Informatyka, studia STACJONARNE, I st. semestr VI

**Komputerowe systemy rozpoznawania** **2020/2021**

Prowadzący: prof. dr hab. inż. Adam Niewiadomski Poniedziałek 13:45

Data oddania: _____ Ocena: _____

Przemysław Lis 229940
Michał Olczak 229972

# Projekt 1. Klasyfikacja dokumentów tekstowych

## 1. Cel projektu

Celem projektu jest stworzenie aplikacji która podejmie się klasyfikacji 21578 artykułów prasowych z 1987 roku [1]. Aplikacja odpowiedzialna będzie za sparsowanie danych i określenie jaki kraj opisuje dany artykuł. Klasyfikacja będzie możliwa w obrębie 6 klas (Kanada, USA, UK, Francja, Niemcy, Japonia) w zależności od wyżej opisanego kryterium. Do wykonania zadania wykorzystamy algorytm $k$-NN aby należycie dopasować kraj. Oczekujemy że nasz program po analizie danych algorytmem będzie w stanie poprawnie dopasować kraj o którym mowa lub z którego pochodzi artykuł. W projekcie porównamy również jaki wpływ na poprawność klasyfikacji oraz jej jakość będzie miało użycie różnych cech, metryk, wartości $k$ oraz odpowiedniego podziału na zbiór uczący i testowy

## 2. Klasyfikacja nadzorowana metodą $k$-NN

W aplikacji wykorzystamy algorytm $k$-NN ($k$ nearest neighbours)[2][3]. Jest to algorytm regresji nieparametrycznej. Założeniem algorytmu jest że podobne problemy mają podobne rozwiązania. Algorytm sprawdza $k$ najbliższych sąsiadów (gdzie $k \in \mathbb{N}$) artykułu i w zależności od wyniku klasyfikuje jego położenie. Do obliczenia odległości artykułu od jego sąsiadów używamy różnych metryk np. Odległość Euklidesowa. Jeżeli w sąsiedztwie naszego artykułu badawczego będzie najwięcej węzłów danego typu, to wtedy zostanie

on odpowiednio dopasowany do tego typu. Parametrem wejściowym jest plik tekstowy, który następnie będzie zaklasyfikowany do odpowiedniego kraju. Algorytm do obliczania odległości pomiędzy artykułami będzie wykorzystywał wektory cech wyekstrahowanych ($\vec{v} = \langle C_1, C_2, C_3, \cdots, C_n \rangle$, $n \in [1; 10] \wedge n \in \mathbb{N}$) z artykułów (spójrz rozdział 2.1) oraz odpowiednio wybranej metryki. Algorytm jako parametry będzie przyjmował:

1. Wspomnianą wyżej metrykę (opisane szczegółowo w akapicie 3)
2. Zbiór artykułów których klasę znamy - zbiór uczący
3. Zbiór wyekstrahowanych cech na podstawie której odbywać się będzie klasyfikacja
4. Wartość $k$ która oznacza ile najbliższych sąsiadów będziemy brać pod uwagę przy klasyfikacji

### 2.1. Ekstrakcja cech, wektory cech

1. Liczba całkowita reprezentująca największą ilość wyrazów w pierwszych 5 zdaniach.

$$\sum_{i=0}^{5} s_i, \text{ gdzie s - słowo} \wedge s \in A_{f5}$$

Gdzie $A_{f5}$ - Fragment pierwszych 5 zdań artykułu.

2. Stosunek liczby wystąpień słów kluczowych do długości tekstu.

$$\frac{\sum_{i=0}^{n} s_i}{d}, \text{ gdzie s - słowo kluczowe, d - długość tekstu}$$

3. Długość artykułu.

$$\sum_{i=0}^{n} l_i, \text{ gdzie l - litera}$$

4. Średnia dlugość słowa.

$$\frac{\sum_{i=0}^{n} l_i}{\sum_{j=0}^{n} s_j}, \text{ gdzie l - litera, s - słowo}$$

5. Liczba unikalnych słów.

$$\sum_{i=0}^{n} s_i, \text{ gdzie s - słowo unikalne}$$

6. Pierwsze wystąpienie we fragmencie tekstu nazwy kontynentu lub jego mieszkańca.

$$k_f = k,\ k \in C_6 \wedge k \in A$$

Gdzie $k_f$ - pierwsze wystąpienie kontynentu, $k$ - kontynent ze zbioru $C_6$, $C_6$ - zbiór możliwych kontynentów (słów kluczowych), $A$ - artykuł

$C_6$ = {Europe, European, America, American, Asia, Asian}

7. Liczba słów nie przekraczająca 3 znaków.

$$\sum_{i=0}^{n} len(s_i) <= 3,\text{gdzie s - słowo}$$

8. Liczba słów.

$$\sum_{i=0}^{n} s_i,\ \text{gdzie s - słowo}$$

9. Liczba słów dłuższych niż 8 znaków.

$$\sum_{i=0}^{n} len(s_i) > 8,\text{gdzie s - słowo}$$

10. Najliczniej występująca nazwa własna miasta bądź regionu ze zbioru $C_{10}$ w pierwszych pięciu zdaniach artykułu.

$$max\left(\sum_{i=0}^{n} s_i\right), s_i \in C_{10} \wedge s_i \in A_{f5}$$

Gdzie odpowiednio $s_i$ - wystąpienie regionu w tekście, $C_{10}$ - zbiór słów kluczowych, $A_{f5}$ - Fragment pierwszych 5 zdań artykułu.

$C_{10}$ = {Berlin, Frankfurt, Bonn, Leverkusen, Nuremberg, Hanover, Weisbaden, Stuttgart, Monachium, Tokyo, Yokohama, Ottawa, Paris, Lyon, Torronto, London, Manchester, Liverpool, Birmingham, Washington, New York, Boston, Los Angeles}

### 2.2. Miary jakości klasyfikacji

W naszym projekcie będziemy korzystać z 4 różnych miar jakości otrzymanej klasyfikacji (Accuracy, Precison, Recall, F1). Z pośród wymienionych 4 miar jakości, Accuracy będzie obliczana dla całego zbioru artykułów, natomiast 3 pozostałe będą obliczane zarówno dla całego zbioru artykułów jak i dla pojedynczych klas. Od tej pory będziemy korzystać również z określeń jakości przypisania klas takich jak:

— TP (True positive) - oznacza artykuły, które zostały przypisane do danej klasy i powinny do niej należeć
— FP (False positive) - oznacza artykuły, które zostały przypisane do danej klasy i **NIE** powinny do niej należeć
— TN (True negative) - oznacza artykuły, które **NIE** zostały przypisane do danej klasy, oraz **NIE** powinny się w niej znaleźć
— FN (False Negative) - oznacza artykuły, które **NIE** zostały przypisane do danej klasy, ale powinny się w niej znaleźć

**Accuracy** - to miara która określa dokładność klasyfikacji. Jest to stosunek ilości poprawnie zaklasyfikowanych artykułów do wszystkich artykułów.
Określona wzorem:

$$accuracy = \frac{\sum_{n=1}^{6} TP_n}{A} \tag{1}$$

Gdzie $TP_n$ - ilość wszystkich poprawnie przypisanych artykułów do danej klasy, $A$ - liczba wszystkich artykułów

**Precision** - to miara określająca precyzję klasyfikacji elementów do danej klasy. Liczona jest tak jak accuracy, lecz w obrębie pewnej klasy.
Określa się wzorem:

$$precision_n = \frac{TP_n}{TP_n + FP_n} \tag{2}$$

Gdzie $TP_n$ - liczba poprawnie zaklasyfikowanych artykułów do klasy $n$, $FP_n$ - liczba niepoprawnie zaklasyfikowanych artykułów do klasy $n$

Dla całego projektu wartość precision definiowana jest jako średnia ważona precision dla danych klas:

$$precision = \frac{\sum_{n=1}^{6} precision_n \cdot TP_n}{\sum_{n=1}^{6} TP_n} \tag{3}$$

Gdzie $precision_n$ - wartość precyzji dla danej klasy $n$, $TP_n$ - artykuły poprawnie przypisane do klasy $n$.

Miara precision przyjmuje wartości z zakresu $\langle 0, 1\rangle$ oraz należy do $\mathbb{R}$ i odpowiada za dokładność rozpoznania obiektów w obrębie danej klasy.

**Recall** - jest to miara oznaczająca odsetek poprawnie zaklasyfikowanych artykułów do danej klasy.
Określa się wzorem:

$$recall_n = \frac{TP_n}{TP_n + FN_n} \tag{4}$$

Gdzie $TP_n$ - liczba poprawnie zaklasyfikowanych artykułów do klasy $n$, $FN_n$ - liczba artykułów które nie zostały zaklasyfikowane do klasy $n$ ale powinny się w niej znaleźć

Dla całego projektu wartość recall definiowana jest jako średnia ważona recall dla danych klas:

$$recall = \frac{\sum_{n=1}^{6} recall_n \cdot TP_n}{\sum_{n=1}^{6} TP_n} \tag{5}$$

Gdzie $recall_n$ - wartość recall dla danej klasy $n$, $TP_n$ - artykuły poprawnie przypisane do klasy $n$.

Miara recall przyjmuje wartości z zakresu $\langle 0, 1\rangle$ oraz należy do $\mathbb{R}$ i obrazuje

stosunek poprawnie rozpoznanych artykułów do wszystkich które powinny być rozpoznane z danej klasy.

**F1** - jest to miara uśredniająca wynik. Liczona jest poprzez średnią harmoniczną precision oraz recall.
Określa się wzorem:

$$F1_n = 2 \cdot \frac{precision_n \cdot recall_n}{precision_n + recall_n} \tag{6}$$

Dla całego projektu wartość F1 definiowana jest jako średnia ważona wartości F1 dla danych klas:

$$F1 = \frac{\sum_{n=1}^{6} F1_n \cdot TP_n}{\sum_{n=1}^{6} TP_n} \tag{7}$$

Gdzie $F1_n$ - wartość F1 dla danej klasy $n$, $TP_n$ - artykuły poprawnie przypisane do klasy $n$.

Miara F1 również przyjmuje wartości z zakresu $\langle 0, 1 \rangle$ oraz należy do $\mathbb{R}$.

| | | **Rzeczywista Klasa Artykułu** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **USA** | **UK** | **Canada** | **Germany** | **France** | **Japan** |
| **Dopasowana Klasa Artykułu** | **Usa** | 5411 | 114 | 96 | 65 | 41 | 23 |
| | **UK** | 354 | 3790 | 222 | 142 | 75 | 9 |
| | **Canada** | 311 | 120 | 2255 | 105 | 45 | 29 |
| | **Germany** | 123 | 91 | 18 | 1725 | 135 | 17 |
| | **France** | 100 | 75 | 66 | 25 | 1510 | 9 |
| | **Japan** | 18 | 14 | 52 | 17 | 7 | 312 |

Tabela 1. Przykładowy wynik klasyfikacji

Na podstawie powyższej tabeli oraz wzorów 1, 2, 4, 6 obliczamy wartości miar końcowych dla przykładowych wartości klasyfikacji.

| | Precision | Recall | F1 | Accuracy |
|---|---|---|---|---|
| USA | 0.9410 | 0.8566 | 0.9419 | |
| UK | 0.8253 | 0.9015 | 0.8242 | |
| Canada | 0.7871 | 0.8324 | 0.7886 | |
| Germany | 0.8179 | 0.8297 | 0.8254 | 0.8563 |
| France | 0.8459 | 0.8329 | 0.8503 | |
| Japan | 0.7486 | 0.7666 | 0.7533 | |
| Średnia | 0.8276 | 0.8366 | 0.8306 | |

Tabela 2. Przykładowe wartości miar obliczone na podstawie tabeli 2.2

# 3. Klasyfikacja z użyciem metryk i miar podobieństwa tekstów

W naszym projekcie do obliczenia odległości danych artykułów wykorzystaliśmy trzy metryki:

— Metryka Euklidesowa - Aby obliczyć odległość $d_e$ między dwoma artykułami $X, Y$ należy obliczyć pierwiastek sumy kwadratów odległości wartości cech o tych samych indeksach.

$$d_e(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{8}$$

Gdzie $d_e$ - odległość euklidesowa, $X, Y$ - wektory cech artykułów pomiędzy którymi liczymy odległość, $x_i, y_i$ - wartości cech wektorów $(X, Y)$[1]

— Metryka Manhattan (miejska) - Aby obliczyć odległość $d_m$ między dwoma artykułami $X, Y$ należy obliczyć sumę wartości bezwzględnych odległości poszczególnych cech $x_i, y_i$

$$d_m(X,Y) = \sum_{i=1}^{n}|x_i - y_i| \tag{9}$$

gdzie: $d_m$ - odległość miejska, $X, Y$ - wektory cech artykułów pomiędzy którymi liczymy odległość, $x_i, y_i$ - wartości cech wektorów $(X, Y)$[1]

— Metryka Czebyszewa - Aby obliczyć odległość $d_{Ch}$ między dwoma artykułamu $X, Y$ należy wybrać maksymalną wartość, wartości bezwzględnych odległości pomiędzy poszczególnymi wartościami cech $x_i, y_i$

$$d_{Ch}(X,Y) = \max_i |x_i - y_i| \tag{10}$$

gdzie: $d_{Ch}$ - odległość Czebyszewa, $X, Y$ - wektory cech artykułów pomiędzy którymi liczymy odległość, $x_i, y_i$ - wartości cech wektorów $(X, Y)$[1]

[1] W przypadku cech liczbowych jest to różnica ich wartości, w przypadku cech tekstowych obliczany jest 1-$sim_n$ (patrz wzór 11)

Aby obliczyć podobieństwo łańcuchów tekstowych zastosowaliśmy miarę $n$-gramów. Metoda ta opiera się na sprawdzaniu podobieństwa dwóch łańcuchów tekstowych na podstawie ilości wspólnych podciągów w łańcuchach $s_1$ oraz $s_2$. Jako $s_1$ zawsze przyjmujemy krótsze z dwóch słów. Jeżeli krótsze słowo ma więcej niż 3 litery to przyjmujemy $n = 3$, w przeciwnym przypadku n jest równe długości krótszego słowa. Miara ta przyjmuje wartości z zakresu $\langle 0, 1\rangle$ oraz należy do $\mathbb{R}$.

$$sim_n(s_1, s_2) = \frac{1}{N - n + 1} \sum_{i=1}^{N-n+1} h(i) \tag{11}$$

gdzie, $sim_n$ - wartość prawdopodobieństwa $h(i)$ - funkcja zwracającaa wartość 1 jeżeli ciąg z $s_1$ występuje przynajmniej raz w $s_2$ w przeciwnym wypadku funkcja zwraca wartość 0, $N-n+1$ - ilość możliwych $n$-elementowych podciągów w $s_1$

Aby obliczyć odległość pomiędzy dwoma wartościami tekstowymi musimy najpierw obliczyć podobieństwo pomiędzy tymi tekstami (wzór 11 i następnie odjąć je od 1:

$$d(s_1, s_2) = 1 - sim_n(s_1, s_2) \tag{12}$$

gdzie: $d$ - odległość pomiędzy dwoma cechami tekstowymi, $s_1, s_2$ - wartości tekstowe cech, $sim_n$ - podobieństwo pomiędzy dwoma wartościami tekstowymi (wzór 11).

Przykładowo dla $s_1$ = "kaptur" i $s_2$ = "kapelusz" mamy $n = 3$ i:

$$d(s1, s2) = 1 - \frac{1}{4} \cdot (1 + 0 + 0 + 0) = 1 - \frac{1}{4} = \frac{3}{4} \tag{13}$$

Przyjmijmy przykładowe wektory:

1. X = [”Boston”, ””, ”America”, ”Statue of Liberty”, ””, ”Inc”, ”White House”, ”United States”, 0, 7, 1, 5, 24, 234, 12]
2. Y = [”New York”, ””, ”Canada”, ”Europe”, ”kilograms”, ”Inc”, ”Germany”, ”Berlin Wall”, 14, 2, 4, 4, 0, 200, 110]

Dla podanych wektorów otrzymujemy:

— Metryka **Euklidesowa**

$$\begin{aligned} d_e(X,Y) &= (1^2+0^2+1^2+1^2+1^2+0^2+1^2+1^2 \\ &+ 14^2+5^2+3^2+1^2+24^2+34^2+98^2)^{\frac{1}{2}} = \\ &= \sqrt{1+0+1+1+1+0+1+1+196+25+9+1+576+1156+9604} \\ &= 107.58 \end{aligned} \tag{14}$$

— Metryka **Manhattan**

$$\begin{aligned} d_m(X,Y) &= |1-0|+|1-1|+|1-0|+|1-0|+|1-0|+|1-1| \\ &+ |1-0|+|1-0|+|0-14|+|7-2|+|1-4|+|5-4| \\ &+ |24-0|+|234-200|+|12-110| = 1+0+1+1+1+0 \\ &+ 1 + 1 + 14 + 5 + 3 + 1 + 24 + 34 + 98 = 185 \end{aligned} \tag{15}$$

— Metryka **Czebyszewa**:

$$\begin{aligned} d_{Ch}(X,Y) &= \max(|1|,|0|,|1|,|1|,|1|,|0|, \\ &|1|,|1|,|14|,|5|,|3|,|1|,|24|,|34|,|98|) = 98 \end{aligned} \tag{16}$$

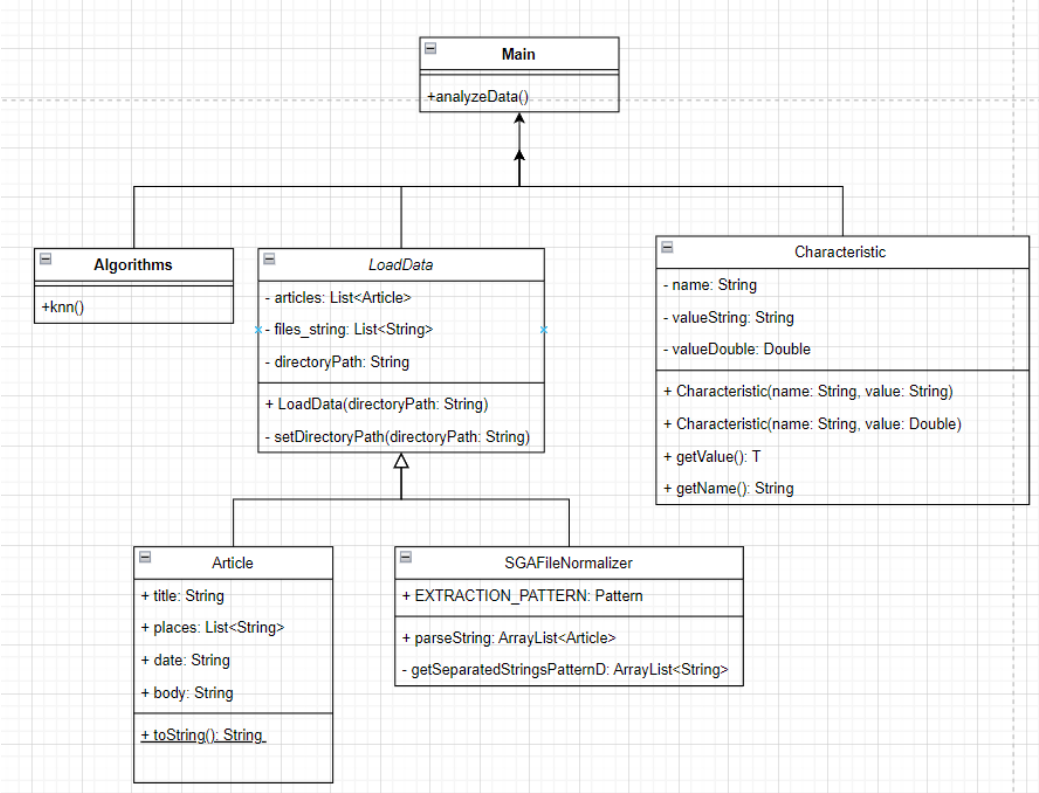
# 4. Budowa aplikacji

## 4.1. Diagramy UML

**- Klasa Main**
Program posiada klasę Main której zadaniem jest obsługa całego algorytmu. funkcją tej klasy jest kontrolowanie wczytania danych jak i obsługa algorytmu jak i użycia charakterystyk. Klasa ta jako rezulatat będzie zwracać wyniki metryk dla danych wartości algorytmu.

**- Klasa Algorithms**
W tej klasie będzie przechowywany algorytm KNN. Cała implementacja tego algorytmu będzie miała swoje miejsce w tej klasie.

**- Klasa LoadData**
Kolejną klasą jest LoadData która będzie służyła do ładowania plików. Aby poprawnie to zrobić będą potrzebne kolejne klasy takie jak **Article** która przechowuje informacje na temat artykułu. Posiada ona atrybuty takie jak




Rysunek 1. Diagram UML projektu

tytuł, miejsce, date oraz treść artykułu. Klasa ta posiada metodę toString aby zwrócić wszystkie dane odnośnie artykułu. Kolejną klasą która będzie potrzebna w poprawnym ładowaniu danych jest **SGAFileNormalizer** która w swoich atrybutach posiada definicje typu regex która pomoże wyekstrachować dane z dokumentu w taki sposób jaki jest nam potrzebny do poprawnego działania algorytmu.

**- Klasa Characteristic**
Jest to klasa która będzie generycznie tworzona w oparciu o charakterystyki z punktu 2.1. Głównym zadaniem tej klasy będzie przechowywanie danych oraz algorytmów potrzebnych do obliczania macierz charakterystyk.

### 4.2. Prezentacja wyników, interfejs użytkownika

Krótki ilustrowany opis jak użytkownik może korzystać z aplikacji, w szczególności wprowadzać parametry klasyfikacji i odczytywać wyniki. Wersja JRE i inne wymogi niezbędne do uruchomienia aplikacji przez użytkownika na własnym komputerze.
**Sekcja uzupełniona jako efekt zadania Tydzień 05 wg Harmonogramu Zajęć na WIKAMP KSR.**

## 5. Wyniki klasyfikacji dla różnych parametrów wejściowych

Wstępne wyniki miary Accuracy dla próbnych klasyfikacji na ograniczonym zbiorze tekstów (podać parametry i kryteria wyboru wg punktów 3.-8. z opisu Projektu 1.). **Sekcja uzupełniona jako efekt zadania Tydzień 05 wg Harmonogramu Zajęć na WIKAMP KSR.**

## 6. Dyskusja, wnioski, sprawozdanie końcowe

Wyniki kolejnych eksperymentów wg punktów 2.-8. opisu projektu 1. Wykresy i tabele obowiązkowe, dokładnie opisane w „captions" (tytułach), konieczny opis osi i jednostek wykresów oraz kolumn i wierszy tabel.

**Ewentualne wyniki realizacji punktu 9. opisu Projektu 1., czyli „na ocenę 5.0" i ich porównanie do wyników z części obowiązkowej**.Dokładne interpretacje uzyskanych wyników w zależności od parametrów klasyfikacji opisanych w punktach 3.-8 opisu Projektu 1. Szczególnie istotne są wnioski o charakterze uniwersalnym, istotne dla podobnych zadań. Omówić i wyjaśnić napotkane problemy (jeśli były). Każdy wniosek/problem powinien mieć poparcie w przeprowadzonych eksperymentach (odwołania do konkretnych wyników: wykresów, tabel).
Dla końcowej oceny jest to najważniejsza sekcja sprawozdania, gdyż prezentuje poziom zrozumienia rozwiązywanego problemu.

** Możliwości kontynuacji prac w obszarze systemów rozpoznawania, zwłaszcza w kontekście pracy inżynierskiej, magisterskiej, naukowej, itp. **

**Sekcja uzupełniona jako efekt zadania Tydzień 06 wg Harmonogramu Zajęć na WIKAMP KSR.**

## 7. Braki w realizacji projektu 1.

Wymienić wg opisu Projektu 1. wszystkie niezrealizowane obowiązkowe elementy projektu, ewentualnie podać merytoryczne (ale nie czasowe) przyczyny tych braków.

## Literatura


[1] Zbiór 21578 sklasyfikowanych tekstów. `http://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection`

[2] `https://www.nature.com/articles/s41598-022-10358-x`

[3] Materiał wideo objaśniający algorytm $k$-nn. `https://www.youtube.com/watch?v=IPqZKn_cMts`

[4] R. Tadeusiewicz: Rozpoznawanie obrazów, PWN, Warszawa, 1991.

[5] A. Niewiadomski, Methods for the Linguistic Summarization of Data: Applications of Fuzzy Sets and Their Extensions, Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2008.

Literatura zawiera wyłącznie źródła recenzowane i/lub o potwierdzonej wiarygodności, możliwe do weryfikacji i cytowane w sprawozdaniu.