. Implementacje wykorzystujące kolejki priorytetowe do zapisywania odległości pozwalają na zmniejszenie tej złożoności do $O(n^2)$ przy wykorzystaniu miary *single-link* oraz $O(n^2 \log n)$ dla pozostałych miar[156].

3. Odmienność między obiektami

Opisane miary podobieństwa między skupieniami zakładają, że dla dowolnych dwóch obiektów możliwe jest policzenie odległości lub podobieństwa między nimi. Funkcje *SIM* korzystają z wartości podobieństw poszczególnych par obiektów. Obiekty traktujemy jako wektory cech. Cechy mogą być reprezentowane przez różne typy danych: numeryczne, nominalne lub binarne. W zależności od typu danych występujących w wektorze, konieczne jest zastosowanie metody liczenia odmienności specyficznej dla określonego typu. Dla wektorów składających się z danych mieszanych stosuje się metodę porównywania poszczególnych cech obiektów osobno, a następnie oblicza się podobieństwo lub odmienność uśrednioną.

[154] Ibidem.
[155] Manning Ch., Schutze H., Prabhakar R., op.cit.
[156] Ibidem.

3.1. Cechy numeryczne

Dla wektorów składających się tylko z danych numerycznych najczęściej używanymi miarami odległości jest odległość euklidesowa oraz Manhattan (*city block distance*). Obie miary są szczególnymi przypadkami bardziej ogólnej funkcji zwanej odległością Minkowskiego.

Odległość Minkowskiego między wektorami x_i, x_j jest definiowana jako[157]:

$$dist(x_i, x_j) = \left(\left| x_{i1} - x_{j1} \right|^h + \left| x_{i2} - x_{j2} \right|^h + \ldots + \left| x_{ir} - x_{jr} \right|^h \right)^{\frac{1}{h}}$$

Jeśli $h=2$, jest to odległość euklidesowa[158]:

$$dist(x_i, x_j) = \sqrt{\left(x_{i1} - x_{j1}\right)^2 + \left(x_{i2} - x_{j2}\right)^2 + \ldots + \left(x_{ir} - x_{jr}\right)^2}$$

Jeśli $h=1$, jest to odległość Manhattan[159]:

$$dist(x_i, x_j) = \left| x_{i1} - x_{j1} \right| + \left| x_{i2} - x_{j2} \right| + \ldots + \left| x_{ir} - x_{jr} \right|$$

3.2. Cechy binarne i nominalne

Powyższe wzory stosowane dla cech numerycznych nie mają zastosowania, gdy wektor składa się z wartości binarnych lub nominalnych. Ponieważ wartości zarówno dla wektorów binarnych, jak i nominalnych nie mają określonego porządku, nie jest możliwe określenie liczbowej miary odległości między dowolną cechą dla wybranej pary obiektów. Atrybut binarny przyjmuje jedną z dwóch możliwych wartości, często zapisywanych w postaci 0 i 1; atrybuty nominalne natomiast określają jedną z kategorii mających postać nazw lub identyfikatorów liczbowych. W obu przypadkach porównania mogą odbywać się tylko na zasadzie równości lub nierówności wartości atrybutu. Najczęściej wykorzystywane miary odmienności między wektorami wykorzystują macierz określającą liczbę zgodnych oraz niezgodnych wartości atrybutów w wektorach. Dla dwóch wektorów binarnych x_i, x_j macierz ta wygląda następująco[160]:

	1	**0**
1	a	b
0	c	d

gdzie:
a – liczba atrybutów przyjmujących wartość 1 dla obu wektorów;
b – liczba atrybutów, dla których $x_{if} = 1$ i $x_{jf} = 0$ dla atrybutu o pozycji f w wektorze x;
c – liczba atrybutów, dla których $x_{if} = 0$ i $x_{jf} = 1$ dla atrybutu o pozycji f w wektorze x;
d – liczba atrybutów przyjmujących wartość 0 dla obu wektorów.

W przypadku cech binarnych dokonuje się podziału na cechy o wartościach symetrycznych oraz cechy o wartościach asymetrycznych. Dla pierwszych zakłada się, że wartości 1 i 0 są tak samo istotne dla obiektu, przekazują taką samą ilość informacji. Cechy asymetryczne natomiast dotyczą przypadków, gdy jedna z możliwych wartości jest istotniejsza od drugiej (np. wartość 1 oznaczająca występowanie pewnego objawu u pacjenta niesie ze sobą ważniejszą informację niż wartość 0 mówiąca, że dany objaw nie występuje). Przy

[157] Ibidem.
[158] Liu B., op.cit.
[159] Ibidem.
[160] Ibidem.

założeniu, że wszystkie atrybuty w wektorach x są binarne i symetryczne, do obliczenia odległości stosuje się wzór[161]:

$$dist(x_i, x_j) = \frac{b + c}{a + b + c + d}$$

Dla wektorów binarnych asymetrycznych ignorowane są te cechy, dla których oba porównywane wektory przyjmują wartość 0:

$$dist(x_i, x_j) = \frac{b + c}{a + b + c}$$

W podobny sposób liczy się odległość dla atrybutów nominalnych. Jeżeli r jest liczbą wszystkich atrybutów w wektorze, a q jest liczbą zgodnych wartości atrybutów dla dwóch wektorów x_i, x_j to odległość między tymi wektorami wynosi[162]:

$$dist(x_i, x_j) = \frac{r - q}{r}$$

3.3. Wektory mieszane

W rzeczywistości rzadko zdarza się, aby wektory cech składały się jedynie z danych jednego typu. Najczęściej mamy do czynienia z wektorami mieszanymi, w których mogą występować dane binarne, nominalne i numeryczne. Jednym ze sposobów radzenia sobie z takim problemem jest sprowadzenie wszystkich cech do jednolitej postaci. Jeżeli jeden z typów danych występuje w wektorze częściej niż pozostałe, mogą one zostać przekonwertowane do tego typu. Innym sposobem na obliczenie odległości dla wektorów mieszanych jest policzenie odległości dla każdej cechy osobno, zgodnie z odpowiednim dla typu cechy wzorem, a następnie połączenie wszystkich wyników w pojedynczą miarę odległości. Jednym ze sposobów obliczenia takiej miary jest metoda Gowera[163, 164]:

$$dist(x_i, x_j) = \frac{\sum_{f=1}^{r} \delta_{ij}^f d_{ij}^f}{\sum_{f=1}^{r} \delta_{ij}^f}$$

Funkcja ta zwraca miarę odległości w postaci liczby z przedziału [0, 1], gdzie r jest liczbą cech w wektorze. Wartość δ_{ij}^f jest równa 1, jeżeli w obu porównywanych wektorach cecha występuje (nie ma wartości pustej, brakującej) lub 0 – w przeciwnym wypadku. Wartość d_{ij}^f z zakresu [0, 1] jest odległością wyliczoną dla cechy f. W zależności od typu cechy, wykorzystywane są różne wzory na wyliczenie tej wartości. Dla cech binarnych lub nominalnych otrzymujemy:

$$d_{ij}^f = \begin{cases} 1 & dla \quad x_{if} = x_{jf} \\ 0 & dla \quad x_{if} \neq x_{jf} \end{cases}$$

[161] Ibidem.
[162] Ibidem.
[163] Hastie T., Tibshirani R., Friedman J.H., op.cit.
[164] Liu B., op.cit.

W przypadku cech numerycznych stosowany jest wzór[165]:

$$d_{ij}^{f} = \frac{|x_{if} - x_{jf}|}{R_f}$$

gdzie:
R_f – miara rozstępu cechy, tzn.[166]:

$$R_f = max(f) - min(f)$$

V. Miary

W dziedzinie Information Retrieval opracowano zestaw miar jakości, pozwalających na ocenę wyniku wyszukiwania dokumentów. Trzy najpopularniejsze i najszerzej stosowane miary oceny to precyzja, kompletność i F-miara.

Precyzję *(precision)* definiujemy jako procent wyszukanych dokumentów, które są relewantne z punktu widzenia zapytania. Istnieje szeroki zakres interpretacji tego, co rozumiemy pod pojęciem dokument, mogą to być to zarówno instancje dokumentów, jak i etykiety, którymi są one opisywane:

$$precyzja = \frac{|\{relewantne\ dokumenty\} \cap \{wyszukane\ dokumenty\}|}{|\{wyszukane\ dokumenty\}|}$$

Kompletność *(recall)* definiujemy jako procent relewantnych dokumentów, które zostały wyszukane. Do obliczenia miary kompletności wymagana jest znajomość całego zbioru poprawnych odpowiedzi:

$$kompletność = \frac{|\{relewantne\ dokumenty\} \cap \{wyszukane\ dokumenty\}|}{|\{relewantne\ dokumenty\}|}$$

W celu uwzględnienia specyficznych właściwości obu powyższych miar w ramach jednego wskaźnika wprowadzono dodatkową miarę – **F-miarę** (średnia ważona precyzji i kompletności):

$$F = \frac{2 * precyzja * kompletność}{precyzja + kompletność}$$

W zadaniu klasyfikacji tekstów pojęcia *true positives* (TP), *true negatives* (FN), *false positives* (TN), *false negatives* (FP) porównują wyniki działania klasyfikatora:

- TP (*true positives*) – liczba poprawnie sklasyfikowanych przykładów z wybranej klasy;
- FN (*false negatives*) – liczba błędnie sklasyfikowanych przykładów z tej klasy, tj. decyzja negatywna, podczas gdy przykład w rzeczywistości jest pozytywny;
- TN (*true negatives*) – liczba przykładów poprawnie nieprzydzielonych do wybranej klasy;
- FP (*false positives*) – liczba przykładów błędnie przydzielonych do wybranej klasy, podczas gdy w rzeczywistości do niej nie należą.

[165] Ibidem.
[166] Ibidem.

W kontekście klasyfikacji dokumentów w oparciu o powyższe cząstkowe miary stosowana jest ogólna miara o nazwie **dokładność** *(accuracy)*, która określa procent poprawnie skategoryzowanych dokumentów:

$$dokładność = \frac{TP + TN}{TP + TN + FP + FN}$$

W dziedzinie klasyfikatorów wieloetykietowych stosuje się też miarę **dopasowania** *(fit)*, która ma za zadanie określić procent dokumentów, dla których została dopasowana dokładna liczba poprawnych kategorii[167].

W zadaniach przetwarzania tekstów istotną kwestią jest korekta ciągów znaków. Do rozwiązywaniu tego rodzaju kwestii stosuje się **odległość Levenshteina.** Jest to miara odmienności napisów (skończonych ciągów znaków) stworzona przez Vladmira Levenshteina. Powyższa metryka określa odległość pomiędzy dwoma napisami jako najmniejszą liczbę działań prostych potrzebnych do transformacji jednego napisu w drugi. Działaniem prostym jest: wstawienie znaku, usunięcie znaku, zamiana znaku[168].

W zadaniach oceny wagi i znaczenia naukowców wprowadzono w 2005 roku **indeks Hirscha.** Jest to współczynnik określający wagę prac naukowych danego autora, opisujący cały jego dorobek, a nie tylko znaczenie jednej poszczególnej pracy. Naukowiec ma indeks o wartości h, jeśli h z jego N publikacji posiada co najmniej h cytowań każda i pozostałe publikacje (w liczbie N – h) nie mają więcej niż h cytowań każda[169]. Indeks Hirscha rozwiązuje główne problemy wskaźników bibliometrycznych, takie jak całkowita liczba publikacji danej osoby lub całkowita liczba cytowań danej osoby. Całkowita liczba publikacji nie bierze pod uwagę jakości prac naukowych, podczas gdy całkowita liczba cytowań nie uwzględnia rozkładu cytowań. Oznacza to, że jedna publikacja z duża ilością odwołań jest tak samo ważna jak wiele publikacji z małą liczbą cytowań. Indeks Hirscha pozwala równocześnie mierzyć jakość i ilość dorobku naukowego.

[167] Manning Ch., Schutze H., Prabhakar R., op.cit.
[168] Levenshtein V., *Binary codes capable of correcting deletions, insertions, and reversals*, „Soviet Physics – Doklady", 10, 707–710, 1966.
[169] Hirsch J., *An index to quantify an individual's scientific research output*, „Proceedings of National Academy of Science of USA", 2005.

VI. Bibliografia

Almpanidis G., Kotropoulos C., Pitas I., *Combining text and link analysis for focused crawling – An application for vertical search engines*, „Information Systems", 32(6), 886–908, 2007.

Andrade M., Valencia A., *Automatic extraction of keywords from scientific text: Application to the knowledge domain of protein families*, „BioInformatics", 14(7), 600–607, 1998.

Charniak E., *Statistical techniques for natural language parsing*, „AI Magazine", 18(4), 33–43, 1997.

Cimiano P., *Ontology Learning and Population from Text*, Springer, Berlin – Heidelberg 2010.

Daille B., Gaussier E., Lange J., *Towards Automatic Extraction of Monolingual and Bilingual Terminology*, in: *Proceedings of COLING 94*, COLING, Kyoto 1994.

Devine J., Egger-Sider F., *Beyond Google: The invisible web in the academic library*. „The Journal of Academic Librarianship", 30(4), 265–269, 2004.

Diligenti M., Coetzee F., Lawrence S., Giles C.L., Gori M., *Focused Crawling Using Context Graphs*, in: El Abbadi A. et al., eds., *Proceedings of the 26th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., San Francisco 2000.

Duda C., Frey G., Kossmann D., Zhou Ch., *AJAX Search: Crawling, indexing and searching web 2.0 applications*, „Proceeding VLDB Endowment", 1, 1440–1443, 2008.

Fragoudis D., Meretakis D., Likothanassis S., *Best terms: An efficient feature-selection algorithm for text categorization*, „Knowledge and Information Systems", 8(1), 16–33, 2005.

Frantzi K., Ananiadou S., Mima H., *Automatic recognition of multiword terms*, „International Journal of Digital Libraries", 3(2), 117–132, 2000.

Gawrysiak P., Rybiński H., Protaziuk G., *Text Onto Miner – A Semi Automated Ontology Building System*, in: *Proceedings of 17th International Symposium on Intelligent Systems*, 563–573, Springer, Berlin – Heidelberg 2008.

Hastie T., Tibshirani R., Friedman J., *The Elements of Statistical Learning*, Springer, New York 2009.

Hirsch J., *An index to quantify an individual's scientific research output*, in: „Proceedings of National Academy of Science of USA", 2005.

Hoare Z., *Landscapes of Naive Bayes classifiers*, „Pattern Analysis and Application", 11(1), 59–72, 2008.

Hsu C.C., Wu F., *Topic-specific crawling on the web with the measurements of the relevancy context graph*, „Information Systems", 31(4), 232–246, 2006.

Huang C.L., Wang C.J., *A GA-based feature selection and parameters optimization for support vector machines*, „Expert Systems With Applications", 31(2), 231–240, 2006.

Hulth A., *Improved Automatic Keyword Extraction Given More Linguistic Knowledge*, in: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, 216–223, ACL, Stroudsburg 2003.

Joachims T., *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, in: Nedellec C., Rouveirol C., *Proceedings of the 10th European Conference on Machine Learning*, Springer, London 1998.

Jones K., *A statistical interpretation of term specificity and its application in retrieval*, „Journal of Documentation", 28(1), 11–21, 1972.

Juan A., Ney H., *Reversing and Smoothing the Multinomial Naive Bayes Text Classifier*, in: Ińesta Quereda J.M., Micó L., eds., *Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems*, ICEIS Press, Ciudad Real 2002.

Justeson J., Katz S., *Technical terminology: Some linguistic properties and an algorithm for identification in text*, „Natural Language Engineering", 1, 9–27, 1995.

Kibriya A.M., Frank E., Pfahringer B., Holmes G., *Multinomial Naive Bayes for Text Categorization Revisited*, in: Webb G.I., Yu X., Eds., *Proceedings of 17th Australian Joint Conference on Artificial Intelligence*, Springer, Berlin 2004.

Konchady M., *Text Mining Application Programming*, Charles River Media, Rockland 2006.

Kupiec J., Pedersen J., Chen F., *A Trainable Document Summarizer*, in: Fox E., Ingwersen P., Fidel R., *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Seattle 1995.
Levenshtein V., *Binary codes capable of correcting deletions, insertions, and reversals*, „Soviet Physics – Doklady", 10, 707–710, 1966.
Liu B., *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*, Springer, New York 2010.
Luhn H., *The automatic creation of literature abstracts*, „IBM Journal of Research and Development", 156–165, 1958.
Manning C., Schutze H., Prabhakar R., *Introduction to Information Retrieval*, Cambridge University Press, 2008.
McCallum A., Nigam K., *A Comparison of Event Models for Naive Bayes Text Classification*, in: *AAAI/ICML-98 Workshop on Learning for Text Categorization*, AAAI Press, Palo Alto 1998.
Mihalcea R., Tarau P., *Text Rank: Bringing Order into Texts*, in: *Proceedings of Conference* on *Empirical Methods in Natural Language Processing (EMNLP 2004)*, ACL, Barcelona 2004.
Noble W., *What is a support vector machine?*, „Nature Biotechnology", 24, 1565–1567, 2006.
Pant G., Srinivasan P., Menczer F., *Crawling the Web*, in: Levene M., Poulovassilis A., eds., *Web Dynamics: Adapting to Change in Content, Size, Topology and Use*, Springer, Berlin – Heidelberg 2004.
Pianta E., Tonelli S., *Association for Computational Linguistics*, in: *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala Univeristy, 2010.
Rose S., Engel D., Cramer N., Cowley W., *Automatic Keyword Extraction from Individual Documents*, in: Berry M.W., Kogan J., *Text Mining: Applications and Theory*, 19–37, Wiley, New York 2010.
Rybiński H. i inni, *Text mining tools for ontology maintenance,* reports from phase 2–5, Wydział Elektroniki i Technik Informacyjnych Politechniki Warszawskiej, 2006–2007.
Salton G., McGill M.J., *Introduction to Modern Information Retrieval*, McGraw-Hill, New York 1983.
Sebastiani F., *Text Categorization*, in: Sirmakessis S., ed., *Text Mining and Its Applications*, Springer, Berlin – Heidelberg 2004.
Shestakov D., *Search Interfaces on the Web: Querying and Characterizing*, University of Turku, 2008.
Silva C., Ribeiro B., *On text-based mining with active learning and background knowledge using SVM*, „Journal of Soft Computing – A Fusion of Foundations, Methodologies and Applications", 11(6), 519–530, 2007.
Storn R., Price K., *Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces*, „Journal of Global Optimization", 11, 341–359, 1997.
Witten I.H., Paynter G.W., Frank E., Gutwin C., Nevill-Manning C.G., *Automatic Keyphrase Extraction*, working paper 00/5, Department of Computer Science, The University of Waikato, 2000.
Yang Y., Liu X., *A Re-Examination of Text Categorization Methods*, in: *Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York 1999.

Źródła internetowe:

Keyphrase Extraction Algorithm, http://www.nzdl.org/Kea/description.html, dostęp 14.08.2012.
The Robot Exclusion Protocol, http://www.robotstxt.org/robotstxt.html, dostęp 13.08.2012.
Wikipedia, *Support vector machine*, http://en.wikipedia.org/wiki/Support_vector_machine, dostęp 13.08.2012.

SPIS RYSUNKÓW

SPIS TABEL

WYKAZ SKRÓTÓW I AKRONIMÓW

BWNP	Bazy Wiedzy o Nauce Polskiej
CKSST	Centralna Komisja do spraw Stopni i Tytułów
CRF	Conditional Random Fields
CSS	Cascading Style Sheets
DE	Differential Evolution
DOI	Digital Object Identifier
DOM	Document Object Model
EES	Elsevier Editorial System
ERC	European Research Council
FIFO	First In, First Out
GNU	GNU's Not Unix
GNU GPL	NU General Public License
HAC	Hierarchical Agglomerative Clustering
HTML	Hyper Text Markup Language
HTTPS	HyperText Transfer Protocol Secure
ID	Identity document
ISSN	International Standard Serial Number
JMX	Java Management Extensions
JPA	Java Persistence API
JSP	Java Server Pages
KBN	Komitet Badań Naukowych
KEA	Keyphrases Extraction Algorithm
LSI	Lokalny System Informatyczny
MNB	Multinomial Naive Bayes
MNiSW	Ministerstwo Nauki i Szkolnictwa Wyższego
NB	Naive Bayes
NCBiR	Narodowe Centrum Badań i Rozwoju
NCN	Narodowe Centrum Nauki
NIH	National Institutes of Health

NSF	National Science Foundation
OAI	Open Archives Initiative
OECD	Organization for Economic Co-operation and Development
OPI	Ośrodek Przetwarzania Informacji – Instytut Badawczy
OSF	Obsługa Strumieni Finansowania
OSJ	Ontology of Scientific Journals
PKE	Polish Keyword Extractor
PO IG	Program Operacyjny Innowacyjna Gospodarka
PN FBN	Polsko-Norweski Fundusz Badań Naukowych
PSPB	Polsko-Szwajcarski Program Badawczy
RAKE	Rapid Automatic Keyword Extraction
REST	Representational State Transfer
SDD	Słownik Dziedzin i Dyscyplin
SFB	Sonderforschungsbereiche
SMO	Sequential Minimal Optimization
SQL	Structured Query Language
SSL	Secure Socket Layer
SVM	Support Vector Machine
TF-IDF	Term frequency – inverse document frequency
UML	Unified Modeling Language
URL	Uniform Resource Locator
WAR	Web Application Archive
XML	Extensible Markup Language