AUTOMATYKA • 2010 • Tom 14 • Zeszyt 3/2

Wojciech Chmiel*, Piotr Kadłuczka*, Stanisław Jędrusik**

Nadzorowana kategoryzacja tekstów angielskojęzycznych

1. Wstęp

Klasyfikacja tekstu (TC – *text classification*) jest zagadnieniem polegającym na automatycznym podziale dokumentów na kategorie. W obecnym czasie, duża liczba praktycznych zastosowań tego zagadnienia wiąże się np. z sortowaniem tekstów naukowych, technicznych, medycznych, patentowych, wypełnianiem hierarchicznych katalogów sieciowych, selektywnym udostępniałem dokumentów, filtracją spamu, określaniem tematyki, a także gatunku książek. Jest ona szczególnie przydatna w przedsiębiorstwach, w których tworzy się i przetwarza dużą ich liczbę. TC pozwala uwolnić się od kosztownej, ręcznej organizacji baz danych oraz ograniczyć czas konieczny do przeprowadzenia klasyfikacji, jeśli jest on dla danego zadania krytyczny. Łączy ona techniki stosowane w dziedzinie uczenia maszynowego (ML – *machine learning*) oraz wyszukiwania i udostępniania informacji (IR – *information retrieval*). Obecnie metody stosowane w kategoryzacji tekstów mogą śmiało konkurować z systemami opartymi na pracy wykwalifikowanych ekspertów.

Nadzorowana kategoryzacja tekstu polega na automatycznym przydziale tekstu do zbioru predefiniowanych klas, zwanych często kategoriami lub tematami (*topics*). Ze względu na swoje znaczenie, zagadnienie to od wielu lat jest tematem licznych prac naukowych. Gromadzenie informacji interesującej dla konkretnego odbiorcy jest działaniem subiektywnym, ponieważ jedynie on może stwierdzić, czy dana informacja jest dla niego interesująca. Subiektywny, z punktu widzenia użytkownika, podział informacji jest źródłem dostarczającym wiedzę o jego preferencjach. Preferencje te wyrażają się podziałem gromadzonych przez niego dokumentów np. na katalogi. Taki podział jest naturalnym źródłem wiedzy dla systemów uczących się i klasyfikujących informacje. Stąd zagadnienia związane z wyszukiwaniem i udostępnianiem informacji są doskonałym i naturalnym polem zastosowania metod z dziedziny automatycznych systemów uczących się. W ostatnim okresie

* Katedra Automatyki, Wydział EAIiE, Akademia Górniczo-Hutnicza w Krakowie
** Katedra Informatyki Stosowanej, Wydział Zarządzania, Akademia Górniczo-Hutnicza w Krakowie

można zauważyć wzrost zainteresowania automatyczną kategoryzacją dokumentów, ze względu na powiększające się oczekiwania odbiorców wobec tego typu systemów oraz wzrost liczby przetwarzanych informacji.

2. Kategoryzacja tekstów

Zagadnienie kategoryzacji może być zdefiniowane następująco. Niech $\Phi : D \times C \to \{T, F\}$ będzie nieznaną funkcją celu, określająca sposób klasyfikacji przez eksperta. Funkcja ta jest aproksymowana za pomocą funkcji $\hat{\Phi} : D \times C \to \{T, F\}$ zwaną klasyfikatorem, gdzie $C = \{c_1, c_2, c_3, ..., c_k\}$ jest predefiniowanym zbiorem kategorii zawierającym k kategorii, natomiast D jest zbiorem dokumentów o nieznanym rozmiarze. Dla tak zdefiniowanego klasyfikatora, jeśli $\Phi(c_i, d_i) = T$, wtedy d_i jest określany jako pozytywny przykład (członek) klasy c_i, natomiast jeśli $\Phi(c_i, d_i) = F$, to d_i jest określany jako przykład negatywny klasy c_i. Klasy można traktować jak pewne etykiety określające przynależność danego dokumentu. Najczęściej podczas klasyfikacji zakłada się, że jedynie dostępna jest informacja, którą można uzyskać z dokumentu (inne informacje, opisujące klasyfikowany tekst są niedostępne). Należy sobie oczywiście zdawać sprawę o relatywizmie przydziału dokumentu do danej klasy. Dany dokument może należeć do wielu klas, zależnie od subiektywnego spojrzenia eksperta. Ze względu na różne typy klasyfikacji i różne rodzaje definicji klas, dokument może należeć do jednej lub kilku klas jednocześnie. Zagadnienie klasyfikacji, w którym istnieją predefiniowanie klasy, określa się mianem **klasyfikacji nadzorowanej**.

Ważnym kierunkiem prac badawczych jest problem **klasyfikacji nienadzorowanej**, gdzie sam system jest odpowiedzialny zarówno za klasyfikację, jak i za „odkrywanie" w zbiorze testowym nowych klas (najczęściej na podstawie model rozkładu losowego cech dokumentu).

W zagadnieniach klasyfikacji można również wyodrębnić klasyfikacje jednoetykietowe oraz wieloetykietowe. W pierwszym przypadku, klasyfikacja odnosi się do sytuacji, w której do każdego dokumentu $d_i \in D$ przypisana jest tylko jedna klasa $c_i \in C$. W drugim przypadku, każdemu dokumentowi przypisuje się n klas, przy czym $0 \leq n \leq k$. Odmianą klasyfikacji jednoetykietowej, jest tzw. klasyfikacja binarna, w której każdy dokument przypisuje się do zbioru $c_i \in C$ lub jego komplementarnego dopełnienia $\overline{C}$. Z punktu widzenia algorytmicznego, klasyfikacja binarna jest prostszym zagadnieniem niż klasyfikacja jednoetykietowa.

W większości algorytmów klasyfikujących, etap nauki lub klasyfikacji nowych dokumentów poprzedzany jest etapem indeksacji dokumentów. Etap ten polega na przypisaniu dokumentowi d_j reprezentacji, która jest bezpośrednio używana przez algorytm klasyfikujący. Najczęściej dokumentowi d_j odpowiada reprezentacja wektorowa w postaci wag terminów $\vec{d}_j = (t_{f1}, t_{f2}, ..., t_{fn})$, gdzie t_{fi} jest częstotliwością występowania i-tego terminu (słowa lub ogólnie cechy) w dokumencie d_j. Na tym etapie pod uwagę brane są jedyne terminy ze słownika Γ zawierającego n terminów. Terminy w słowniku są najczęściej ter-

minami występującymi w co najmniej l dokumentach ze zbioru uczącego. Ogólnie, przyjmuje się, że $1 \leq l \leq 5$. Zawartość słownika ma podstawowy wpływ na jakość klasyfikacji. Z tego powodu, w artykule zaproponowano nową metodę wyboru terminów do słownika na podstawie analizy zawartości zbioru uczącego.

Zadaniem automatycznej klasyfikacji dokumentów, rozważanym w pracy, jest automatyczne etykietowanie nieznanych dotąd dokumentów elektronicznych. Etykiety te mogą określać tematykę dokumentu. Nadzorowana klasyfikacja dokumentów, wymaga zdefiniowania rozważanych kategorii oraz przydzielenia do każdej z nich pewnej liczby dokumentów.

3. Drzewa decyzyjne

Drzewo decyzyjne (DT – *Decision Tree*) pozwala na reprezentację wiedzy na podstawie szeregu warunków zapisanych w postaci klauzul Horna. Aby uzyskać odpowiedź (predykcję) systemu opartego na DT, należy mu zadać określoną liczbę pytań. Toteż struktura drzewa, opisująca pewną wiedzę, wynika z możliwości udzielenia przeczącej lub potwierdzającej odpowiedzi na zadawane pytania. W przykładzie przedstawionym na rysunku 1, znajdujący się na samej górze korzeń drzewa, rozstrzyga kwestię, czy pewna cecha obiektu zwana X spełnia czy też nie określony warunek: $X < 1{,}678$. Jeśli warunek jest spełniony, to proces decyzyjny zostaje przeniesiony do lewej gałęzi drzewa, w przeciwnym przypadku podążamy prawą gałęzią drzewa. Idąc prawą gałęzią, natrafiamy na pytanie, czy pewna cecha obiektu, zwana Y jest większa od 1,11? Jeśli to fałsz to, następną kwestią do rozstrzygnięcia będzie, czy kolejna cecha obiektu, Z jest mniejsza od 0,8? Jeśli warunek nie jest spełniony to docieramy do liścia, uzyskując odpowiedź (predykcję) co do rodzaju obiektu. W tym przypadku jest to C.

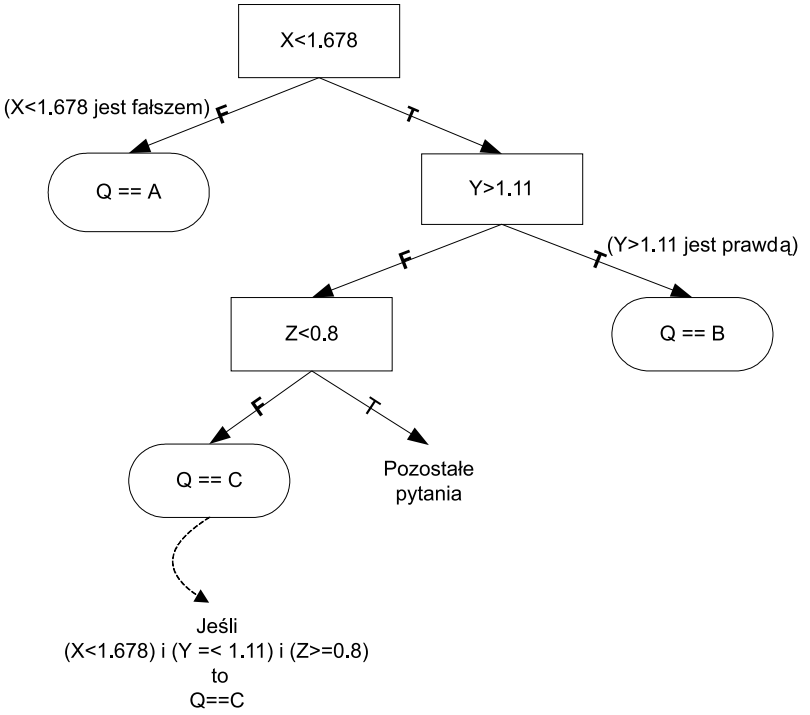
W celach badawczych, zaimplementowano wersję drzewa decyzyjnego zaproponowaną przez Rossa Quinlana, noszącą nazwę C4.5 [10, 11]. Algorytm działa w podobny sposób jak większość systemów uczących się opartych na empirii. Budowa drzewa decyzyjnego w C4.5, bazuje na szeroko stosowanym algorytmie *TDIDT* (*Top Down Iterative Decision Tree*). Na początku korzeń zawiera wszystkie próbki treningowe, których podział zachodzi na kolejnych poziomach drzewa. Inaczej mówiąc, systemy te rozważają pewien zbiór sklasyfikowanych przypadków, opisanych wektorem atrybutów, konstruując mapowanie z przestrzeni atrybutów (dyskretnych i ciągłych) do przestrzeni klas. Algorytm C4.5 zaproponowany przez Quinlana podczas tworzenia drzewa decyzyjnego stosuje zasadę „dziel i rządź" [4].

Algorytm działa rekurencyjnie dla każdego węzła drzewa. W każdym przypadku określamy, czy węzeł będzie liściem czy węzłem rozgałęziającym.

– Węzeł drzewa będzie **liściem**, jeśli aktualnie rozważany zbiór sklasyfikowanych zdarzeń D spełnia warunek stopu. W takim przypadku liść będzie skojarzony z np. klasą posiadającą najbardziej liczną reprezentację w D. Spełnienie kryterium stopu oznacza zakończenie wywołań rekurencyjnych.

Kryterium stopu może zadziałać, gdy:

a) wszystkie rozważane przypadki należą do jednej klasy – liść reprezentuje tą klasę,
b) nie ma już dostępnych atrybutów – liść reprezentuje klasę większościową,
c) gdy liczność $|D|$ jest mniejsza niż zakładany próg. Wtedy liść reprezentuje klasę większościową.



Rys. 1. Przykład drzewa decyzyjnego

– Węzeł drzewa będzie **węzłem rozgałęziającym** według kryterium wyboru atrybutu. Stosując test T, z wzajemnie się wykluczającymi wynikami T_1, T_2, T_3, ..., T_n, uzyskuje się podział D na podzbiory D_1, D_2, .., D_n, w których D_i zawiera przypadki uzyskane jako wynik T_i. Korzeń drzewa D, w wyniku zastosowania testu T, posiada jedno poddrzewo uzyskane w oparciu o T_i. Dalszy podział odbywa się w oparciu o rekurencyjne wywołanie kryterium T dla podzbioru D_i. Za każdym razem dokonujemy wyboru atrybutu, tworzymy rozgałęzienia według wartości, jakie przyjmuje dany atrybut i dla każdego węzła potomnego wywołujemy rekurencyjnie algorytm, z listą atrybutów zmniejszoną o właśnie wybrany atrybut.

Zakładając, że nie ma zdarzeń z identyczną wartością atrybutów należących do różnych klas, każdy test T, na którego podstawie uzyska się nietrywialny podział D, doprowa-

dzi w końcu do jednego z przypadków opisanych powyżej (do liścia lub węzła rozgałęziającego). Aby uzyskane w wyniku działania powyższej procedury drzewo charakteryzowało się jak najmniejszym rozmiarem (co ułatwia jego obsługę, zmniejsza złożoność obliczeniową i najczęściej prowadzi do lepszej predykcji), stosuje się cały zbór testów i wybiera ten podział, który maksymalizuje wartość kryterium. Aktualnie algorytm C4.5, jako kryterium podziału, stosuje się zysk informacyjny (*information gain*), bazujący na mierze informacji entropii informacji. C4.5 w sytuacji, gdy zysk informacyjny wynosi zero, stosuje dodatkowe kryteria stopu.

Zastosowanie powyższego schematu podziału prowadzi do uzyskania możliwie dobrego dopasowania drzewa do zbioru uczącego (treningowego). W rzeczywistych zastosowaniach, klasyfikowane dane są w dużym stopniu zaszumione. Praktyczne doświadczenia prowadzą do wniosku, że zaszumienie skutkuje zbytnim rozbudowaniem drzew, próbujących dostosować się do danych zawierających anomalie. Stosowanie mniejszych drzew z małą liczbą hipotez, zmniejsza prawdopodobieństwo pojawienia się zjawiska zbytniego dopasowania się do danych tzw. przetrenowania (*overfits*). Celem zapobieżenia takiemu zjawisku, stosuje się specjalne techniki przycinania drzewa (*pruning*). Są to zarówno kryteria stopu (przedstawione już wcześniej), jak i techniki określane mianem *prepruning* (stosowane podczas konstrukcji drzewa) i *postpruning* (stosowane dla drzew już utworzonych).

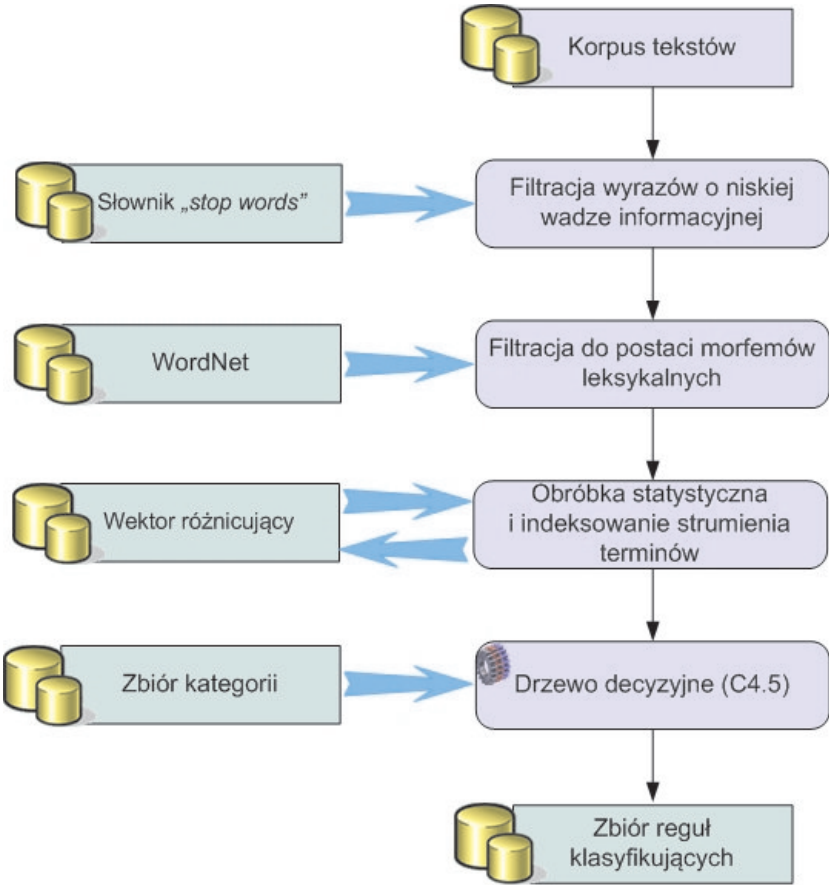
Pierwsza technika najczęściej opiera się na testach statystycznych, pozwalających na określenie chwili zaprzestania rozwijania drzewa, gdy w aktualnie rozważanym zbiorze D nie ma statystycznie znaczących powiązań pomiędzy jakimkolwiek atrybutem a klasą obiektu – np. test χ^2 jak to jest w C4.5. Druga technika ma na celu identyfikację węzłów i poddrzew, które są związane z istnieniem anomalii w klasyfikowanych danych. Wykorzystuje się tu metodę zastąpienia poddrzewa (*subtree replacement*) oraz metodę przycięcia poddrzewa (*subtree raising*), opierając się m.in. na tzw. testach wagi, estymacjach błędów lub zasadzie *MDL* (*minimum description length principle*).

Wśród metod stosowanych do nadzorowanej kategoryzacji tekstów, prócz drzew decyzyjnych można wyróżnić algorytmy: naiwny klasyfikator bayesowski (*Naive Bayes*) [8], klasyfikatory linowe [6], algorytm *k*-najbliższych sąsiadów (*k-nearest neighbors* – kNN) [7], SVM (*Support Vector Machines*) [15], GIS (*Generalized Instance Set*) [5], algorytm klasyfikacji bazujący na centroidach (*Centroid-Based Document Classifier*) [16].

4. Algorytm indeksowania korpusu uczącego

Proces indeksowania tekstów zawartych w korpusie (zbiorze) w celu określenia najlepszego słownika Γ, uzyskania odpowiedniej postaci reprezentujących je wektorów oraz nauki drzewa przedstawiono na rysunku 2. Korpus tekstów przeznaczony do nauki drzewa,

jest poddawany serii transformacji, w celu usunięcia ze słownika terminów (wyrazów) nie mających znaczenia z punku widzenia klasyfikacji oraz wstępnej minimalizacji rozmiaru wektora decyzyjnego. Pierwszym elementem jest **filtracja** *stop words* – terminów charakteryzujących się dużą liczbą wystąpień w języku angielskim. Zgodnie z prawem Zipfa iloczyn częstotliwości występowania danego wyrazu oraz jego „nośności" informacyjnej jest w przybliżeniu stały. W związku z tym, wyrazy najczęściej występujące w języku angielskim są usuwane z wszystkich tekstów zawartych w korpusie.



Rys. 2. Uproszczony algorytm tworzenia reprezentacji tekstów oraz nauki drzewa decyzyjnego

W tabeli 1 przedstawiono listę 315 najczęściej występujących terminów w języku angielskim, które są usuwane podczas procesu budowy słownika oraz indeksowania tekstów.

Tabela 1
Najczęściej występujące wyrazy w języku angielskim (315)

the	and	that	for	was	with	his	from	but	had
last	have	who	not	has	were	their	are	one	week
they	govern	all	year	its	britain	when	out	would	new
been	more	which	into	said	million	year	say	new	about
more	this	then	there	of	is	in	to	we	by
be	if	an	as	on	or	de	at	of	let
so	any	our	it	no	a		about	above	across
after	after-wards	again	against	all	almost	alone	along	already	also
although	always	among	amongst	an	and	another	any	anyhow	anyone
anything	anywhere	are	around	as	at	be	became	because	become
becomes	becoming	been	before	beforehand	behind	being	below	beside	besides
between	beyond	both	but	by	can		cannot	co	could
down	during	each	eg	either	else	elsewhere	enough	etc	even
ever	every	everyone	has	everywhere	except	few	first	for	former
formerly	from	further	had	everything	have	he	hence	her	here
hereafter	hereby	herein	hers	hereupon	herself	him	himself	his	how
however	i	ie	if	whereafter	inc	indeed		into	is
it	its	itself	last	latter	latterly	least	less	ltd	many
may	me	meanwhile	might	moreover	more	most	mostly	much	must
my	myself	namely	neither	neverthe-less	never	next	no	nobody	none
noone	nor	not	nothing	otherwise	or	now	off	often	on
once	one	only	onto	nowhere	other	others	of		our
ours	ourselves	out	over	sometimes	per	perhaps	rather	same	seem
seemed	seeming	seems	several	throughout	should	since	some-how	some	so
someone	something	sometime	own	somewhere	still	such	than	that	the
their	them	themselves	then	thence	there	thereafter	thereby	therefore	therein
thereupon	these	they	this	those	though	through	she	thru	
thus	to	together	too	toward	towards	under	until	Up	upon
us	very	via	was	whereby	well	were	what	whatever	when
whence	whenever	where	in	whereas	we	wherein	where-upon	wherever	whether
whither	which	while	who	whoever	whole	whom	whose	why	will
with	within	without	would	yourselves	you	your	yours	yourself	yet

Kolejnym etapem przygotowania reprezentacji wektorowej, jest **filtracja do postaci morfemów leksykalnych**, czyli doprowadzenie terminów występujących w tekście do postaci rdzenia pozbawionego elementów fleksyjnych. W celu przeprowadzenia tej operacji w sposób prawidłowy, unikając błędnego usuwania prefiksów i sufiksów prowadzącego do pojawienia się fałszywych elementów w wektorze reprezentującym tekst, zastosowano specjalistyczny słownik języka angielskiego WordNet [1]. WordNet jest pewnego rodzaju bazą danych, w którym przeszukiwanie odbywa się nie wze względu na podobieństwo alfabetyczne, lecz podobieństwo koncepcyjne. Obecnie system ten wyewoluował do postaci bazy danych zawierającej ogromny zbiór semantycznych zależności pomiędzy słowami w języku angielskim. WordNet, prócz wykorzystania go do uzyskania rdzenia wyrazu, stosowany jest do modyfikacji wagi danego wyrazu w opisie dokumentu, w przypadku pojawienia się synonimów. Przykładowo, jeśli w dokumencie występuje termin „samochód" i „auto" to są one uznawane za równoznaczne. Ponadto, jeśli dany wyraz jest homonimem (jest wieloznaczny), to w takim przypadku jego ranga w opisie dokumentu jest obniżana. W przyszłych wersjach systemu, przewidywane jest znaczne rozszerzenie zastosowania słownika WordNet, ponieważ zawierając taksonomię semantyczną umożliwia on generalizację (lub odwrotnie – specjalizację) pojęć zawartych w dokumencie. Zastosowanie takiego podejścia może doprowadzić do bardziej zwięzłego słownika, a więc i do prostszego opisu dokumentu.

Obróbka statystyczna i indeksowanie strumienia terminów dla każdego wyrazu, który został wytypowany jako współrzędna wektora reprezentującego poszczególne teksty, określa współczynnik tf_i (*trem-frequency*): $tf_i = a_{ij}/A_j$, gdzie: a_{ij} – liczba wystąpień i-tego wyrazu w j-tym dokumencie, A_j – liczba wszystkich terminów w j-tym dokumencie. Dodatkowo na tym etapie tworzony jest **wektor różnicujący** (*purity vector*) [2]. Jest to wektor terminów, które umożliwiają najlepszy podział korpusu tekstów na klasy. Jego umiejętny dobór pozwala na ograniczenie liczby współrzędnych (atrybutów) występujących w wektorach, będących reprezentacjami tekstów zawartych w korpusie. Tak uzyskane wektory, wraz z przydzieloną im klasą, stają się reprezentacją zbiorów uczących oraz testowych dla algorytmu drzewa decyzyjnego. W wyniku działania drzewa decyzyjnego otrzymuje się **zbiór reguł klasyfikujących**, pozwalający na automatyczną kategoryzację nowych dokumentów – niewystępujących w próbce uczącej.

5. Odwrotna częstość dokumentu

W algorytmie przedstawionym na rysunku 1 podstawowym problemem jest nadmierna wielkość wektora reprezentującego dokument. Z punktu widzenia jakości oraz wydajności procesu kategoryzacji, rozmiar wektora nie powinien być zbyt duży. W przyjętym rozwiązaniu parametr tf_i określa częstotliwość występowania cechy lub terminu (w tym przypadku słowa) w danym dokumencie. W zbiorze słów pojawiających się w dokumentach podzielonych na klasy, część z nich jest mniej lub bardziej charakterystyczna dla danej klasy. Toteż w celu zwiększenia wydajności procesu kategoryzacji, a co za tym idzie, wyboru tych terminów, które lepiej charakteryzują poszczególne klasy (posiadają większą zdolność

różnicowania klas) oraz zmniejszeniu rozmiaru wektora reprezentacji, w przedstawionym algorytmie, zastosowano mechanizm oparty na elementach algorytmu klasyfikującego bazującego na centroidach. Algorytm ten określa siłę różnicowania każdego terminu występującego w dokumencie, uwzględniając jednocześnie tę cechę dla wszystkich zaindeksowanych terminów. W podstawowej wersji algorytmu dokumenty są reprezentowane przez wektory w przestrzeni wektorowej. W tym celu dla każdego dokumentu jest tworzona reprezentacja – wektor częstotliwości $d_{tf} = (tf_1, tf_2, \ldots, tf_n)$, gdzie tf_i jest częstotliwością występowania i-tego terminu w dokumencie. Metoda ta oparta jest na spostrzeżeniu, że wraz ze wzrostem częstotliwości pojawiania się danego wyrazu w dokumentach zawartych w korpusie, spada jego użyteczności jako parametru ułatwiającego identyfikację tekstu jako członka danej klasy. Powyższe spostrzeżenie można zastosować do wektora częstotliwości reprezentującego dokument, poprzez pomnożenie każdej jego współrzędnej tf_i przez współczynnik $\log(N/d_{fi})$, gdzie N jest liczbą dokumentów w korpusie, natomiast d_{fi} określa liczbę dokumentów zawierający i-ty termin. Wyrażenie to nosi nazwę odwrotnej częstotliwości dokumentów (*Inverted Document Frequency* – IDF). Prowadzi to do następującej wektorowej reprezentacji dokumentu:

$$\vec{d}_i = (tf_1 \log(N/df_1), tf_2 \log(N/df_2), \ldots, tf_n \log(N/df_n)) \tag{1}$$

Dodatkowo, każdy utworzony według wzoru (1) wektor poddawany jest normalizacji:

$$\left\|\vec{d}_i\right\| = 1 \tag{2}$$

W modelu opartym na wektorowej reprezentacji dokumentów, podobieństwo dokumentów d_i i d_j jest określane jako wartość kosinusa kąta pomiędzy wektorami $\vec{d}_i$ i $\vec{d}_j$:

$$\cos(\vec{d}_i, \vec{d}_j) = \frac{\vec{d}_i \bullet \vec{d}_j}{\left\|\vec{d}_i\right\| * \left\|\vec{d}_j\right\|} \tag{3}$$

Zakładając, że wektory $\vec{d}_i$ i $\vec{d}_j$ mają znormalizowane, jednostkowe długości, na podstawie (3) otrzymujemy następujące wyrażenie pozwalające na określenie podobieństwa dwóch dokumentów:

$$\cos(\vec{d}_i, \vec{d}_j) = \vec{d}_i \bullet \vec{d}_j \tag{4}$$

Gdzie $\bullet$ oznacza iloczyn wektorowy. Dla zbioru S dokumentów można wyznaczyć wektor $\vec{C}$, będący jego reprezentacją w przestrzeni wektorowej, zwany *centroidem*:

$$\vec{C} = \frac{1}{|S|} \sum_{d \in S} \vec{d} \tag{5}$$

Wektor $\vec{C}$ można więc interpretować jako średnią liczbę wszystkich terminów w dokumentach zawartych w zbiorze S. Analogicznie jak w przypadku podobieństwa dwóch

dokumentów (równanie (4)) możemy określić podobieństwo dwóch zbiorów dokumentów oraz podobieństwo zbioru i pojedynczego dokumentu. Dla pierwszego przypadku podobieństwo można określić następująco:

$$\cos(\vec{C}_i, \vec{C}_j) = \frac{\vec{C}_i \bullet \vec{C}_j}{\left\|\vec{C}_i\right\| * \left\|\vec{C}_j\right\|} \tag{6}$$

natomiast dla drugiego:

$$\cos(\vec{d}_i, \vec{C}_j) = \frac{\vec{d}_i \bullet \vec{C}_j}{\left\|\vec{d}_i\right\| * \left\|\vec{C}_j\right\|} \tag{7}$$

Dla każdego zbioru dokumentów należących do tej samej klasy, określamy odpowiadający jej centroid $\vec{C}$. W przypadku istnienia *k* zbiorów (klas dokumentów), otrzymujemy zbiór centroidów $\left\{\vec{C}_1, \vec{C}_2, ..., \vec{C}_k\right\}$, gdzie $\vec{C}_i$ jest centroidem wyznaczonym dla *i*-tej klasy [1, 14].

Dla nowego dokumentu *x*, przynależność do klasy jest ustalana następująco:

1) wyznaczany wektorową reprezentację dokumentu $\vec{x}$, korzystając z odwrotnej częstotliwości dokumentów (IDF – równanie (1)) określonych dla całego zbioru uczącego,
2) dokonujemy normalizacji wektora $\vec{x}$ (równanie (2)),
3) obliczamy wartość podobieństwa pomiędzy dokumentem a centroidami ze zbioru $\left\{\vec{C}_1, \vec{C}_2, ..., \vec{C}_k\right\}$ za pomocą wyrażenia (7). Ostatecznie, klasę *C',* do której należy dokument, określa się na podstawie wyrażenia:

$$C' = \arg \max_{C_j \subset \left\{\vec{C}_1, \vec{C}_2, ..., \vec{C}_k\right\}} (\cos(\vec{x}, \vec{C}_j)) \tag{8}$$

Złożoność obliczeniowa fazy nauki algorytmu bazującego na centroidach jest liniowo zależna od liczby dokumentów i terminów występujących w dokumentach. Określenie wektorowej reprezentacji dokumentów jest możliwe w co najwyżej trzech przebiegach algorytmu przez zbiór uczący. Podobnie, *k* centroidów można wyznaczonyć w pojedynczym przebiegu algorytmu przez zbór treningowy. Ostatecznie, klasyfikacja nowego dokumentu może zostać dokonana w czasie $O(km)$, gdzie *k* to liczba klas, a *m* liczba terminów występujących w dokumencie. Ogólna złożoność obliczeniowa algorytmu jest niska i zbliżona do złożoności obliczeniowej szybkich algorytmów klasyfikujących takich jak *Naive Bayesian.*

6. Wektor różnicujący

W przedstawionym algorytmie, siłę różnicowania terminów określa się podobnie jak w przypadku indeksu Giniego (*Gini coefficient*) [9]. Niech *k*, będzie liczbą klas, natomiast

$\{\vec{C}_1, \vec{C}_2, ..., \vec{C}_k\}$ centroidami odpowiadającymi poszczególnym klasom. Dla wszystkich m różnych terminów występujących w zbiorze uczącym, niech wektor $\vec{T}_i = \{T_{1,i}, T_{2,i}, ..., T_{k,i}\}$ będzie liczbą wystąpień i-tego terminu wśród dokumentów należących do każdego z k centroidów. Zdolność różnicowania i-tego terminu P_i można określić następująco:

$$P_i = \sum_{j=1}^{k} T_{j,i}^2 \quad \text{gdzie} = 1, ..., m \tag{9}$$

Zauważmy, że współczynnik P_i przyjmuje najmniejszą wartość, gdy $T_{1,i} = T_{2,i} = ... = T_{k,i}$. W związku z tym, wektor różnicujący (*purity vector*) można zdefiniować, biorąc pod uwagę n najlepiej różnicujących terminów charakteryzujących się największą wartością P_i.

$$\vec{P} = \{P_1, ..., P_n\} \tag{10}$$

W zaproponowanym algorytmie wartość n dobiera się arbitralnie tak, aby zmniejszyć rozmiar wektora opisującego dokument. Pozwala to na wzrost szybkości oraz jakości klasyfikacji dokumentów.

7. Eksperymenty obliczeniowe

Metodyka przeprowadzonego eksperymentu zakłada zastosowanie 14 000 dokumentów w celu nauki drzewa decyzyjnego. Następnie, badanie efektywności klasyfikacji przez tak utworzone drzewo jest przeprowadzone na 4200 dokumentach testowych. W tym celu zaimplementowano oprogramowanie w C++ oraz C#, pozwalające w sposób automatyczny importować i filtrować teksty z predefiniowanych kategorii tematycznych portali internetowych.

W przedstawionych eksperymentach określono k = 14 klas (kategorii) tekstów. Teksty oraz ich kategorie uzyskano, korzystając z jednego ze znanych amerykańskich portali (Yahoo), klasyfikujących strony WWW pod względem tematycznym. Zgodnie z podziałem portalu Yahoo, wyróżniono czernaście następujących kategorii*: Adult, Business, Regional, Science, Home, News, Arts, Shopping, Recreation, Health, Games, Computers, Society, Sports*. Podział na kategorie w serwisie odzwierciedla rzeczywistą tematykę dokumentów. W procesie nauki drzewa wykorzystano łącznie 14 000 dokumentów HTML (po 1000 dla każdej kategorii), które przed użyciem zostały pozbawione tagów HTML, skryptów, definicji stylów oraz dodatkowo zweryfikowane pod względem tematyki. Zgodnie z metodyką stosowaną w algorytmie C4.5 proces tworzenia drzewa jest dwuetapowy. W pierwszym etapie wykorzystano 7000 z tych dokumentów, w celu stworzenia wstępnej wersji drzewa decyzyjnego. Kolejny etap doskonalenia drzewa polegający na jego przycinaniu (*post-pruning*) przeprowadzono korzystając z pozostałych 7000 dokumentów.

W celu określenia wydajności klasyfikacji, wyznaczono wartość współczynnika, zwanego efektywnością klasyfikacji:

$$E_i = \frac{TP_i}{|C_i|} * 100\% \quad i = 1, ..., k \tag{11}$$

gdzie:

E_i – wartość efektywności klasyfikacji dla *i*-tej klasy w [%],
TP_i – liczba poprawnie sklasyfikowanych (*true positive*) dokumentów z *i*-tej klasy testowej (czyli takich, dla których dokonana automatyczna klasyfikacja jest zgodna z rzeczywistą kategorią dokumentu określoną przez eksperta),
$|C_i|$ – liczba wszystkich dokumentów w *i*-tej klasie,
k – liczba klas.

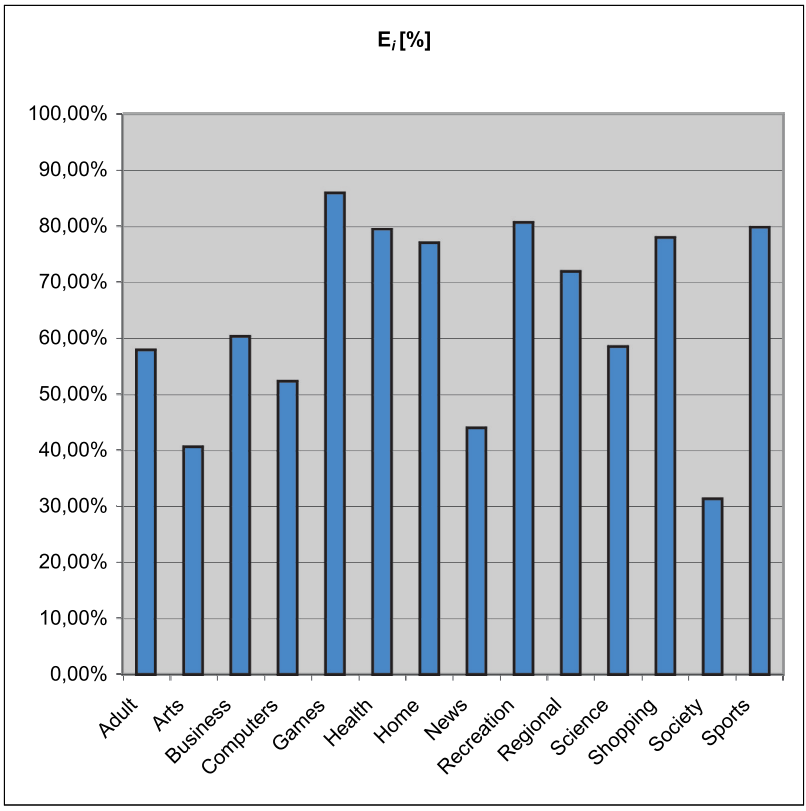
Średnia efektywność została określona następująco:

$$E = \frac{\sum_{i=1}^{k} E_i}{k} * 100\% \quad i = 1, ..., k \tag{12}$$

Efektywność opisana wzorami (11) oraz (12) jest typową miarą jakości klasyfikacji, stosowaną dla metod klasyfikacji jednoetykietowej [13] (zakłada się, że dokument należy tylko do jednej klasy). Mając zbiór testowy podzielony już na kategorie, dokonuje się klasyfikacji dokumentów z powyższego zbioru za pomocą wytrenowanego drzewa decyzyjnego. W celu wyznaczenia wartości TP_i, dla *i*-tej klasy określana jest liczba dokumentów prawidłowo skategoryzowanych przez drzewo decyzyjne (czyli takich, których predefiniowania kategoria zgadza się z kategorią określoną przez drzewo decyzyjne). Do badań efektywności algorytmu użyto 4200 tekstów z korpusu tekstów Reuters-21578 [12], które to zostały przydzielone do jednej z 14 już wcześniej określonych kategorii. W tabeli 2 zamieszczono wyniki przeprowadzonych testów. Na przekątnej tablicy umieszczono wartość $TP_i/|C_i|$, natomiast w pozostałych komórkach wiersza umieszczono liczbę dokumentów błędnie sklasyfikowanych jako należące do innych kategorii (*false positive*). Rysunek 3 ilustruje efektywność klasyfikacji dla poszczególnych klas dokumentów, określoną wyrażeniem (11).

Należy zaznaczyć, że proces uczenia oraz testowania drzewa decyzyjnego odbywał się na korpusie tekstów **bardzo niskiej** jakości. W swej większości teksty znajdujące się w na wielu sklasyfikowanych stronach w Internecie są krótkie (rzędu 30–100 słów), a ich tematyka jest często niejednoznaczna. Należy także podkreślić dużą „rozległość" i niejednoznaczność predefiniowanych klas dokumentów, które służyły do nauki drzewa. Każda z nich była jeszcze podzielona na kilkadziesiąt podklas. Przykładowo, klasa *Society*, dla której uzyskano najniższy współczynnik efektywności klasyfikacji, składa się z 21 odległych o siebie tematycznie podklas (*Crime, Death, Future, Genealogy, Government, History, Holidays, Law, Military, Paranormal, People, Philosophy, Politics, Relationships,*

Religion_and_Spirituality, Sexuality, Subcultures, Support_Groups, Transgendered, Urban_Legends, Work). Klasa *Sport* ma ich natomiast ponad 100. Tak dobrana próbka ucząca miała na celu zbadanie zachowania się opracowanego algorytmu w skrajnie trudnych warunkach, gdy dostarczane do treningu teksty będą krótkie i mało różniące się od siebie. Na taką sytuację często natrafiamy np. w przypadku sieci WWW.



Rys. 3. Efektywność klasyfikacji dla poszczególnych klas określona wzorem (11)

Jak to przedstawiono na rysunku 3, podczas testów dla kolejnych klas osiągnięto następujące wartości parametru efektywności klasyfikacji określonej wzorem (11): *Adult* – 57,9%, *Arts* – 40,6%, *Business* – 60,4%, *Computers* – 52,3%, *Games* – 85,9%, *Health* – 79,5%, *Home* – 77,1%, *News* – 44,0%, *Recreation* – 80,7%, *Regional* – 71,9%, *Science* – 58,5%, *Shopping* – 78,0%, *Society* – 31,4%, *Sports* – 79,8%. Jak widać, najwyższą efektywność klasyfikacji uzyskano dla klas dokumentów, w których występują specyficzne terminy (*Games*, *Recreation*, *Sports* oraz *Health*), a najniższą dla klasy dokumentów z klas *Society* oraz *News*, w przypadku których raczej trudno się spodziewać istnienia specyficznej terminologii.

Tabela 2
Efektywność klasyfikacji dokumentów

	Adult	Arts	Business	Computers	Games	Health	Home	News	Recreation	Regional	Science	Shopping	Society	Sports
Adult	280/484	0	17	23	4	6	5	27	12	36	8	15	34	17
Arts	6	115/283	10	25	6	15	4	26	4	16	2	12	24	18
Business	4	3	194/321	17	6	15	5	14	14	12	21	5	7	4
Computers	9	10	16	136/260	13	6	8	8	5	6	9	10	15	9
Games	1	3	4	5	268/312	5	3	3	2	4	1	1	9	3
Health	5	5	5	8	1	229/288	1	7	5	6	5	2	4	5
Home	6	4	3	5	6	11	272/353	3	5	1	6	14	10	7
News	9	16	14	12	4	7	9	121/275	16	12	14	10	21	10
Recreation	1	4	7	3	4	4	5	6	217/269	9	4	2	3	0
Regional	3	2	9	5	6	2	4	16	9	205/285	8	2	6	8
Science	4	2	25	25	4	8	5	12	8	16	190/325	7	16	3
Shopping	10	5	8	8	1	2	7	9	3	6	4	251/322	6	2
Society	18	21	10	17	7	9	6	22	7	8	12	9	71/226	9
Sports	1	3	3	5	2	2	2	7	2	3	4	5	2	162/203

8. Podsumowanie

Uzyskane wyniki wskazują na duży potencjał leżący w połączeniu metod stosowanych w kategoryzacji opartej o centroidy z algorytmami drzewa decyzyjnego. Oba te algorytmy osobno wykazują się sporą efektywnością klasyfikacji [3] dla dobrze określonych klas dokumentów.

W zaprezentowanych w artykule eksperymentach, proces uczenia oraz testowania drzewa decyzyjnego odbywał się na niskiej jakości korpusie tekstów dostępnych w Internecie. Pozwoliło to na zbadanie zachowania się opracowanego algorytmu w sytuacji, gdy przetwarzane teksty, w etapie nauki oraz testów, są krótkie i mało zróżnicowane.

Wartość średniej efektywności klasyfikacji określonej wzorem (12) wynosi $E = 64\%$. Należy zaznaczyć, że powyższy algorytm uzyskiwał dużo wyższe wartości średniej efektywności (rzędu 93–96%) dla dziedzin, w których występują specyficzne pojęcia, takie jak matematyka dyskretna, analiza matematyczna, topologia, fizyka, etc. W tym przypadku efektywność klasyfikacji osobno dla algorytmu C.45 oraz opartego o centroidy wynosiła odpowiednio 79% i 86% [3]. Można zatem stwierdzić, że zastosowanie metod zaczerpniętych z algorytmów klasyfikujących opartych o centroidy oraz *indeks Giniego*, do wyznaczenia najlepszego słownika oraz reprezentacji dokumentów, pozwala na uzyskanie zadawalających wyników w przypadku niskiej jakości próbki uczącej.

Literatura

[1] Fellbaum Ch.(ed.), *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

[2] Han E.H, Karypis G., *Centroid-Based Document Classification: Analysis & Experimental Results.* Proc. of the Fourth European Conference on the Principles of Data Mining and Knowledge Discovery, 2000, 424–431.

[3] Han E.H, Karypis G., *Centroid-based document classification algorithms: Analysis & experimental results.* Technical Report TR-00-017, Department of Computer Science, University of Minnesota, Minneapolis, 2000.

[4] Hunt E.B., Marin J., Stone P.J., *Experiments in Induction*. Academic Press, New York, 1966.

[5] Lam W., Ho Ch. Y., *Using a generalized instance set for automatic text categorization*. Proc. of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, 1998, 81–89.

[6] Lewis D.D., Shapire R.E., Callan J.P., Papka R., *Training algorithms for linear text classifiers.* Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, 1996, 298–306.

[7] Masand B., Linoff G., Waltz D., *Classifying news stories using memory based reasoning*. Proc. of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, 1998, 59–64.

[8] McCallum A., Nigam K.A., *Comparison of event models for naive bayes text classification*. Proc. AAAI/ICML-98 Workshop on Learning for Text Categorization, Technical Report WS-98-05, 1998.

[9] Ogwang T., *A Convenient Method of Computing the Gini Index and its Standard Error*. Oxford Bulletin of Economics and Statistics, 62, Oxford, 2000, 123–129.

[10] Quinlan J.R., *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[11] Quinlan J.R., *Improved use of continuous attributes in c4.5*. Journal of Artificial Intelligence Research, 4, 1996, 77–90.
[12] *Reuters-21578 text categorization test collection.* www.daviddlewis.com/ resources/ testcollections/reuters21578, Cambridge, 1998.
[13] Sebastiani F., *Text Categorization*. Dipartimento di Matematica Pura e Applicata Universit'a di Padova, Padwa, 2004.
[14] Tibshirani R., Hastie T., Narasimhan B., Chu G., *Diagnosis of multiple cancer types by shrunken centroids of gene expression.* Department of Health, Research and Policy, and Statistics, Stanford University, Stanford, 2002.
[15] Vapnic V., *The Nature of Statistical Learning Theory*. Springer, 1995.
[16] Yang Y., Liu X., *A re-examination of text categorization methods*. Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, 1999, 42–49.