Rozdział 1

Odległości na przestrzeni ciągów znaków

Rozdział 2

Analiza skupień metodą k-średnich

Analiza skupień polega na wyróżnieniu w zbiorze ustalonej liczby rozłącznych skupień obserwacji w jakimś sensie do siebie podobnych, równocześnie zachowując maksymalne zróżnicowanie obserwacji pomiędzy poszczególnymi podzbiorami [8]. W niniejszym rozdziale przedstawimy analizę skupień metodą k-średnich w trzech odsłonach: przedstawimy metodę wsadową, przy użyciu stochastycznego spadku gradientu oraz metodę pośrednią, tzw. miniwsadową.

2.1. Metoda k-średnich

[TO DO: DOROBIC PRZYKLADY:
- (2.1, 2.1) POKAZAC SKUPIENIA OTRZYMANE PRZY UZYCIU 3/4 ALGORYTMOW NP. NA IRISIE
- (2.3) NARYSOWAC JAKIS DENDROGRAM
- (2.3) POKAZAC ODMIENNOSCI NAJBLIZSZEGO, NAJDALSZEGO, SREDNIEGO
- (2.4) PRZYKLADY JAKOSCI NA PODSTAWIE JAKICHS LOSOWYCH KLAS
]

Rozważmy przestrzeń euklidesową $\mathbb{R}^p$ i niech będzie dana liczba skupień k. Wówczas zadanie znalezienia skupień o wyżej wymienionych własnościach można sprowadzić do dobrze określonego zadania optymalizacji. Weźmy próbę n-elementową obserwacji $\mathbf{x}_i,\ i = 1, \ldots, n$ o wartościach w $\mathbb{R}^p$. Suma kwadratów odległości między obserwacjami próby wynosi [8]

$$T = \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2, \tag{2.1}$$

gdzie $\|\cdot\|_2$ oznacza normę euklidesową. Niech funkcja $C : \{1, \ldots, n\} \to \{1, \ldots, k\}$ oznacza przydzielenie danej obserwacji danemu skupieniu, tzn. jeśli $C(i) = l$, to oznacza, że $\mathbf{x}_i$ należy do l-tego skupienia. Zakładając, że dokonano podziału próby na k podzbiorów, można całkowitą sumę kwadratów rozłożyć na sumę kwadratów odległości między obserwacjami z tego samego skupienia oraz na sumę kwadratów odległości między obserwacjami z różnych

skupień [8]:

$$T = W + B = \frac{1}{2}\sum_{l=1}^{k}\sum_{C(i)=k}\sum_{C(j)=k}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + \frac{1}{2}\sum_{l=1}^{k}\sum_{C(i)=k}\sum_{C(j)\neq k}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \tag{2.2}$$

Mając tak sformułowany rozkład sumy T widzimy, że zmieniając podział punktów na skupienia zmienia się zarówno suma W, jak i B. Można więc sformułować problem analizy skupień jako zadanie minimalizacji sumy W lub, równoważnie, maksymalizacji sumy B. Maksymalizacja B to po prostu maksymalizacja rozproszenia punktów z różnych podzbiorów, co jest równoznaczne z minimalizacją rozproszenia punktów z tego samego skupienia. Stąd, rozwiązaniem problemu analizy skupień jest dokonanie takiego podziału próby, aby zminimalizować sumę W. Ze względu na złożoność obliczeniową, niemożliwe jest bezpośrednie rozwiązanie tego problemu [8].

Przez n_l oznaczmy liczność l-tego skupienia i niech $\mathbf{m}_l = \frac{1}{n_l}\sum_{C(i)=l}\mathbf{x}_i$ oznacza wektorową średnią obserwacji z l-tego skupienia. Łatwo zauważyć, że [8]

$$W = \frac{1}{2}\sum_{l=1}^{k}\sum_{C(i)=l} n_l\|\mathbf{x}_i - \mathbf{m}_l\|_2^2 \tag{2.3}$$

Średnie $\mathbf{m}_l$, $l = 1, \ldots, k$ nazywamy *środkami skupień*. Równanie 2.3 można uprościć do następującej postaci, która w praktyce jest łatwa w optymalizacji:

$$\widetilde{W} = \frac{1}{2}\sum_{l=1}^{k}\sum_{C(i)=l}\|\mathbf{x}_i - \mathbf{m}_l\|_2^2 = \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{m}_{C(i)}\|_2^2 \tag{2.4}$$

Algorytmy, które rozwiązują problem minimalizacji sumy 2.4, znane są pod nazwą *metody k-średnich*.

Zauważmy, że wszystkie znane i stosowane wersje algorytmów k-średnich są zbieżne. W tym celu wystarczy, aby w każdej iteracji algorytmu suma $\widetilde{W}$ była zmniejszona. W przeciwnym przypadku algorytm zostaje zatrzymany. Warto zauważyć jednak, że rozwiązanie takie może nie prowadzić do rozwiązania optymalnego, tj. algorytm może zatrzymać działanie w minimum lokalnym wartości $\widetilde{W}$, zamiast zbiec do minimum globalnego. Stąd zaleca się wielokrotne stosowanie danego algorytmu z różnymi warunkami początkowymi [8].

2.2. Algorytmy

2.2.1. Algorytm wsadowy

Algorytm *wsadowy* (ang. *batch algorithm*) zostaje zainicjalizowany przez losowe wyznaczenie k punktów jako początkowe środki skupień. Dalej następuje przydzielenie wszystkich punktów próby do najbliższego skupienia, a następnie przeliczenie środków jako średniej ze wszystkich obserwacji w danym skupieniu. Procedura ta jest powtarzana aż do ustabilizowania się algorytmu, tj. do momentu aż żaden punkt próby nie zmieni skupienia [20].

Algorytm wsadowy jest najbardziej popularnym i najczęściej stosowanym algorytmem, gdyż jest szybki i zazwyczaj daje dobre rezultaty. Jednakowoż jeśli liczba obserwacji w zbiorze

Algorithm 1 Algorytm wsadowy k-średnich

given: k, data set X
initialize randomly $m_l,\ \forall l = 1, ..., k$
repeat
  for $i = 1, \ldots, n$ **do**
    $C(i) = \arg\min_l \|\mathbf{x}_i - \mathbf{m}_l\|_2^2$
  end for
  for $l = 1, \ldots, k$ **do**
    $\mathbf{m}_l = \frac{1}{n_l} \sum_{C(i)=l} \mathbf{x}_i$
  end for
until convergence

jest bardzo duża, to obliczanie średnich z obserwacji we wszystkich skupieniach jest bardzo kosztowne obliczeniowo, zbiegając w czasie $O(knp)$, gdzie p to liczba zmiennych. Stąd Bottou i Bengio [3] zaproponowali algorytm oparty na stochastycznym spadku gradientu.

2.2.2. Algorytmy oparte na spadku gradientu

Algorytmy oparte na spadku gradientu są często stosowane np. w regresji liniowej [2]. Idea polega na szukaniu minimum z danej funkcji kosztu, w kolejnych krokach algorytmu aktualizując zmienną, w kierunku, w którym spadek gradientu był największy. Każda aktualizacja zależy od parametru, zwanego *parametrem uczenia*, który musi być odpowiednio dobrany. W niniejszym podrozdziale opiszemy trzy algorytmy oparte na spadku gradientu.

Mając daną funkcję kosztu $\widetilde{W} = \widetilde{W}(\mathbf{m}, \mathbf{x}_i) = \frac{1}{2} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{m}_{C(i)}\|_2^2$, możemy znaleźć minimum używając tzw. *spadku gradientu*. W każdej iteracji algorytmu uaktualniamy wektor $\mathbf{m}$ na podstawie gradientu $\widetilde{W}(\mathbf{m}, \mathbf{x}_i)$:

$$\mathbf{m}_l^{(t+1)} = \mathbf{m}_l^{(t)} + \gamma \sum_{i=1}^{n} \frac{\partial \widetilde{W}(\mathbf{m}, \mathbf{x}_i)}{\partial \mathbf{m}} \tag{2.5}$$

gdzie γ jest odpowiednio dobranym *parametrem uczenia*, a t oznacza iterację algorytmu [2]. Parametrem uczenia, które daje najlepsze rezultaty dla algorytmu k-średnich jest $\frac{1}{n_{C(i)}}$. Stąd też algorytm *wsadowego spadku gradientu*, w każdej iteracji algorytmu aktualizuje wektor $\mathbf{m}$ następująco [3]:

$$\mathbf{m}_l^{(t+1)} = \mathbf{m}_l^{(t)} + \sum_{C(i)=l} \frac{1}{n_l} (\mathbf{x}_i - \mathbf{m}_l^{(t)}) \tag{2.6}$$

Algorytm *stochastycznego spadku gradientu*, ozn. SGD (ang. *stochastic gradient descent*), jest daleko idącym uproszczeniem. Zamiast liczyć gradient z $\widetilde{W}(\mathbf{m}, \mathbf{x}_i)$ wprost, każda iteracja estymuje gradient na podstawie *jednej losowo wybranej* obserwacji $\mathbf{x}_i$ [2]:

$$\mathbf{m}_l^{(t+1)} = \mathbf{m}_l^{(t)} + \gamma \frac{\partial \widetilde{W}(\mathbf{m}, \mathbf{x}_i)}{\partial \mathbf{m}} \tag{2.7}$$

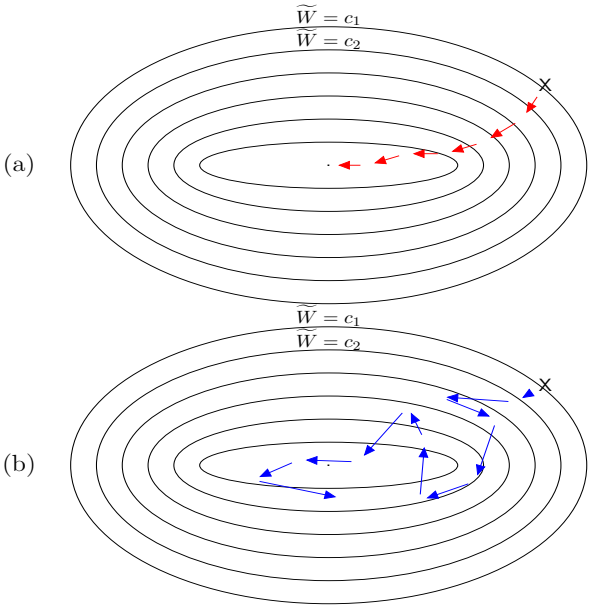
gdzie γ jest odpowiednio dobranym parametrem uczenia. Tak samo jak w przypadku algorytmu wsadowego, parametrem uczenia, które daje najlepsze rezultaty jest $\frac{1}{n_{C(i)}}$. Stąd też

algorytm stochastycznego spadku gradientu w każdej iteracji algorytmu aktualizuje wektor $\mathbf{m}$ następująco [3]:

$$n_l^{(t+1)} = n_l^{(t)} + \begin{cases} 1, & \text{gdy } l = C(i1) \\ 0, & \text{wpp.} \end{cases} \tag{2.8}$$

$$\mathbf{m}_l^{(t+1)} = \mathbf{m}_l^{(t)} + \begin{cases} \frac{1}{n_l}(\mathbf{x}_i - \mathbf{m}_l^{(t)}), & \text{gdy } l = C(i) \\ 0, & \text{wpp.} \end{cases} \tag{2.9}$$

Algorytm SGD opiera się na przeliczaniu średniej po każdym przydzieleniu obserwacji do skupienia, choć z powodu stochastycznego szumu, takie rozwiązanie może nie prowadzić do lokalnego minimum, a jedynie w jego „pobliże". Na rys. 2.1 przedstawiono przykładową drogę aktualizacji parametrów. Kolejne elipsy oznaczają stałą wartość funkcji kosztu $\widetilde{W}(\mathbf{m}, \mathbf{x}_i)$ w zależności od wartości zmiennej $\mathbf{m}$, a centrum (środek?) oznacza (lokalne) minimum tej funkcji. Jeśli algorytm rozpoczyna działanie w punkcie oznaczonym przez X, to w przypadku algorytmu wsadowego spadku gradientu, w kolejnych iteracjach zmienna $\mathbf{m}$ zmienia swoją wartość, przybliżając się do (lokalnego) minimum funkcji $\widetilde{W}$. Natomiast algorytm stochastycznego spadku gradientu w każdej iteracji przybliża się w stronę minimum w sposób losowy, tzn. może nigdy nie osiągnąć właściwego minimum, a jedynie „krążyć" wokół niego.



Rysunek 2.1: Przykładowa droga wsadowego spadku gradientu (rys. a) i stochastycznego spadku gradientu (rys. b).

Algorytm oparty na stochastycznym spadku gradientu rozpoczyna się tak samo jak algorytm wsadowy, tj. inicjalizacją losowych k środków skupień. Następnie zbiór obserwacji jest mieszany i obserwacje po kolei są przydzielane do najbliższego skupienia. Środki skupień przeliczane są po każdym przydzieleniu punktu do skupienia. Procedura ta powtarzana jest do uzyskania zbieżności [3]. Algorytm ten jest dużo szybszy od dwóch wcześniejszych, kosztem dokładności rozwiązania [2].

Algorytm *mini-wsadowy* (ang. *mini-batch k-means*) jest połączeniem dwóch poprzednich algorytmów, tj. w każdej iteracji przydzielanych do najbliższego skupienia jest b losowo wybranych obserwacji, po czym następuje przeliczenie środków skupień [13]. Algorytm ten jest

Algorithm 2 Algorytm SGD k-średnich

given: k, data set X, iterations t
initialize randomly $m_l,\ \forall l = 1, ..., k$
initialize $n_l = 0,\ \forall l = 1, ..., k$
repeat
    randomly pick one observation $\mathbf{x}_i$ from X
    $C(i) = \arg\min_l \|\mathbf{x}_i - \mathbf{m}_l\|_2^2$
    $n_{C(i)} = n_{C(i)} + 1$
    $\mathbf{m}_{C(i)} = \mathbf{m}_{C(i)} + \frac{1}{n_{C(i)}} \|\mathbf{x}_i - \mathbf{m}_{C(i)}\|_2$
until convergence

porównywalnie szybki do algorytmu SGD, osiągając przy tym lepsze rezultaty z powodu mniejszego stochastycznego szumu.

Algorithm 3 Algorytm mini-wsadowy k-średnich

given: k, data set X, iterations t, mini-batch size b
initialize randomly $m_l,\ \forall l = 1, ..., k$
initialize $n_l = 0,\ \forall l = 1, ..., k$
repeat
    $B = b$ observations randomly picked from X
    for $i : \mathbf{x}_i \in B$ **do**
        $C(i) = \arg\min_l \|\mathbf{x}_i - \mathbf{m}_l\|_2^2$
    end for
    for $i : \mathbf{x}_i \in B$ **do**
        $n_{C(i)} = n_{C(i)} + 1$
        $\mathbf{m}_{C(i)} = \mathbf{m}_{C(i)} + \frac{1}{n_{C(i)}} \|\mathbf{x}_i - \mathbf{m}_{C(i)}\|_2$
    end for
until convergence

[CZY PODAWAC TUTAJ PRZYKLAD + RYSUNKI Z [13] O TYM ZE MINI-BATCH JEST TAKI SUPER W POR. Z INNYMI??]

2.3. Metody hierarchiczne

Metody hierarchiczne to zbiór algorytmów analizy skupień, które nie wymagają znajomości liczby skupień. W niniejszym podrozdziale przedstawimy pokrótce schemat ich działania w dowolnej przestrzeni obserwacji.

Powyżej przedstawiliśmy algorytm k-średnich, dzielący zbiór z przestrzeni euklidesowej na podzbiory punktów podobnych do siebie, tj. mieliśmy do czynienia ze zmiennymi liczbowym. Problem ten można uogólnić dla obserwacji z dowolnej przestrzeni. Po pierwsze odległość euklidesową z równania 2.4 można zamienić na dowolną funkcję odmienności d między obserwacjami z danej przestrzeni. Po drugie trzeba zastąpić średnie skupień $\mathbf{m}_l, l = 1, \ldots, k$ inną wartością wektorową, która miałaby sens w przypadku np. atrybutów jakościowych. Wartość ta to punkt ze zbioru obserwacji, który minimalizuje sumę odległości między nim samym, a

pozostałymi punktami ze skupienia [8]:

$$\mathbf{m}_l = \min_{\mathbf{y} \in X} \sum_{i=C(l)} d(\mathbf{x}_i, \mathbf{y}), \tag{2.10}$$

gdzie X to zbiór obserwacji.

Przejdźmy do metod hierarchicznych analizy skupień. Jak wspomniano wcześniej, nie wymagają one specyfikowania liczby podzbiorów. Zamiast tego trzeba zdefiniować miarę odmienności (rozłącznych) zbiorów obserwacji, opartą na odmienności pojedynczych punktów w tych zbiorach. Jak sugeruje nazwa, metody te konstruują hierarchiczną reprezentację, w której skupienie na każdym poziomie hierarchii powstaje poprzez połączenie skupień z najbliższego niższego poziomu. Na najniższym poziomie znajdują się skupienia złożone z pojedynczych obserwacji. Najwyższy poziom to skupienie zawierające wszystkie obserwacje ze zbioru [6].

Metody hierarchiczne można podzielić na dwie grupy: aglomeracyjne oraz dzielące. W tej pierwszej, na początku tworzy się tyle skupień ile jest obserwacji w zbiorze, traktując każdą obserwację jako osobne skupienie. Następnie w każdym kroku łączona jest ta para podzbiorów, które są od siebie najmniej odmienne. W ten sposób na kolejnym poziomie otrzymujemy (przynajmniej) o jedno skupienie mniej. Procedura łącząca skupienia najmniej odmienne trwa nadal, w każdym kolejnym kroku zmniejszając liczbę skupień. W ostatnim kroku algorytmu otrzymujemy jedno duże skupienie zawierające wszystkie obserwacje z próby [6, 8].

Metoda dzieląca działa odwrotnie: zaczynamy od jednego skupienia zawierającego cały zbiór. Następnie w każdym kroku algorytmu jedno ze skupień dzielone jest na dwa skupienia, w których odmienność jest największa. W ten sposób na kolejnym poziomie otrzymujemy o jedno skupienie więcej. Procedura dzieląca skupienia najbardziej odmienne trwa nadal, w każdym kolejnym kroku otrzymując coraz więcej skupień. W ostatnim kroku algorytmu dostajemy podzbiory jednoelementowe, dostając n rozłącznych skupień (gdzie n to liczba obserwacji). Warto przy tym zauważyć, że algorytm ten jest znacznie bardziej złożony obliczeniowo niż algorytm aglomeracyjny [6, 8].

Zastanówmy się teraz w jaki sposób mierzyć odmienność między podzbiorami. Ponieważ metoda dzieląca jest o wiele bardziej złożona obliczeniowo, skupimy się na metodzie aglomeracyjnej, jako tej częściej stosowanej w praktyce. Odmienność między skupieniami można definiować na różne sposoby, jednak literatura podaje zazwyczaj trzy najbardziej popularne, tj. odmienność najbliższego sąsiada, odmienność najdalszego sąsiada oraz średnią odmienność. Oznaczmy przez D_{ij} odmienność między skupieniem i-tym a j-tym, których liczności wynoszą odpowiednio n_i, n_j[6, 8].

Definicja 2.1. Odmienność najbliższego sąsiada *(ang.* single linkage dissimilarity*) między skupieniem i-tym a j-tym, definiujemy jako najmniejszą spośród wszystkich możliwych odmienności między parami obserwacji z i-tego i j-tego skupienia:*

$$D_{ij} = \min_{\mathbf{x}_a \in C(i), \mathbf{x}_b \in C(j)} d(\mathbf{x}_a, \mathbf{x}_b).$$

Definicja 2.2. Odmienność najdalszego sąsiada *(ang.* complete linkage dissimilarity*) między skupieniem i-tym a j-tym, definiujemy jako największą spośród wszystkich możliwych odmienności między parami obserwacji z i-tego i j-tego skupienia:*

$$D_{ij} = \max_{\mathbf{x}_a \in C(i), \mathbf{x}_b \in C(j)} d(\mathbf{x}_a, \mathbf{x}_b).$$

Definicja 2.3. Odmienność średnią *(ang.* average linkage dissimilarity*) między skupieniem i-tym a j-tym, definiujemy jako średnią odmienności między parami obserwacji z i-tego i j-tego skupienia:*

$$D_{ij} = \frac{1}{n_i n_j} \sum_{\mathbf{x}_a \in C(i), \mathbf{x}_b \in C(j)} d(\mathbf{x}_a, \mathbf{x}_b).$$

Odmienność najbliższego sąsiada wymaga żeby jedna odmienność $d(\mathbf{x}_a, \mathbf{x}_b)$, gdzie $\mathbf{x}_a \in C(i)$, $\mathbf{x}_b \in C(j)$, miała małą wartość, aby uznać dwa podzbiory za bliskie sobie, bez względu na odmienności innych obserwacji z tych dwóch grup. Metoda ta będzie zatem mieć skłonność do łączenia, skupień połączonych przez szereg bliskich obserwacji pośrednich. Takie zjawisko nazywane jest *efektem łańcuchowym* i uważane jest za wadę tej metody. Skupienia otrzymane w wyniku jej działania często nie są zwarte, jako że podobieństwo skupień określone jest na podstawie dwóch najbliższych obserwacji [6]. Odmienność najbliższego sąsiada daje zazwyczaj skupienia wąskie i wydłużone [8].

Metoda odmienności najdalszego sąsiada ma działanie odwrotne. Dwa skupienia są do siebie podobne, wtedy i tylko wtedy, gdy wszystkie obserwacje z ich połączenia są dość podobne. Jednakowoż, metoda ta może nie zachowywać własności „bliskości" dwóch podzbiorów, tj. obserwacje przypisane do danego skupienia mogą znajdować się o wiele bliżej obserwacji z innego podzbioru, niż do punktów ze swojego skupienia [6]. W wyniku jej działania otrzymujemy podzbiory o kulistym kształcie [8].

Metoda średniej odmienności jest kompromisem pomiędzy dwoma powyższymi. Próbuje ona dać skupienia relatywnie zwarte i relatywnie oddalone od siebie [6]. Podobnie jak metoda odmienności najdalszego sąsiada, daje ona skupienia o kształcie kulistym [8].

Jeśli odmienności poszczególnych obserwacji mają silną tendencję do skupiania się, przy czym każde skupienie jest zwarte i dobrze odseparowane, to wszystkie trzy metody dadzą podobne rezultaty [6].

2.4. Metody oceny jakości podziału na skupienia

[TUTUAJ KORZYSTALAM MOCNO Z DOKUMENTACJI SCIKIT: `http://scikit-learn.org/stable/modules/clustering.html#clustering-evaluation` ZAWRZEC TO GDZIES??]

Dokonawszy podziału zbioru na skupienia, należy ocenić jakość zastosowanego algorytmu. Ocena skuteczności działania algorytmów analizy skupień nie należy do zadań tak prostych jak np. ocena modelu klasyfikacji pod nadzorem. W szczególności żadna metoda ocena nie powinna brać pod uwagę wartości etykiety skupienia, ale powinna sprawdzać, czy zbiór danych jest dobrze podzielony, tzn. czy obserwacje w poszczególnych skupieniach są do siebie „podobne", a obserwacje z różnych skupień – „niepodobne", zgodnie z przyjętą metryką podobieństwa. W niniejszych podrozdziale przedstawimy kilka miar oceny jakości podziału na skupienia.

Przyjmujemy następującego założenia: Niech K i C oznaczają dwa różne podziały n-elementowego zbioru X na skupienia. Zazwyczaj K oznacza podział uzyskany przy pomocy algorytmu dzielącego zbiór, a C jest zbiorem prawdziwych klas, do których należą obserwacje, choć K i C mogą również oznaczać dwa niezależne podziały uzyskane przy pomocy różnych algorytmów.

2.4.1. Skorygowany indeks Randa [Adjusted Rand Index [JAK TO PRZETŁUMACZYĆ ???]]

Niech a_1 oznacza liczbę par elementów z X, które mają wspólne skupienie zarówno w K, jak i w C, natomiast przez a_0 oznaczmy liczbę par elementów z X, które mają zostały przypisane do innych skupień zarówno w K, jak i w C. Elementy, które spełniają jeden z powyższych warunków oznaczają zgodność podziałów K i C. Możemy wówczas zdefiniować *indeks Randa* [10]:

$$\mathrm{RI} = \frac{a_0 + a_1}{\binom{n}{2}} \tag{2.11}$$

Wartości indeksu Randa znajdują się w przedziale $[0, 1]$. W praktyce jednak RI leży często pomiędzy 0.5, a 1. Co więcej, miara ta nie gwarantuje, że losowy podział zbioru da wartość indeksu bliską zeru. Z tego powodu RI jest zazwyczaj używana w skorygowanej formie [7]:

$$\mathrm{ARI} = \frac{2(a_0 a_1 - b_0 b_1)}{(a_0 + b_0)(b_0 + a_1) + (a_0 + b_1)(b_1 + a_1)}, \tag{2.12}$$

gdzie b_0 oznacza liczbę par elementów z X, które należą do tego samego skupienia w K, ale do różnych skupień w C, natomiast b_1 oznacza liczbę par elementów z X, które należą do różnych skupień w K, ale do tego samego skupienia w C.

Zalety:

- Losowe przyporządkowanie do skupień daje wartość ARI bliską zeru dla dowolnej liczby skupień i obserwacji w zbiorze.
- Miara ARI daje wartości z przedziału $[-1, 1]$, gdzie -1 oznacza niezależne przyporządkowanie, natomiast 1 oznacza pełną zgodność.
- Brak założeń o strukturze skupienia: przy pomocy ARI można porównywać podział na skupienia uzyskany przy pomocy różnych algorytmów, które mają różne założenia o strukturze skupienia.

Wady:

- W przypadku sprawdzenia jakości działania jednego algorytmu, wymagana jest znajomość prawdziwego podziału zbioru, co w praktyce rzadko występuje lub wymaga ręcznego podziału zbioru.

2.4.2. Jednorodność, zupełność oraz miara V

Niech C oznacza zbiór prawdziwych klas, do których należą obserwacje. Mówimy, że podział zbioru jest *jednorodny*, jeśli wszystkie skupienia zawierają jedynie obserwacje z jednej klasy. Podział zbioru jest *zupełny*, jeśli wszystkie obserwacje z danej klasy są w tym samym skupieniu. Jednorodność i zupełność podziału może być często w opozycji do siebie, tzn. gdy jednorodność rośnie, to zupełność zazwyczaj maleje i odwrotnie. Przykładowo, rozważmy dwa skrajne podziały. W przypadku pierwszego, gdy przydzielamy wszystkie obserwacje do jednego skupienia, to dostajemy idealną zupełność – wszystkie elementy z jednej klasy należą do tego samego skupienia. Jednakowoż, podział taki jest tak *nie*jednorodny, jak to tylko możliwe, skoro wszystkie klasy znajdują się w jednym skupieniu. Z drugiej strony, rozważmy przydzielenie każdej obserwacji do osobnego skupienia. W tym przypadku, podział jest idealnie jednorodny – każde skupienie zawiera jedynie obserwacje z jednej klasy. Jednak w

terminach zupełności, taki podział bardzo słaby, chyba że rzeczywiście każda klasa zawiera jeden element. Miara V jest ważoną średnią harmoniczną dwóch powyższych miar [11].

Jednorodność. Żeby spełnić kryteria, jak musi spełniać podział jednorodny, każde ze skupień musi zawierać tylko i wyłącznie te obserwacje, które należą do jednej klasy. To znaczy, że rozkład klas w każdym ze skupień powinien skośny i zawierać tylko jedną klasę, tj. entropia powinna wynosić zero. Aby ustalić jak blisko od idealnego znajduje się dany podział, sprawdzamy warunkową entropię rozkładu klas pod warunkiem zaproponowanego podziału. W idealnie jednorodnym podziale, tak wartość, tj. $H(C|K)$ wynosi zero. Jednak w nieidealnej sytuacji, wartość ta zależy od wielkości zbioru i rozkładu klas. Stąd, zamiast badać warunkową entropię, normalizujemy tę wartość przez maksymalną redukcję entropii jaką informacja o podziale może przynieść, tj. $H(C)$ [11].

Zauważmy, że $H(C|K)$ jest maksymalne (i równe $H(C)$), kiedy podział na skupienia nie wnosi żadnej nowej informacji – rozkład klas w każdym skupieniu jest równy ogólnemu rozkładowi klas. $H(C|K)$ jest równy zeru, gdy każde skupienie zawiera jedynie obserwacje z jednej klasy, tj. w przypadku idealnie jednorodnego podziału. W przypadku zdegenerowanym, gdy $H(C) = 0$, kiedy istnieje tylko jedna klasa, definiujemy jednorodność jako równą 1. W idealnie jednorodnym rozwiązaniu, taka normalizacja, tj. $\frac{H(C|K)}{H(C)}$, wynosi zero. Stąd, aby utrzymać konwencję, że wartość 1 jest pożądana, a wartość 0 jest niepożądana, definiujemy jednorodność jako [11]:

$$h = \begin{cases} 1, & \text{gdy } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)}, & \text{w przeciwnym przypadku} \end{cases} \tag{2.13}$$

gdzie

$$H(C) = -\sum_{c=1}^{|C|} \frac{n_c}{n} \cdot \log \frac{n_c}{n},$$

$$H(C|K) = -\sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{n_{c,k}}{n} \cdot \log \frac{n_{c,k}}{n_k},$$

gdzie n_c oznacza liczność klasy c, $c = \{1, \ldots, |C|\}$, a $n_{c,k}$ oznacza liczbę elementów, która należy do klasy c i skupienia k, $k = \{1, \ldots, |K|\}$.

Zgodność. Miara zgodności jest symetryczna do miary jednorodności. Aby spełnić warunki zgodności, podział zbioru musi przydzielić wszystkie obserwacje z jednej klasy, do tego samego skupienia. Żeby policzyć zgodność, badamy rozkład przypisanych skupień w obrębie jednej klasy. W idealnie zgodnym podziale, każdy z rozkładów będzie skośny i będzie zawierać jedynie tylko jedno skupienie. Możemy oszacować poziom skośności, licząc warunkową entropię zaproponowanego podziału pod warunkiem klas, tj. $H(K|C)$. W idealnie zgodnym rozwiązaniu $H(K|C) = 0$. Jednak w najgorszym możliwym przypadku, gdy wszystkie obserwacje z tej samej klasy są we wszystkich skupieniach, rozkład tych ostatnich jest równy rozkładowi rozmiarów klastrów, $H(K|C)$ jest maksymalne i równe $H(K)$. W końcu, w zdegenerowanym przypadku, kiedy $H(K) = 0$, kiedy mamy tylko jedno skupienie, definiujemy zgodność jako równą 1. Stąd, robiąc symetryczne wyliczenie do poprzedniego, definiujemy zgodność jako [11]:

$$c = \begin{cases} 1, & \text{gdy } H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)}, & \text{w przeciwnym przypadku} \end{cases} \tag{2.14}$$

gdzie

$$H(K) = -\sum_{k=1}^{|K|} \frac{n_k}{n} \cdot \log \frac{n_k}{n},$$
$$H(K|C) = -\sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{n_{c,k}}{n} \cdot \log \frac{n_{c,k}}{n_c}$$

Miara V. Mając zdefiniowane miary jednorodności i zgodności, możemy wyliczyć miarę V jako ważoną średnią harmoniczną tych dwóch miar [11]:

$$\mathrm{V}_\beta = \frac{(1+\beta) \cdot h \cdot c}{(\beta \cdot h) + c} \tag{2.15}$$

Zauważmy, że jeśli β jest mniejsza niż 1, jednorodność ma większą wagę niż zgodność. Często za β przyjmuje się po prostu 1, dając równą wagę obu miarom.

Warto zwrócić uwagę na fakt, że jednorodność, zgodność oraz miara V są niezależne od liczby klas, skupień, liczby obserwacji oraz użytego algorytmu. Stąd miary te mogą być używane do porównania każdego algorytmu dzielącego na skupienia, niezależnie od powyższych parametrów. Co więcej, wyliczając zarówno jednorodność, jak i zgodność, może zostać otrzymana bardziej precyzyjna ocena jakości podziału skupień.

Zalety:

- Miary te dają wartości z przedziału $[0, 1]$, gdzie 0 oznacza najgorszy możliwy przypadek, natomiast 1 to bardzo dobre rozwiązanie.
- Intuicyjna interpretacja: podział z niską wartością miary V może zostać oceniony w terminach jednorodności i zgodności, aby mieć lepsze pojęcie o błędach jakich dokonał algorytm.
- Brak założeń o strukturze skupienia: przy pomocy powyższych miar można porównywać podział na skupienia uzyskany przy pomocy różnych algorytmów, które mają różne założenia o strukturze skupienia.

Wady:

- Powyższe miary nie są znormalizowane pod względem losowego przypisania do skupienia. Oznacza to, że w zależności od liczby obserwacji, podziału na skupienia i klasy, losowy przydział do skupień nie zawsze da takie same wartości jednorodności, zgodności oraz miary V. W szczególności, losowe przyporządkowanie może nie dać wartości powyższych miar równych zero, zwłaszcza gdy liczba skupień jest duża. Problem ten może być bezpiecznie zignorowany, gdy liczba obserwacji jest większa od tysiąca, a liczba skupień mniejsza niż 10. Dla mniejszej próbki i większej liczby skupień, bezpieczniej jest używać miary ARI.
- W przypadku sprawdzenia jakości działania jednego algorytmu, wymagana jest znajomość prawdziwego podziału zbioru, co w praktyce rzadko występuje lub wymaga ręcznego podziału zbioru.

2.4.3. Miara silhouettes

Sylwetki obserwacji (czy też *silhouettes*) są użyteczne, gdy odległości są określone na relatywnej skali (ratio scale??) oraz gdy pożądane są wyraźnie odseparowane skupienia. Aby

skonstruować sylwetkę obserwacji potrzebne są dwie rzeczy: podział zbioru na skupienia C oraz miarę odległości d pomiędzy obserwacjami [12].

Weźmy każdą obserwację z próbki $\mathbf{x}_i$ i oznaczmy przez $C(i)$ skupienie, do którego należy. Średnią odmiennością $\mathbf{x}_i$ od swojego skupienia nazywamy [12]:

$$a(\mathbf{x}_i) = \frac{\sum_{u \in C(i)} d(\mathbf{x}_i, u)}{n_{C(i)}}$$

Średnią odmiennością $\mathbf{x}_i$ od skupienia J nazywamy:

$$c(\mathbf{x}_i, J) = \frac{\sum_{u \in J} d(\mathbf{x}_i, u)}{n_J}$$

Wówczas możemy wyliczyć odległość obserwacji $\mathbf{x}_i$ od najbliższego skupienia innego niż $C(i)$:

$$b(\mathbf{x}_i) = \min_{j \neq i} c(\mathbf{x}_i, J)$$

Skupienie L, dla którego minimum jest osiągnięte (tj. $b(\mathbf{x}_i) = c(\mathbf{x}_i, L)$) nazywamy sąsiadem obserwacji $\mathbf{x}_i$. Jest ono drugim najlepszym wyborem dla tej obserwacji. Stąd, znajomość sąsiada jest bardzo użyteczna, gdyż mówi o tym które skupienie zostałoby wybrane, gdyby nie zostało nim skupienie $C(i)$. Sylwetka obserwacji jest definiowana następująco [12]:

$$s(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{\max b(\mathbf{x}_i), a(\mathbf{x}_i)}$$

Sylwetki otrzymujemy dla każdej obserwacji z osobna, ale najczęściej interesująca jest miara zdefiniowana dla całego zbioru, stąd sylwetką zbioru nazywamy średnią z sylwetek po wszystkich obserwacjach:

$$sil = \sum_{i=1}^{n} s(\mathbf{x}_i)$$

Zalety:

- Miara ta daje wartości z przedziału $[-1, 1]$, gdzie -1 oznacza niepoprawny podział, natomiast 1 to bardzo gęste skupienia. Wartości w okolicy zera sugerują nakładające się skupienia.
- Wartość sylwetki jest wyższa, gdy skupienia są gęste i dobrze odseparowane, co jest jedną z najbardziej pożądanych cech analizy skupień.
- Miara nie wymaga znajomości prawdziwych klas, na jakie można podzielić zbiór.

Wady:

- Miara silhouette przyjmuje w ogólności większe wartości dla wypukłych skupień.

Rozdział 3

Kategoryzacja tematyczna tekstów przy użyciu metryk w przestrzeni ciągów znaków

Wikipedia[1] jest to wielojęzyczna encyklopedia internetowa, która działa w oparciu o zasadę otwartej treści. Portal umożliwia każdemu z użytkowników odwiedzających stronę, edycję i aktualizację treści w czasie rzeczywistym. Wikipedia ma ponad 35.9 miliona artykułów we wszystkich wersjach językowych, w tym prawie 5 milionów w wersji angielskiej i nieco ponad 1.1 miliona artykułów w języku polskim (dane na sierpień 2015) [1].

Każdy artykuł może być edytowany przez dowolnego użytkownika, jak również nowe teksty może stworzyć każda osoba odwiedzająca portal. Przy procesie edycji oraz pisania artykułu obowiązują liczne reguły, między innymi takie jak wskazanie źródeł bibliograficznych, podlinkowanie do innych artykułów oraz nadanie artykułowi kategorii tematycznych [1].

Ta ostatnia zasada może w szczególności przysporzyć nieco kłopotów. Liczba dostępnych kategorii jest bardzo duża (ponad 125 tys. w polskiej wersji językowej), co więcej poziom ich szczegółowości jest zróżnicowany, tzn. mamy kategorie bardzo ogólne (np. *Matematyka*), jak i dość szczegółowe (np. *Działania dwuargumentowe*). Można się spodziewać, że automatyczny podział tekstów na kategorie na podstawie słów, jakie w nich występują, mógłby dać lepszy, bardziej dopasowany do treści, temat. Sprawdzenie jak automatyczny podział tekstów na kategorie tematyczne, przy użyciu występujących w nich słów oraz ich liczności, jest zasadniczym celem tej pracy. Idea działania jest następująca: algorytm analizuje jakie słowa w jakich ilościach występują w danym artykule i następnie przydziela go do grupy artykułów które zawierają takie same i podobne słowa w zbliżonych licznościach. To podejście opiera się na założeniu, że artykuły o podobnej tematyce będą zawierały takie same lub podobne do siebie słowa.

Schemat działania algorytmu jest następujący:

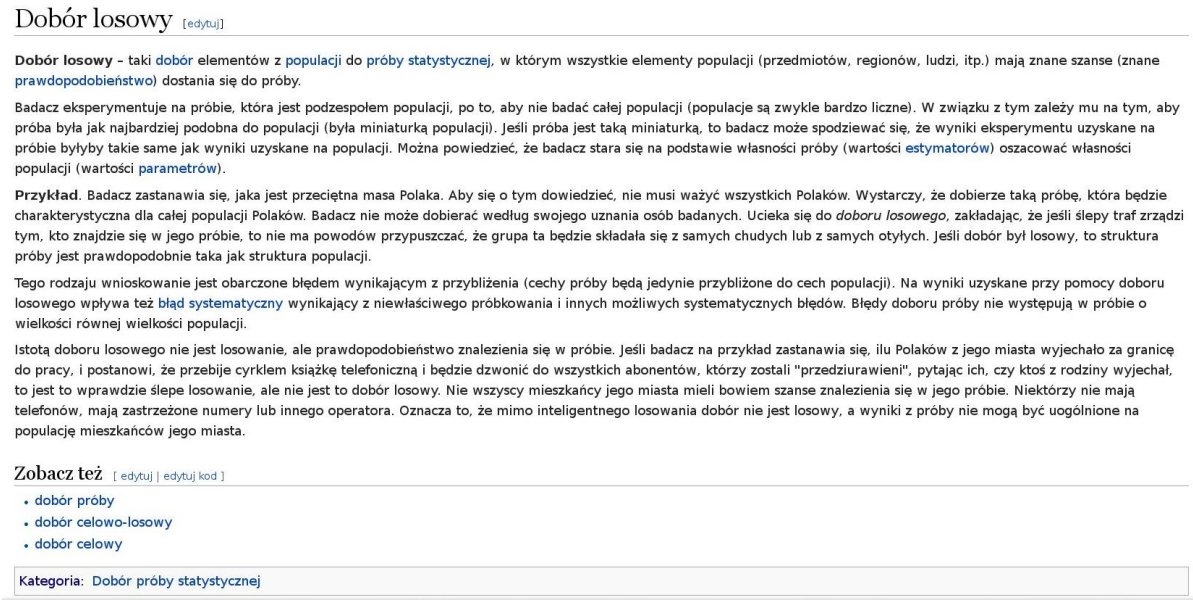
1. Wstępne przetwarzanie danych.
2. Utworzenie skupień „podobnych" słów.
3. Stworzenie macierzy o wymiarach liczba artykułów $\times$ liczba słów, gdzie wartością jest liczność występowania danego słowa w artykule.

[1] `www.wikipedia.org`

4. Użycie algorytmu k-średnich do podzielenia tekstów na skupienia.

3.1. Opis danych

Skuteczność algorytmu automatycznej kategoryzacji tematycznej testowana jest na artykułach polskiej wersji Wikipedii. Omówimy teraz rzeczony zbiór danych.



Dobór losowy [edytuj]

Dobór losowy - taki dobór elementów z populacji do próby statystycznej, w którym wszystkie elementy populacji (przedmiotów, regionów, ludzi, itp.) mają znane szanse (znane prawdopodobieństwo) dostania się do próby.

Badacz eksperymentuje na próbie, która jest podzespołem populacji, po to, aby nie badać całej populacji (populacje są zwykle bardzo liczne). W związku z tym zależy mu na tym, aby próba była jak najbardziej podobna do populacji (była miniaturką populacji). Jeśli próba jest taką miniaturką, to badacz może spodziewać się, że wyniki eksperymentu uzyskane na próbie byłyby takie same jak wyniki uzyskane na populacji. Można powiedzieć, że badacz stara się na podstawie własności próby (wartości estymatorów) oszacować własności populacji (wartości parametrów).

Przykład. Badacz zastanawia się, jaka jest przeciętna masa Polaka. Aby się o tym dowiedzieć, nie musi ważyć wszystkich Polaków. Wystarczy, że dobierze taką próbę, która będzie charakterystyczna dla całej populacji Polaków. Badacz nie może dobierać według swojego uznania osób badanych. Ucieka się do *doboru losowego*, zakładając, że jeśli ślepy traf zrządzi tym, kto znajdzie się w jego próbie, to nie ma powodów przypuszczać, że grupa ta będzie składała się z samych chudych lub z samych otyłych. Jeśli dobór był losowy, to struktura próby jest prawdopodobnie taka jak struktura populacji.

Tego rodzaju wnioskowanie jest obarczone błędem wynikającym z przybliżenia (cechy próby będą jedynie przybliżone do cech populacji). Na wyniki uzyskane przy pomocy doboru losowego wpływa też błąd systematyczny wynikający z niewłaściwego próbkowania i innych możliwych systematycznych błędów. Błędy doboru próby nie występują w próbie o wielkości równej wielkości populacji.

Istotą doboru losowego nie jest losowanie, ale prawdopodobieństwo znalezienia się w próbie. Jeśli badacz na przykład zastanawia się, ilu Polaków z jego miasta wyjechało za granicę do pracy, i postanowi, że przebije cyrklem książkę telefoniczną i będzie dzwonić do wszystkich abonentów, którzy zostali "przedziurawieni", pytając ich, czy ktoś z rodziny wyjechał, to jest to wprawdzie ślepe losowanie, ale nie jest to dobór losowy. Nie wszyscy mieszkańcy jego miasta mieli bowiem szanse znalezienia się w jego próbie. Niektórzy nie mają telefonów, mają zastrzeżone numery lub innego operatora. Oznacza to, że mimo inteligentnego losowania dobór nie jest losowy, a wyniki z próby nie mogą być uogólnione na populację mieszkańców jego miasta.

Zobacz też [edytuj | edytuj kod]

- dobór próby
- dobór celowo-losowy
- dobór celowy

Kategoria: Dobór próby statystycznej

Rysunek 3.1: Przykładowy artykuł z portalu Wikipedia. Na niebiesko wyróżnione zostały linki do innych tekstów z portalu. Źródło: `http://pl.wikipedia.org/wiki/Dob%C3%B3r_losowy`

Zgromadzone dane to zbiór 1 075 568 artykułów z polskiej Wikipedii z dnia 2. listopada 2014 roku w postaci plików XML [2]. Teksty składają się z treści sformułowanych w języku naturalnym, wzorów, kodów, linków wewnętrznych Wikipedii, linków do źródeł zewnętrznych, odniesień do źródeł bibliograficznych, cytatów, przypisów, rysunków (zdjęć, wykresów) wraz z podpisami, tabel, komentarzy, uwag, spisu treści, sekcji „Zobacz też", kategorii oraz znaczników typowych dla plików HTML-owych. Każdy z wyżej wymienionych elementów jest wyróżniony w tekście w inny sposób (np. linki wewnętrzne Wikipedii zawsze znajdowały się wewnątrz podwójnych nawiasów kwadratowych), co nie pozostaje bez znaczenia przy wstępnym przetwarzaniu danych.

3.2. Wstępne przetwarzanie danych

Ważnym elementem przy pracy z danymi jest ich wstępna obróbka, szczególnie gdy są to dane tekstowe. Jednocześnie nie ma ogólnych, odpowiednich dla wszystkich zagadnień, reguł postępowania – schemat działania trzeba dostosować pod konkretny problem i posiadane dane. Naturalnie kwestię wstępnej obróbki danych można pominąć, jednak wiąże się to z ryzykiem negatywnego wpływu na działanie algorytmu. Co więcej w przypadku gdy szczególnie istotne są słowa znajdujące się w tekście, przygotowanie danych może okazać się jedną z kluczowych kwestii, jeśli chodzi o jakość działania algorytmu.

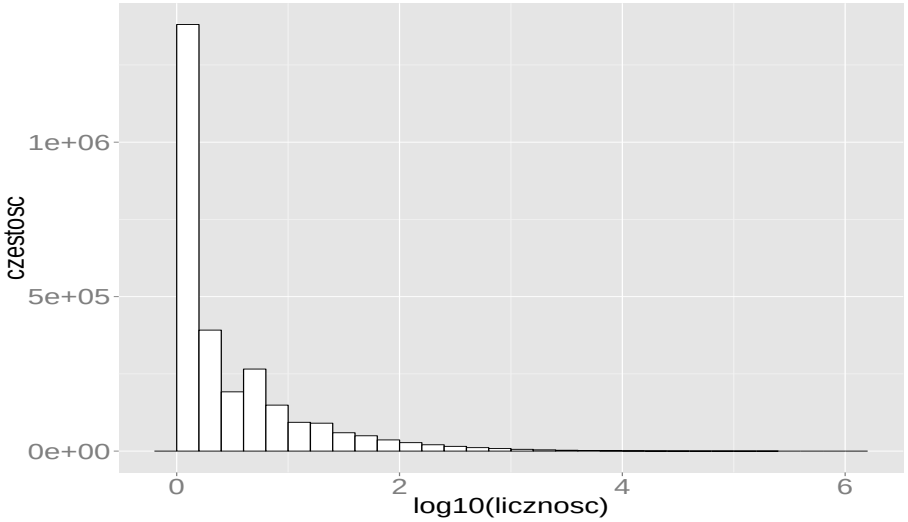
[2] Źródło: `http://dumps.wikimedia.org/plwiki/20141102/`

Jak wspomniano wcześniej pobrane dane składają się w dużej mierze z treści sformułowanych w języku naturalnym, jak i zawartości technicznej takiej jak linki czy znaczniki HTML-owe. Określenie istotności treści zawartej w poszczególnych częściach jest zasadniczym problemem przy wstępnej obróbce danych. Przyjmijmy, że część związaną z językiem naturalnym będziemy nazywać „tekstową", a pozostałe części tekstu – „techniczną". Związane są z nimi następujące aspekty (zaznaczmy, że rozważania te wymagają nieformalnego podejścia, intuicji oraz początkowego przejrzenia tekstów, co jest nieodłączną częścią praktycznej analizy danych):

- Część tekstowa zawiera główny opis artykuły, można się więc spodziewać, że w tej części zawarte zostanie meritum tekstu.
- Linki składają się ze słowa, które pojawia się w tekście, jak i odniesienia do innej strony. Ta pierwsza część może więc zawierać dużo informacji o temacie tekstu.
- Wzory oraz kody mogą dużo powiedzieć o tematyce artykułu (np. bardzo łatwo odróżnić wzór chemiczny od matematycznego), jednak w większości składają się one ze krótkich ciągów znaków (np. pojedynczych liter), które mogą być charakterystyczne dla wielu problemów.
- Cytaty mogą być ważne, choć często mogą mocno odbiegać od głównej tematyki tekstu.
- Przypisy są dodatkową informacją zawartą w tekście, często poruszające tematy poboczne.
- Odniesienia bibliograficzne, choć ważne z punktu widzenia wartości treści zawartych w artykule, nie wnoszą istotnych informacji o tematyce artykułu.
- Część artykułów zawiera bardzo dużo tabel, które czasem stanowią niemal jedyną treść. Jednakowoż służą one uporządkowaniu wiedzy i treść w nich zawarta często nie wnosi istotnych informacji o tematyce tekstu, zawierając jedynie słowa hasłowe bądź wylistowania danych zagadnień.
- Podpisy pod rysunkami są zazwyczaj powtórzeniem zdań bądź ich fragmentów z części opisowej.
- Spis treści zawiera tytuły, które są następnie powtórzone w tekście.
- Sekcja „Zobacz też" może być ważna, gdyż wskazuje na połączenia tematyczne między tekstami.
- Podobnie kategorie wskazują w szczególności na tematy poruszanego zagadnienia.
- Znaczniki HTML-owe oraz wyróżnienia powyższych elementów nie wnoszą żadnej informacji o tematyce tekstu i służą jedynie odpowiedniemu wyświetleniu treści.

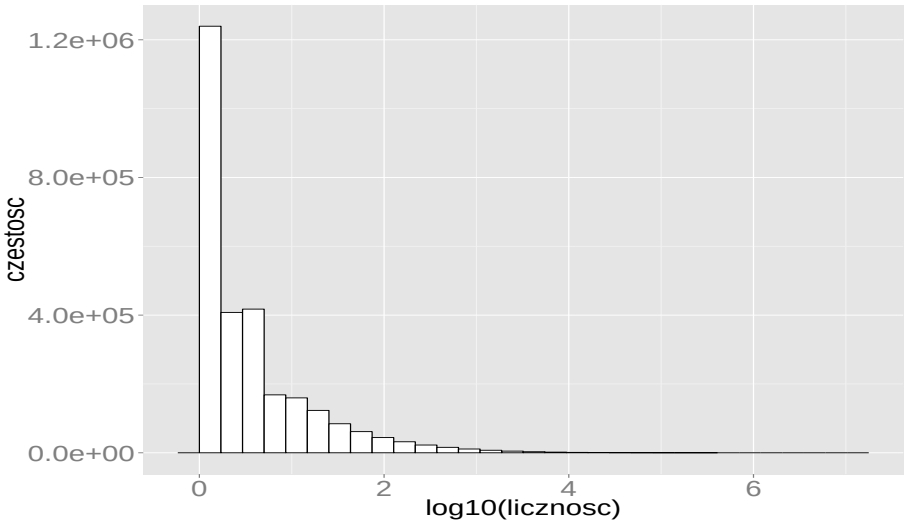
Wstępną obróbkę danych przeprowadzamy uwzględniając powyższe aspekty. Początkowo z części tekstowej usuwamy wszystkie znaki nie będące literami alfabetu łacińskiego. Poddajemy jej dalszej obróbce po przetworzeniu części technicznej. Słowa, które występują w tekście, a pod którymi znajduje się link, pozostawiamy bez zmian. Wzory oraz kody usuwamy w całości ze względu na trudność w rozróżnieniu czego podany fragment dotyczy oraz ze względu na krótkie znaki, które zawierają. Cytaty pozostawiamy bez zmian. Przypisy usuwamy z artykułów, jako że zawierają jedynie dodatkową z punktu widzenia głównego tekstu, treść. Źródła bibliograficzne usuwamy w całości. Teksty oczyszczamy także z tabel, rysunków, podpisów pod nimi oraz spisu treści. Sekcję „Zobacz też" oraz kategorie pozostawiamy jako potencjalne

źródło podobnej tematyki do tej zawartej w tekście. Wszelkie znaczniki HTML-owe usuwamy jako bezwartościowe z punktu widzenia treści artykułu. Podobnie teksty oczyszczamy z tytułów, które pojawiają się w większości tekstów, tj. *Zobacz też, Linki zewnętrzne* oraz *Bibliografia*.



Rysunek 3.2: Histogram logarytmu dziesiętnego z liczby artykułów, w których dane słowo występuje.

Tak otrzymane teksty dzielimy na słowa, które przekształcamy do wyrazów o małych literach. Nie chcemy rozróżniać słów ze względu na wielkość liter, gdyż nie powinna ona mieć znaczenia dla tematyki treści. Do każdego tekstu dodajemy informację o tym, jakie słowa i w jakich licznościach w nim występują. W ten sposób otrzymujemy 2 806 765 różnych słów we wszystkich, tj. w 1 075 568, artykułach. 49% słów występuje tylko w jednym tekście (p. rys. 3.2). Nieco ponad 3.5% słów znajduje się w stu i więcej artykułach, natomiast jedynie 0.6% wszystkich wyrazów pojawia się w więcej niż tysiącu tekstów.



Rysunek 3.3: Histogram logarytmu dziesiętnego z liczby wystąpień słów we wszystkich artykułach.

44% słów występuje dokładnie raz we wszystkich artykułach (p. rys. 3.3). Prawie 4.3% wyrazów pojawia się ponad sto razy, natomiast mniej niż jeden procent słów występuje tysiąc i więcej razy.

Słowa występujące tylko w jednym tekście to zazwyczaj słowa w obcych językach, przykładowo: `jużdortransstroj, youtsos, odety, knežlaz, pallebitzke, rulicach, werkowie, rumilla, metyklotiazyd, bazelak`, choć czasem są to słowa będące odmianą słów bardziej częściej występujących, np. słowo `uchybiają` jest odmianą czasownika `uchybiać`. Takie słowa warto wziąć pod uwagę przy analizie, gdyż są „podobne" do słów częściej występujących, a więc mogą polepszyć jakość dopasowania pod względem tematycznym.

	Słowo	Liczba artykułów	Liczba wystąpień
1	w	1 003 961	10 330 250
2	i	726 209	4 290 653
3	na	689 559	3 215 428
4	z	649 564	3 252 352
5	do	559 672	2 297 910
6	się	537 360	2 094 912
7	roku	420 178	1 292 537
8	a	378 941	919 278
9	od	378 327	935 799
10	jest	343 846	993 143
11	przez	336 596	852 463
12	oraz	288 718	663 760
13	po	286 818	765 075
14	o	273 574	674 431
15	ur	241 888	365 934
16	to	234 629	527 121
17	jako	232 919	663 364
18	latach	232 017	380 362
19	był	229 219	503 938
20	został	228 985	509 705

Tabela 3.1: Lista dwudziestu najczęściej występujących słów. Kolumna oznaczona jako *Liczba artykułów* oznacza liczbę tekstów, w których wystąpiło dane słowo, natomiast ostatnia kolumna mówi o ilości wystąpień wyrazu ogółem we wszystkich artykułach.

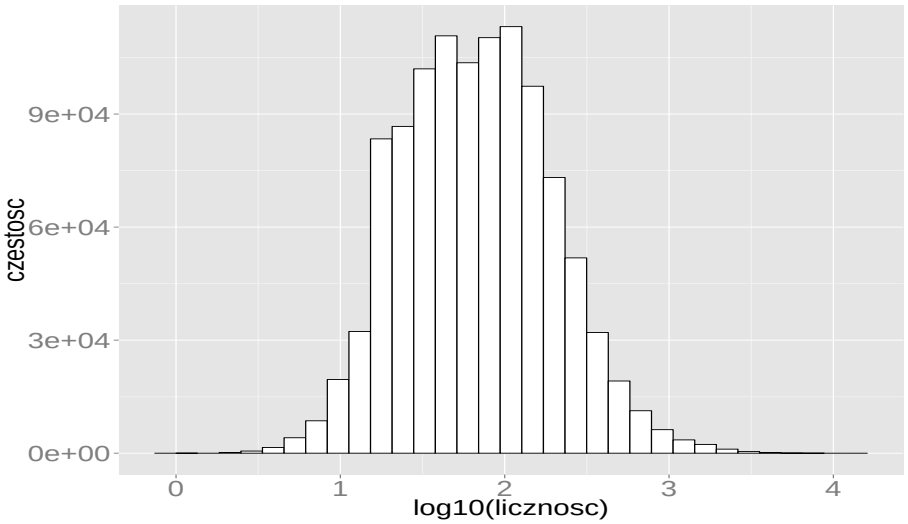
Słowa najczęściej występujące prezentuje tabela 3.1. W większości przypadków są to słowa nie istotne w kontekście analizy tematycznej tekstu (ang. *stopwords*). Takich słów można wyróżnić więcej, np.

`ach, aj, albo, bardzo, bez, bo, być, ci, cię, ciebie, co, czy, daleko, dla, dlaczego, dlatego, do, dobrze, dokąd, dość, dużo, dwa, dwaj, dwie, dwoje, dziś, dzisiaj, gdyby, gdzie, go, ich, ile, im, inny, ja, ją, jak, jakby, jaki, je, jeden, jedna, jedno, jego, jej, jemu, jeśli, jest, jestem, jeżeli, już, każdy, kiedy, kierunku, kto, ku, lub, ma, mają, mam, mi, mną, mnie, moi, mój, moja, moje, może, mu, my, na, nam, nami, nas, nasi, nasz, nasza, nasze, natychmiast, nią, nic, nich, nie, niego, niej, niemu, nigdy, nim, nimi, niż, obok, od, około, on, ona, one, oni, ono, owszem, po, pod, ponieważ, przed, przedtem, są, sam, sama, się, skąd, tak, taki, tam, ten, to, tobą, tobie, tu, tutaj, twoi, twój, twoja, twoje, ty, wam, wami, was, wasi, wasz,`

`wasza, wasze, we, więc, wszystko, wtedy, wy, żaden, zawsze, że`

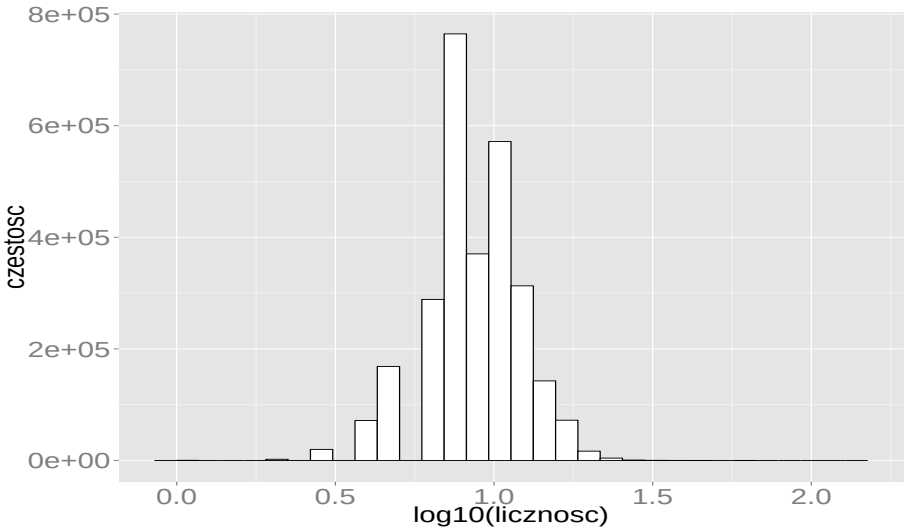
Te i podobne słowa oraz wyrazy jedno- i dwuznakowe, należy usunąć ze zbioru danych, gdyż nie wnoszą żadnej istotnej informacji o tematyce tekstu. Dokonujemy tego w kolejnym etapie wstępnej obróbki tekstów. Łącznie ze zbioru słów usuwamy 1 822 wyrazy.

Średnia liczba unikalnych słów występujących w artykule wynosi 121, mediana to zaledwie 66 unikalnych wyrazów. W 2 272 tekstach wystąpiło mniej niż pięć słów. Artykuły te zawierały przede wszystkim tabele i rysunki, stąd po wstępnej obróbce danych pozostało w nich niewiele wyrazów. Histogram logarytmu dziesiętnego liczby unikalnych słów w tekstach przedstawia rys. 3.4.



Rysunek 3.4: Histogram logarytmu dziesiętnego liczby unikalnych słów w artykule.

Mediana długości słów wynosi 9, średnia jest nieco wyższa (p. rys. 3.5). Najkrótsze występujące słowa są jednoznakowe, jest ich 249. Najdłuższe słowo ma 128 znaków. Słowa o długości ponad 11 znaków stanowią 19.5% wszystkich słów.



Rysunek 3.5: Histogram logarytmu dziesiętnego długości słów.

3.3. Utworzenie skupień „podobnych” słów / Jakiś mądry tytuł

Po wstępnej obróbce danych dostajemy 2 806 765 unikalnych słów, z czego blisko połowa występuje jedynie raz we wszystkich tekstach. Algorytm dzielący teksty tematycznie bierze pod uwagę liczności słów, które znajdują się w artykule a daną wejściową jest macierz o wymiarach liczba artykułów × liczba słów. Jeśliby nie przetworzyć dalej danych, to macierz ta miałaby wymiary 1 075 568 × 2 806 765, gdzie przeważająca większość rekordów byłaby równa zeru. Algorytm mógłby nie poradzić sobie z tak dużą ilością danych, zwłaszcza gdy większa część rekordów jest „niewypełniona”. Stąd zachodzi potrzeba zmniejszenia wymiaru danych.

Ponieważ rekordy składają się ze słów, można wyznaczyć skupienia wyrazów „podobnych” do siebie. Podobieństwo (odmienność) słów można określić za pomocą odległości na przestrzeni ciągów znaków opisanych w rozdziale 1. Następnie dany tekst można przedstawić jako sumę liczby wystąpień słów z danego skupienia, zamiast liczby wystąpień pojedynczych słów. Przykładowo, jeśli skupienie A składa się ze słów $s,\ t,\ u$, które wystąpiły w danym artykule, odpowiednio, $x,\ y,\ z$ razy, to słowa ze skupienia A wystąpiły w tym tekście łącznie $x+y+z$ razy. Dzieląc wszystkie wyrazy na skupienia, możemy znacząco zmniejszyć liczbę wyrazów, biorąc pod uwagę liczności grup słów, zamiast liczności pojedynczych wyrazów.

Aby podzielić zbiór słów na skupienia, najbardziej naturalne wydaje się zbudowanie macierzy odległości wszystkich słów od siebie i zastosowanie algorytmu aglomeracyjnego. Jednakowoż wiąże się to z dużą złożonością pamięciową i obliczeniową, stąd też podejście takie nie zostało wykorzystane w niniejszej pracy. Inny pomysł polega na przyłączaniu do skupienia słów dopóty, dopóki średnia odległość w zbiorze nie przekroczy zadanej liczby. Wstępne testy na losowej próbce tysiąca słów wykazały, że jakość takiego podziału jest słaba, a czas obliczeń względnie długi.

Stemming. Stąd postanowiono dokonać podziału zbioru słów w inny sposób. Początkowo przeprowadzamy tzw. *stemming*, czyli sprowadzenie słowa do jego rdzenia. Odmiana słowa nie zmienia jego tematyki, a dzięki takiemu podejściu możemy znacznie ograniczyć liczbę unikalnych słów w zbiorze. Przykładowo słowa `zjednoczonych`, `zjednoczyli`, `drużynom`, `drużynie` zostaną sprowadzone odpowiednio do form `zjednoczyć`, `drużyna`. Dzięki takiemu podejściu, każde skupienie będzie miało swoje *słowo-reprezentanta* (środek), które jednoznacznie charakteryzuje podzbiór. Będziemy je nazywać środkiem, słowem-reprezentantem lub po prostu *reprezentantem* skupienia. W ten sposób do odpowiedniego podzbioru słów możemy odnosić się poprzez jego reprezentanta.

	Język	Liczba słów	Procent ogółu
1	polski	664 315	23.7
2	angielski	41 087	1.5
3	niemiecki	21 117	0.8
4	francuski	20 438	0.7
5	ogółem	746 942	26.6

Tabela 3.2: Liczba słów na których zastosowano *stemming* w poszczególnych językach

.

Do przeprowadzenia *stemmingu* używamy programu *Hunspell* [3] – korektora pisowni i analizatora morfologicznego używany w wielu programach typu *open source*. Aplikacja ta ma wbudowany słownik słów języka polskiego wraz z ich odmianami. Do każdego wyrazu jest też przypisany jego rdzeń. Sposób działania jest następujący: program znajduje szukane słowo w słowniku, a następnie zwraca jego rdzeń lub nie zwraca nic, jeśli słowo nie pasuje do żadnego wyrazu ze słownika. W szczególności słownik nie zawiera wyrazów, w których popełniono tzw. „literówki" ani złączeń słów.

Ponieważ część wyrazów stanowią słowa obcojęzyczne przeprowadzamy *stemming* w języku polskim, angielskim, niemieckim oraz francuskim (p. tabela 3.2). W ten sposób grupujemy 746 942 słów, co stanowi ok. 27% wszystkich słów, w 186 958 skupienia. Przykładowe skupienia prezentuje tabela 3.3. Warto zauważyć, że słowa w podzbiorach są podobne tematycznie, choć do skupienia o reprezentancie `główny` trafiły wyrazy o znaczeniu przeciwnym. Widać więc, że podział taki nie jest idealny.

Reprezentant	Słowa w skupieniu
czas	czasie, czasach, czas, czasom
główny	głównie, główne, główną, głównych, głównego, głównym, główna, głównymi, główny, głównej, główni, głównemu, niegłówny, niegłównym, niegłówne, niegłównych, niegłówną
miał	miał, miały, miałem, miału, miałach, miale, miałów, miałami, miałom
nazwa	nazwa, nazwę, nazwy, nazwą, nazwie, nazw, nazwami, nazwach, nazwom
nr	nr, nry, nru, nrem, nrze, nrów
osoba	osób, osoby, osoba, osobą, osobę, osobom, osobie, osobami, osobach, osobo
udział	udział, udziału, udziałem, udziale, udziały, udziałów, udziałami, udziałach, udziałom
wieś	wsi, wsie, wieś, wsią, wsiach, wsiami, wsiom
zostać	został, została, zostały, zostało, zostanie, zostali, zostać, zostaną, zostania, zostałby, zostałaby, zostałyby, zostaniu, zostałem, zostałoby, zostaniesz, zostanę, zostaliby, zostaliśmy, zostaniemy, zostaniem, zostałam, zostałeś, zostaliście, zostaniecie, zostałaś, zostałbym, zostalibyśmy, zostałbyś, zostano, zostałabym

Tabela 3.3: Przykładowe skupienia uzyskane przy pomocy *stemmingu*.

Podział przy użyciu metryk. Tak zaproponowany podział słów na skupienia wykorzystuje jedynie ok. 27% zbioru wszystkich wyrazów. Co więcej większość z podzbiorów jest zaledwie kilkuelementowa (p. tabela 3.4).

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	1	2	4	5	89

Tabela 3.4: Rozkład liczby słów w skupieniu.

Stąd można zastosować następujące schematy postępowania, które po pierwsze zredukują liczbę używanych grup słów, a po drugie wykorzystają dodatkowo zbiór niepogrupowanych wyrazów. Procedury te polegają na [PLIK R/14]:

1. Dołączeniu do skupień słów jeszcze niepogrupowanych.

[3] `http://hunspell.sourceforge.net/`

2. Dołączeniu do skupień zawierających pięć i więcej elementów, podzbiorów o mniejszej liczności.

3. Zastosowaniu najpierw punktu 1, a następnie punktu 2.

Omówimy teraz bliżej na czym polegają powyższe kroki.

Procedura 1. W kroku tym chcemy użyć większej liczby dostępnych słów. Dzięki temu zwiększą się liczności występowania grup słów w tekście, co być może polepszy jakość podziału artykułów na skupienia. Algorytm ten jednak nie zmieni liczby skupień.

Schemat działania jest następujący: bierzemy słowo nieprzydzielone do żadnego skupienia. Liczymy odległość tego wyrazu (przy użyciu odległości zdefiniowanych w rozdziale 1) od wszystkich słów-reprezentantów dotychczasowo otrzymanych skupień. Słowo przydzielamy do tego podzbioru, do którego reprezentanta było mu najbliżej (p. algorytm 4). Jeśli wyraz ma taką samą (najmniejszą) odległość do kilku środków, wybieramy pierwszy w kolejności. Jeśli odległość słowa do wszystkich reprezentantów jest nieokreślona lub równa nieskończoności, to wyraz ten pomijamy, tj. nie dodajemy go do żadnego ze skupień. Taka sytuacja może się zdarzyć, w przypadku zastosowania odległości opartych na q-gramach, gdy długość słowa jest mniejsza od q.

Algorithm 4 Algorytm przydzielający niepogrupowane słowo do skupienia.

given: data set of representants R, data set of words to categorize W, metric d
for $w \in W$ **do**
    $C(w) = \arg \min_{i:\ r_i \in R} d(w, r_i)$
    if length$(C(w)) > 1$ **then**
        $C(w) = C(w)[1]$
    end if
end for

Zauważmy, że powyższy algorytm podobny jest do metody aglomeracyjnej analizy skupień. Różnice polegają na tym, że do utworzonych już skupień przyłączamy pojedyncze obserwacje (słowa), a kryterium oceny odmienności jest odległość danego słowa od środka (reprezentanta) skupienia.

Procedura 2. W tym kroku chcemy zredukować liczbę uzyskanych skupień. Dzięki temu zwiększą się liczności występowania grup słów w tekście, a ponadto zmniejszy się liczba skupień, co może przyczynić się do lepszego działania algorytmu dzielącego teksty na skupienia.

Schemat działania jest następujący (p. algorytm 5): sprawdzamy, jakie są liczności wszystkich skupień. Jeśli liczność skupienia jest większa od pięciu, to taki podzbiór oznaczamy jako „duży". Do takich skupień będziemy przyłączać mniejsze podgrupy. Jeśli liczność skupienia jest mniejsza lub równa 5, to podzbiór oznaczamy jako „mały". Takie skupienie będziemy przyłączać do podzbiorów „dużych". Te pierwsze skupienia nazwijmy dużymi skupieniami, natomiast te drugie – małymi.

Mając tak podzielone skupienia, weźmy reprezentantów małych podzbiorów. Jeśli długość słowa-reprezentanta nie przekracza trzech znaków, to skupienie takie pomijamy w dalszej analizie. Ma to na celu uniknięcie analizy słów, które nie mają znaczenia, jak zbitek dwóch lub trzech takich samych liter (np. `aa` lub `bbb`). Następnie postępowanie jest podobne jak w algorytmie 4: liczymy odległość środka małego skupienia od wszystkich słów-reprezentantów dużych skupień. Sprawdzamy, która z wyliczonych odległości była najmniejsza. Podzbiór,

Algorithm 5 Algorytm łączący małe i duże skupienia.

given: data set of representants R, vector of clusters' size **s**, metric d
$R_m = \emptyset, R_d = \emptyset$
for $r \in R$ **do**
    if $s_r \leq 5$ **then**
        $R_m = R_m \cup r$
    else
        $R_d = R_d \cup r$
    end if
end for
for $w \in C_m$ **do**
    if $|w| < 4$ **then**
        continue / next
    end if
    $C(w) = \arg \min_{i:\ r_i \in R_d} d(w, r_i)$
    if length$(C(w)) > 1$ **then**
        $C(w) = C(w)[1]$
    end if
end for

którego reprezentantem jest analizowane słowo, przydzielamy do tego skupienia, do którego środka było mu (słowu) najbliżej. Jeśli wyraz ma taką samą (najmniejszą) odległość do kilku reprezentantów, wybieramy pierwszego w kolejności.

Zauważmy, że druga część algorytmu to po prostu metoda aglomeracyjna z innym niż zaprezentowane w rozdziale 2 kryterium liczenia odmienności dwóch skupień. W tej procedurze odmienność między podzbiorami określona jest jako odległość między środkami (reprezentantami) skupień.

Procedura 3. Krok trzeci polega na wykonaniu najpierw procedury pierwszej, a następnie drugiej.

Wybór odległości. Mając trzy powyższe algorytmy, możemy przystąpić do dalszej obróbki zbioru słów. Zanim to jednak nastąpi należy wybrać odległości, dzięki którym będzie to możliwe. W rozdziale 1 przedstawiono pięć odległości opartych na operacjach edycyjnych, trzy odległości oparte na q-gramach oraz dwie miary heurystyczne.

Odległość Hamminga odrzucamy, gdyż można ją zastosować jedynie na napisach o tej samej długości. Odległości najdłuższego wspólnego podnapisu, Levenshteina, optymalnego dopasowania napisów i Damerau-Levenshteina różnią się jedynie zbiorem bazowych operacji edycyjnych, często dając tę samą odległość. Stąd postanowiliśmy użyć dwóch „skrajnych" odległości, tj. takich, które pozwalają na najmniejszą i największą liczbę bazowych operacji edycyjnych, czyli odległość najdłuższego wspólnego podnapisu (lcs) i Damerau-Levenshteina (dl).

Z odległości opartych na q-gramach wyselekcjonowaliśmy odległość Jaccarda (jac) oraz q-gramową (qg) jako najbardziej reprezentatywne. W obu przypadkach wybraliśmy $q = 4$. Dzięki takiemu podejściu unikniemy przetwarzania słów o długości mniejszej niż cztery znaki.

Miary heurystyczne pominęliśmy.

Otrzymane zbiory. [GDZIES TU NAPISAC, ZE UZYWALAM R-A?] Na zbiorze skupień otrzymanym po wykonaniu *stemmingu* zastosowano trzy powyższe algorytmy przy użyciu każdej z czterech odległości, dostając łącznie 13 różnych zbiorów skupień (wliczając w to zbiór, otrzymany ze *stemmingu*). Zbiory te oznaczmy jako *clust_X* gdzie X jest przyrostkiem oznaczającym algorytm i zastosowaną odległość. Metodologia nazewnictwa jest następująca: w przypadku, gdy dołączaliśmy słowa do istniejących skupień (tj. zastosowany był algorytm 4 / procedura 1) dodajemy jedynie przyrostek oznaczający zastosowaną odległość, tj. *lcs, dl, jac* lub *qg*, np. *clust_lcs*. Jeśli użyliśmy algorytmu 5 / procedury 2, zmniejszającego liczbę skupień, to dodajemy przyrostek *red_* oraz zastosowaną odległość, np. *clust_red_lcs*. Jeśli oba algorytmy zostały zastosowane, to łączymy je w nazwie, dostając np. *clust_lcs_red_lcs*. Zbiór otrzymany po wykonaniu *stemmingu* oznaczamy po prostu *clust*.

Liczbę skupień oraz liczbę słów zawartą w skupieniu dla poszczególnych zbiorów zawiera tabela 3.5. Zbiory, na których zastosowano algorytm 4 lub jedynie *stemming* zawierają 186 958 skupień, co dało redukcję ok. 93% względem oryginalnego zbioru słów (tj. 2 806 765). W skupieniach tych znajduje się od prawie 750 000 do ponad 1 000 000 słów, czyli między 27% a 38% wszystkich wejściowych wyrazów. Druga grupa zbiorów, tj. taka, która jest wynikiem działania algorytmu 5 zawiera dokładnie 43 919 skupienia, co daje redukcję równą 98.4%. Słowa zawarte w tych skupieniach stanowią ok. 26% wejściowego zbioru wyrazów. Trzecia grupa zbiorów, ma nieco mniejszą redukcję niż poprzednia i wynosi prawie 98%, zawierając jednocześnie ok. 38% wszystkich wyrazów.

	Zbiór	Liczba skupień	Redukcja	Liczba słów	Procent
1	clust	186 958	93.3%	746 957	27%
2	clust_lcs	186 958	93.3%	1 080 260	38%
3	clust_dl	186 958	93.3%	1 080 260	38%
4	clust_jaccard	186 958	93.3%	1 070 750	38%
5	clust_qgram	186 958	93.3%	1 070 750	38%
6	clust_red_lcs	43 919	98.4%	743 053	26%
7	clust_red_dl	43 919	98.4%	743 053	26%
8	clust_red_jaccard	43 919	98.4%	739 338	26%
9	clust_red_qgram	43 919	98.4%	739 338	26%
10	clust_lcs_red_lcs	65 350	97.7%	1 037 393	37%
11	clust_dl_red_dl	66 378	97.6%	1 060 474	38%
12	clust_jaccard_red_jaccard	69 570	97.5%	1 063 131	38%
13	clust_qgram_red_qgram	62 434	97.8%	1 063 131	38%

Tabela 3.5: Zbiory skupień wraz z ich liczbą oraz liczbą słów w skupieniu. Redukcja oznacza procent zredukowania z wejściowego zbioru słów do liczby otrzymanych skupień. Ostatnia kolumna mówi ile procent wszystkich słów zbioru wejściowego znajduje się w skupieniu.

[TO DO: DODAC PRZYKLADOWE SKUPIENIA DLA WSZYSTKICH 13 ZBIOROW]

3.4. Podział tekstów

Mając tak zdefiniowane skupienia słów możemy przystąpić do podziału zbioru artykułów. Do tego celu użyjemy algorytmu mini-wsadowego k-średnich (algorytm 3 z rozdziału 2). Przypomnijmy, że algorytm ten jest metodą pośrednią pomiędzy algorytmem wsadowym, który w każdej iteracji opiera się na wszystkich obserwacjach, a algorytmem SGD, biorącym w każdej iteracji po jednej obserwacji ze zbioru.

Aby więc użyć algorytmu mini-wsadowego musimy wybrać najpierw liczbę skupień k oraz parametr b, określający ile obserwacji będzie miało swój wkład w każdej iteracji. Zajmijmy się najpierw tą drugą wartością. Ponieważ nie wiemy jak bardzo jakość podziału zależy od parametru b, postanowiliśmy sprawdzić działanie algorytmu dla czterech wartości b: 5 000, 10 000, 35 000 oraz 70 000. Dostaniemy w ten sposób 52 wyniki analizy, oparte na 13 różnych zbiorach wejściowych.

Zanim określimy wartość parametru k, zastanówmy się w jaki sposób będziemy mierzyć jakość otrzymanych podziałów. Cztery na pięć zaprezentowanych miar w rozdziale 2 wymaga znajomości prawdziwego podziału zbioru. Przypomnijmy, że nasz zbiór danych to artykuły z polskiej Wikipedii, które mają określoną kategorię tematyczną. Można więc wykorzystać znaną nam wiedzę o kategoriach i na jej podstawie określić jakość podziału otrzymanego w wyniku działania algorytmu. Liczba różnych kategorii, które określają tematykę artykułów wynosi 56 283. Próba wykonania analizy skupień przy użyciu algorytmu mini-wsadowego z tak dużym k, zakończyła się niepowodzeniem, mimo posiadania dużej ilości pamięci RAM (wraz z partycją wymiany (SWAP) ponad 100 GB). Stąd też nastąpiła potrzeba zredukowania tej liczby. Ponieważ struktura kategorii Wikipedii jest drzewiasta [4], można zastąpić kategorię przypisaną do artykułu kategorią ogólniejszą. Po takiej redukcji otrzymano 6 922 grup tematycznych. Jednak rozkład liczby artykułów w otrzymanych kategoriach był mocno skośny. Taka sytuacja jest silnie niesprzyjająca, gdyż chcemy mieć podobne liczności w grupach. Ręcznie podzielono zatem najbardziej liczne tematy na podtematy, natomiast te o najmniejszej liczności połączono zachowując przy tym podobieństwo tematyki. W ten sposób uzyskano sto różnych tematów o podobnym rozkładzie liczby artykułów. Wstępne testy wykazały, że tak otrzymane k pozwala na dokonanie obliczeń w relatywnie krótkim czasie i nie zajmując dużej ilości pamięci RAM.

Wstępne testy na ok. 2% artykułów wykazały dość dobre przyporządkowanie (jednorodność i zgodność na poziomie, odpowiednio, 0.3 i 0.7). Testy na większej próbce ok. 15% tekstów dały wyniki nieco gorsze (jednorodność i zgodność na poziomie, odpowiednio 0.2 i 0.5). Stąd też postanowiono przeanalizować działanie algorytmu dla trzech różnych liczności próbki: 100% (1 075 568 artykułów), ok. 15% (152 772 artykuły) oraz ok. 2% zbioru (25 000 artykułów). Dla tej ostatniej próbki za wartość parametru b, określającego liczbę obserwacji mających wkład w każdej iteracji algorytmu, przyjęliśmy 5 000 oraz 10 000. Większe wartości (tj. 35 000 oraz 70 000) nie mają sensu, gdyż są większe od liczności próby.

Wszystkie analizy puszczono z tym samym ziarnem losowania, tj. w każdej analizie obserwacje były losowane w tej samej kolejności.

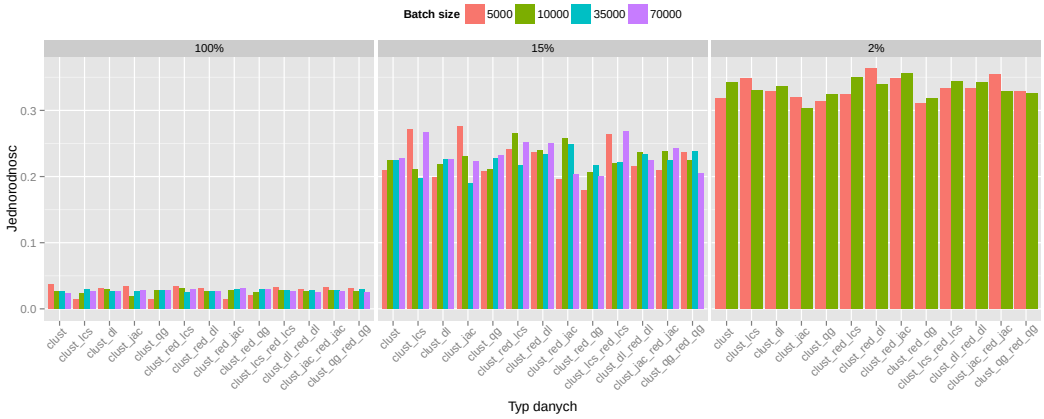
3.5. Analiza wyników

W niniejszym podrozdziale przeanalizujemy uzyskane wyniki dla uzyskanych podziałów tekstów.

Dokładne wyniki można znaleźć w tabelach 3.5, 3.6 oraz 3.7. Na rysunkach 3.6, 3.7, 3.8, 3.9 oraz 3.10 prezentujemy wartości uzyskanych, odpowiednio, jednorodności, zgodności, miary V, skorygowanego indeksu Randa oraz miary silhouettes dla wszystkich zbudowanych podziałów zbiorów. Przypomnijmy, że klasami porównawczymi do uzyskanych podziałów są tematy artykułów.

[4] por. `http://pl.wikipedia.org/wiki/Wikipedia:Drzewo_kategorii`

Weźmy pierwszy wykres, tj. jednorodność. Na pierwszy rzut oka widać, że największe wartości tej miary uzyskał podział zbudowany na 2% zbioru, uzyskując, nieco ponad 0.3 dla wszystkich uzyskanych podziałów. Nieco mniejszą wartość jednorodności dostaliśmy w przypadku podziału 15% artykułów o średniej równej ok. 0.2. Dla całego zbioru jednorodność jest bardzo mała i wynosi jedynie ok. 0.03. Oznacza to, że zdecydowana większość skupień zawiera teksty z różnych klas (tematów), co jest cechą wysoce niepożądaną. Wynika z tego, że czym mniejszy zbiór, tym lepszy jego podział ze względu na temat.



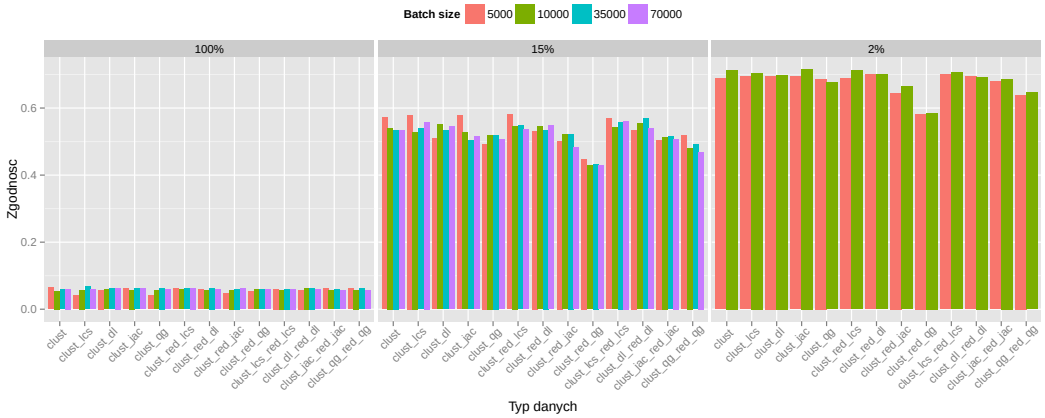
Rysunek 3.6: Jednorodność.

Przyjrzyjmy się teraz wartościom jednorodności ze względu na wielkość parametru b. Można stwierdzić, że wartość b nie ma istotnego wpływu na wysokość jednorodności. Warto przy tym zauważyć, że często jednorodność ma największą wartość dla $b = 5\ 000$ (zwłaszcza w przypadku podziału zbudowanego na całym zbiorze artykułów), co wydaje się być sprzeczne z teorią zaprezentowaną w rozdziale 2.

W końcu spójrzmy na wartości jednorodności w podziale na użyty zbiór (skupień) słów. Najwyższą wartość jednorodności uzyskano dla zbiorów *clust_red_jac* oraz *clust_red_dl*, *clust_jac* i *clust* dla, odpowiednio, podziału opartego na 2%, 15% oraz 100% zbioru. Wysoką wartość jednorodności uzyskano również dla *clust_lcs*, *clust_red_lcs*, *clust_jac_red_jac*. Stąd też można wnioskować, że pozytywny wpływ na jakość podziału miała procedura zmniejszająca liczbę skupień. Co więcej, najlepsze rezultaty uzyskano przy użyciu odległości Jaccarda oraz lcs. Warto przy tym zauważyć, że we wszystkich przypadkach najgorsze podziały uzyskano dla odległości q-gramowej.

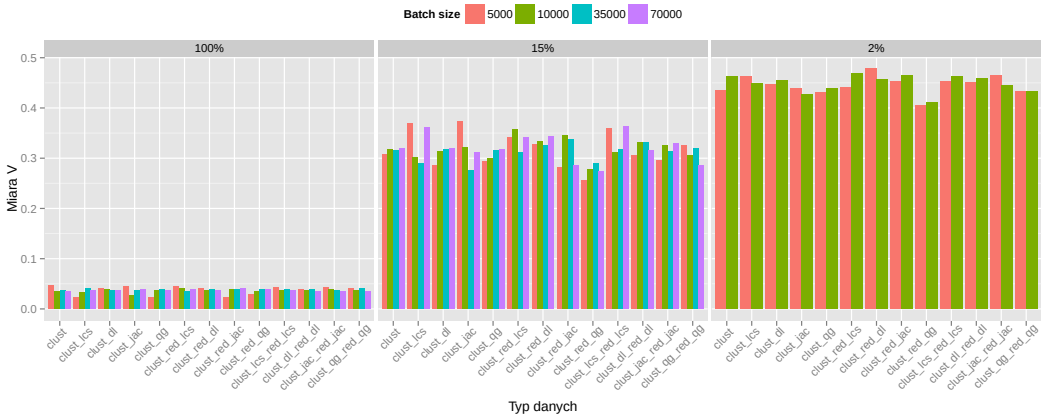
Przejdźmy do analizy zgodności podziału zbiorów na skupienia. Najwyższe wartości tej miary uzyskano na 2% zbioru tekstów i zawierały się w przedziale $[0.58, 0.72]$. Nieco mniejsze wartości zgodności dostaliśmy dla 15% zbioru i wynosiła ona średnio 0.52. Najniższe wartości zgodności uzyskano dla całego zbioru tekstów i były one równe ok. 0.06. Oznacza to, że obserwacje z danej klasy (tematu) znajdują się w prawie wszystkich możliwych skupieniach, a więc podział taki jest słaby. Stąd, czym mniejszy zbiór tym lepszy podział pod względem tematycznym.

Przyjrzyjmy się teraz wartościom zgodności ze względu na wielkość parametru b. Dla analizy wykonanej na 2% zbioru istotnie lepsze rezultaty uzyskano dla $b = 10\ 000$ – w prawie wszystkich przypadkach zgodność jest wyższa o $0.01 - 0.02$. Dalej, dla 15% tekstów w większości przypadków najlepsze wyniki dał algorytm z $b = 5\ 000$. Dla pełnego zbioru artykułów miara zgodności jest równa dla prawie wszystkich grup.



Rysunek 3.7: Zgodność.

Dalej spójrzmy na wartości zgodności w podziale na użyty zbiór (skupień) słów. Widać, że w przypadku dwóch mniejszych zbiorów tekstów, miara ta dla jest istotnie niższa dla skupień, które opierały się na odległości q-gramowej, tj. *clust_qg, clust_red_qg* oraz *clust_qg_red_qg* niż w analogicznych grupach, dla których użyto innej odległości. Z drugiej strony, średnio najlepsze wyniki uzyskano dla tych podzbiorów, do utworzenia których zastosowano metrykę lcs. Pośrednio znalazły się odległości dl oraz jac. Lepsze rezultaty dały algorytmy, gdzie zastosowano algorytm 5 lub najpierw algorytm 4, a następnie 5, choć różnice pomiędzy nimi są niewielkie.

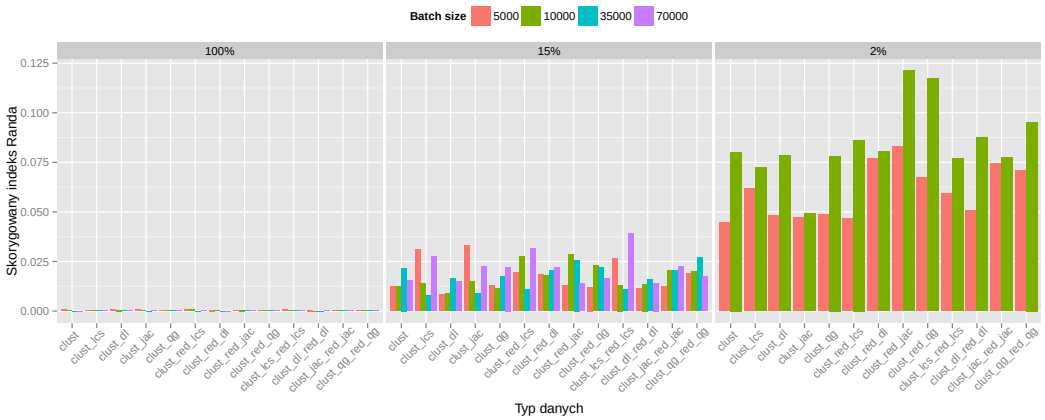


Rysunek 3.8: Miara V.

Ponieważ miara V jest średnią harmoniczną dwóch poprzednich miar, skupimy się jedynie na analizie uzyskanych wyników dla różnych zbiorów wejściowych. Jak można było się spodziewać, najlepsze wyniki uzyskano dla zbiorów, które budowane były przy użyciu odległości lcs. W drugiej kolejności najlepiej wypadła metryka dl. Również lepszy podział uzyskano, gdy zastosowano algorytm 5 lub najpierw algorytm 4, a następnie 5.

Przejdźmy do skorygowanego indeksu Randa. Podobnie jak w przypadku poprzednich miar, najwyższe wartości ARI uzyskano dla najmniejszego zbioru, i wynosiły średnio 0.07; 0.02 – dla analizy opartej o 15% zbioru wejściowego oraz 0.00 na podziale wykonanym na wszystkich

artykułach. Oznacza to, że podział tekstów na skupienia jest mocno niezgodny z klasami (tematami) artykułów.

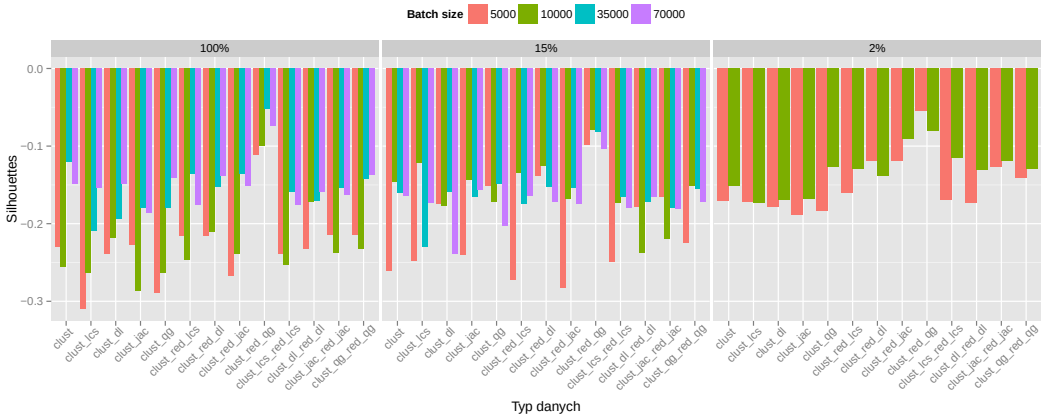


Rysunek 3.9: ARI.

W przypadku zbioru 2% tekstów, wyższą wartość ARI uzyskano dla $b = 10\ 000$. Dla drugiego pod względem liczności obserwacji zbioru, najwyższą wartość uzyskano dla $b = 70\ 000$. W przypadku analizy opartej na wszystkich artykułach b nie miało wpływu na wartość ARI.

Przyjrzyjmy się teraz wartościom ARI pod względem zbioru, który był dzielony na skupienia. Odwrotnie niż wcześniej najwyższe wartości tej miary uzyskano dla *clust_red_jac* oraz *clust_red_qg* dla najmniejszego zbioru. W przypadku dwóch pozostałych najlepszy okazał się podział zastosowany na zbiorach opartych o odległość lcs. Podobnie jak poprzednio lepsze rezultaty dały algorytmy, gdzie zastosowano algorytm 5 lub najpierw algorytm 4, a następnie 5 niż w przypadku zbiorów otrzymanych w wyniku działania tylko 4.

Przejdźmy do miary silhouettes (sylwetek). Ze względu na złożoność obliczeniową niemożliwe było policzenie średniej z sylwetek wszystkich obserwacji. Uzyskane sylwetki są średnią z próbki 10 000 losowych punktów. Wszystkie uzyskane wartości silhouettes są mniejsze od zera, czyli często teksty nie są dobrze przypisane do skupienia, tzn. lepsze byłoby przyporządkowanie danego artykułu do innej grupy niż tej, do której został przypisany.



Rysunek 3.10: Silhouettes.

Największe wartości sylwetek uzyskano dla *clust_red_qg*, co oznacza, że uzyskane przyporządkowania tekstów do skupień były lepsze niż w przypadku pozostałych zbiorów. Najniższe wartości uzyskano dla zbiorów opartych na odległości lcs i Jaccarda.

Podsumowując:

- najlepszy podział artykułów na grupy tematyczne uzyskano dla 2% tekstów,
- parametr b nie miał istotnego wpływu na jakość uzyskanego podziału,
- najlepsze wyniki uzyskano dla zbiorów powstałych przy użyciu odległości lcs,
- lepszy podział dostaliśmy, gdy zbiór przetworzono algorytmem 5, tj. zmniejszono liczbę skupień słów.

[TO DO:
- DLACZEGO WYSZLO TAK SLABO - JESZCZE NIE WIEM
- LICZNOSCI SKUPIEN
- PRZYKLADY JAKIE BYLY ARTYKULY W SKUPIENIACH
- CZY PODZIALY BYLY ZGODNE MIEDZY SOBA
]

3.6. Szczegółowe wyniki

[NIE WIEM GDZIE TO RZUCIĆ, MOŻE WARTO GDZIEŚ NA KONIEC JAKO DODATEK?]

	Typ danych	b	Jedn.	Zg.	Miara V	ARI	Silhouettes	Część
1	clust	5000	0.04	0.06	0.05	0.00	-0.23	100%
2	clust	10000	0.03	0.05	0.03	0.00	-0.26	100%
3	clust	35000	0.03	0.06	0.04	0.00	-0.12	100%
4	clust	70000	0.02	0.06	0.03	-0.00	-0.15	100%
5	clust_lcs	5000	0.02	0.04	0.02	0.00	-0.31	100%
6	clust_lcs	10000	0.02	0.06	0.03	0.00	-0.26	100%
7	clust_lcs	35000	0.03	0.07	0.04	0.00	-0.21	100%
8	clust_lcs	70000	0.03	0.06	0.04	0.00	-0.15	100%
9	clust_dl	5000	0.03	0.06	0.04	0.00	-0.24	100%
10	clust_dl	10000	0.03	0.06	0.04	0.00	-0.22	100%
11	clust_dl	35000	0.03	0.06	0.04	0.00	-0.19	100%
12	clust_dl	70000	0.03	0.06	0.04	0.00	-0.15	100%
13	clust_jac	5000	0.03	0.06	0.04	0.00	-0.23	100%
14	clust_jac	10000	0.02	0.06	0.03	0.00	-0.29	100%
15	clust_jac	35000	0.03	0.06	0.04	0.00	-0.18	100%
16	clust_jac	70000	0.03	0.06	0.04	0.00	-0.19	100%
17	clust_qg	5000	0.02	0.04	0.02	0.00	-0.29	100%
18	clust_qg	10000	0.03	0.06	0.04	0.00	-0.26	100%
19	clust_qg	35000	0.03	0.06	0.04	0.00	-0.18	100%
20	clust_qg	70000	0.03	0.06	0.04	0.00	-0.14	100%
21	clust_red_lcs	5000	0.03	0.06	0.04	0.00	-0.22	100%
22	clust_red_lcs	10000	0.03	0.06	0.04	0.00	-0.25	100%
23	clust_red_lcs	35000	0.03	0.06	0.04	0.00	-0.14	100%
24	clust_red_lcs	70000	0.03	0.06	0.04	0.00	-0.18	100%
25	clust_red_dl	5000	0.03	0.06	0.04	0.00	-0.22	100%
26	clust_red_dl	10000	0.03	0.06	0.04	0.00	-0.21	100%
27	clust_red_dl	35000	0.03	0.06	0.04	-0.00	-0.15	100%
28	clust_red_dl	70000	0.03	0.06	0.04	0.00	-0.14	100%
29	clust_red_jac	5000	0.01	0.05	0.02	0.00	-0.27	100%
30	clust_red_jac	10000	0.03	0.06	0.04	0.00	-0.24	100%
31	clust_red_jac	35000	0.03	0.06	0.04	0.00	-0.14	100%
32	clust_red_jac	70000	0.03	0.06	0.04	0.00	-0.15	100%
33	clust_red_qg	5000	0.02	0.05	0.03	0.00	-0.11	100%
34	clust_red_qg	10000	0.03	0.06	0.04	0.00	-0.10	100%
35	clust_red_qg	35000	0.03	0.06	0.04	0.00	-0.05	100%
36	clust_red_qg	70000	0.03	0.06	0.04	0.00	-0.07	100%
37	clust_lcs_red_lcs	5000	0.03	0.06	0.04	0.00	-0.24	100%
38	clust_lcs_red_lcs	10000	0.03	0.06	0.04	0.00	-0.25	100%
39	clust_lcs_red_lcs	35000	0.03	0.06	0.04	0.00	-0.16	100%
40	clust_lcs_red_lcs	70000	0.03	0.06	0.04	0.00	-0.18	100%
41	clust_dl_red_dl	5000	0.03	0.06	0.04	0.00	-0.23	100%
42	clust_dl_red_dl	10000	0.03	0.06	0.04	-0.00	-0.17	100%
43	clust_dl_red_dl	35000	0.03	0.06	0.04	0.00	-0.17	100%
44	clust_dl_red_dl	70000	0.03	0.06	0.04	0.00	-0.16	100%
45	clust_jac_red_jac	5000	0.03	0.06	0.04	0.00	-0.22	100%
46	clust_jac_red_jac	10000	0.03	0.06	0.04	0.00	-0.24	100%
47	clust_jac_red_jac	35000	0.03	0.06	0.04	0.00	-0.15	100%
48	clust_jac_red_jac	70000	0.03	0.06	0.04	0.00	-0.16	100%
49	clust_qg_red_qg	5000	0.03	0.06	0.04	0.00	-0.21	100%
50	clust_qg_red_qg	10000	0.03	0.06	0.04	0.00	-0.23	100%
51	clust_qg_red_qg	35000	0.03	0.06	0.04	0.00	-0.14	100%
52	clust_qg_red_qg	70000	0.03	0.06	0.04	0.00	-0.14	100%

Tabela 3.6: Wyniki dla całego zbioru.

	Typ danych	b	Jedn.	Zg.	Miara V	ARI	Silhouettes	Część
53	clust	5000	0.21	0.57	0.31	0.01	-0.26	15%
54	clust	10000	0.23	0.54	0.32	0.01	-0.15	15%
55	clust	35000	0.22	0.53	0.32	0.02	-0.16	15%
56	clust	70000	0.23	0.53	0.32	0.02	-0.16	15%
57	clust_lcs	5000	0.27	0.58	0.37	0.03	-0.25	15%
58	clust_lcs	10000	0.21	0.53	0.30	0.01	-0.12	15%
59	clust_lcs	35000	0.20	0.54	0.29	0.01	-0.23	15%
60	clust_lcs	70000	0.27	0.56	0.36	0.03	-0.17	15%
61	clust_dl	5000	0.20	0.51	0.29	0.01	-0.17	15%
62	clust_dl	10000	0.22	0.55	0.31	0.01	-0.18	15%
63	clust_dl	35000	0.23	0.53	0.32	0.02	-0.16	15%
64	clust_dl	70000	0.23	0.54	0.32	0.02	-0.24	15%
65	clust_jac	5000	0.28	0.58	0.37	0.03	-0.24	15%
66	clust_jac	10000	0.23	0.53	0.32	0.01	-0.14	15%
67	clust_jac	35000	0.19	0.50	0.28	0.01	-0.17	15%
68	clust_jac	70000	0.22	0.52	0.31	0.02	-0.16	15%
69	clust_qg	5000	0.21	0.49	0.29	0.01	-0.15	15%
70	clust_qg	10000	0.21	0.52	0.30	0.01	-0.17	15%
71	clust_qg	35000	0.23	0.52	0.32	0.02	-0.15	15%
72	clust_qg	70000	0.23	0.51	0.32	0.02	-0.20	15%
73	clust_red_lcs	5000	0.24	0.58	0.34	0.02	-0.27	15%
74	clust_red_lcs	10000	0.27	0.55	0.36	0.03	-0.13	15%
75	clust_red_lcs	35000	0.22	0.55	0.31	0.01	-0.17	15%
76	clust_red_lcs	70000	0.25	0.54	0.34	0.03	-0.16	15%
77	clust_red_dl	5000	0.24	0.53	0.33	0.02	-0.14	15%
78	clust_red_dl	10000	0.24	0.55	0.33	0.02	-0.13	15%
79	clust_red_dl	35000	0.23	0.53	0.33	0.02	-0.15	15%
80	clust_red_dl	70000	0.25	0.55	0.34	0.02	-0.17	15%
81	clust_red_jac	5000	0.20	0.50	0.28	0.01	-0.28	15%
82	clust_red_jac	10000	0.26	0.52	0.34	0.03	-0.17	15%
83	clust_red_jac	35000	0.25	0.52	0.34	0.03	-0.15	15%
84	clust_red_jac	70000	0.20	0.48	0.29	0.01	-0.17	15%
85	clust_red_qg	5000	0.18	0.45	0.26	0.01	-0.10	15%
86	clust_red_qg	10000	0.21	0.43	0.28	0.02	-0.08	15%
87	clust_red_qg	35000	0.22	0.43	0.29	0.02	-0.08	15%
88	clust_red_qg	70000	0.20	0.43	0.27	0.02	-0.10	15%
89	clust_lcs_red_lcs	5000	0.26	0.57	0.36	0.03	-0.25	15%
90	clust_lcs_red_lcs	10000	0.22	0.54	0.31	0.01	-0.17	15%
91	clust_lcs_red_lcs	35000	0.22	0.56	0.32	0.01	-0.17	15%
92	clust_lcs_red_lcs	70000	0.27	0.56	0.36	0.04	-0.18	15%
93	clust_dl_red_dl	5000	0.21	0.53	0.31	0.01	-0.18	15%
94	clust_dl_red_dl	10000	0.24	0.55	0.33	0.01	-0.24	15%
95	clust_dl_red_dl	35000	0.23	0.57	0.33	0.02	-0.17	15%
96	clust_dl_red_dl	70000	0.22	0.54	0.32	0.01	-0.17	15%
97	clust_jac_red_jac	5000	0.21	0.50	0.30	0.01	-0.17	15%
98	clust_jac_red_jac	10000	0.24	0.51	0.32	0.02	-0.22	15%
99	clust_jac_red_jac	35000	0.22	0.51	0.31	0.02	-0.18	15%
100	clust_jac_red_jac	70000	0.24	0.51	0.33	0.02	-0.18	15%
101	clust_qg_red_qg	5000	0.24	0.52	0.32	0.02	-0.22	15%
102	clust_qg_red_qg	10000	0.22	0.48	0.31	0.02	-0.15	15%
103	clust_qg_red_qg	35000	0.24	0.49	0.32	0.03	-0.16	15%
104	clust_qg_red_qg	70000	0.21	0.47	0.29	0.02	-0.17	15%

Tabela 3.7: Wyniki dla ok. 15% zbioru.

	Typ danych	b	Jedn.	Zg.	Miara V	ARI	Silhouettes	Część
105	clust	5000	0.32	0.69	0.44	0.04	-0.17	2%
106	clust	10000	0.34	0.71	0.46	0.08	-0.15	2%
107	clust_lcs	5000	0.35	0.69	0.46	0.06	-0.17	2%
108	clust_lcs	10000	0.33	0.70	0.45	0.07	-0.17	2%
109	clust_dl	5000	0.33	0.69	0.45	0.05	-0.18	2%
110	clust_dl	10000	0.34	0.70	0.45	0.08	-0.17	2%
111	clust_jac	5000	0.32	0.69	0.44	0.05	-0.19	2%
112	clust_jac	10000	0.30	0.72	0.43	0.05	-0.17	2%
113	clust_qg	5000	0.31	0.69	0.43	0.05	-0.18	2%
114	clust_qg	10000	0.32	0.68	0.44	0.08	-0.13	2%
115	clust_red_lcs	5000	0.32	0.69	0.44	0.05	-0.16	2%
116	clust_red_lcs	10000	0.35	0.71	0.47	0.09	-0.13	2%
117	clust_red_dl	5000	0.36	0.70	0.48	0.08	-0.12	2%
118	clust_red_dl	10000	0.34	0.70	0.46	0.08	-0.14	2%
119	clust_red_jac	5000	0.35	0.64	0.45	0.08	-0.12	2%
120	clust_red_jac	10000	0.36	0.66	0.46	0.12	-0.09	2%
121	clust_red_qg	5000	0.31	0.58	0.41	0.07	-0.05	2%
122	clust_red_qg	10000	0.32	0.58	0.41	0.12	-0.08	2%
123	clust_lcs_red_lcs	5000	0.33	0.70	0.45	0.06	-0.17	2%
124	clust_lcs_red_lcs	10000	0.34	0.71	0.46	0.08	-0.11	2%
125	clust_dl_red_dl	5000	0.33	0.69	0.45	0.05	-0.17	2%
126	clust_dl_red_dl	10000	0.34	0.69	0.46	0.09	-0.13	2%
127	clust_jac_red_jac	5000	0.35	0.68	0.47	0.07	-0.13	2%
128	clust_jac_red_jac	10000	0.33	0.69	0.44	0.08	-0.12	2%
129	clust_qg_red_qg	5000	0.33	0.64	0.43	0.07	-0.14	2%
130	clust_qg_red_qg	10000	0.33	0.65	0.43	0.10	-0.13	2%

Tabela 3.8: Wyniki dla ok. 2% zbioru.

Literatura

[1] Wikipedia - wikipedia, wolna encyklopedia. `http://pl.wikipedia.org/wiki/Wikipedia`. Dostęp: 2015-12-01.

[2] Léon Bottou. Stochastic gradient tricks. Grégoire Montavon, Genevieve B. Orr, Klaus-Robert Müller, redaktorzy, *Neural Networks, Tricks of the Trade, Reloaded*, Lecture Notes in Computer Science (LNCS 7700), strony 430–445. Springer, 2012.

[3] Léon Bottou, Yoshua Bengio. Convergence properties of the k-means algorithms. *Advances in Neural Information Processing Systems 7*, strony 585–592. MIT Press, 1995.

[4] Leonid Boytsov. Indexing methods for approximate dictionary searching: Comparative analysis. *J. Exp. Algorithmics*, 16:1.1:1.1–1.1:1.91, 2011.

[5] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Techincal Journal*, 29:147–160, 1950.

[6] Trevor J. Hastie, Robert John Tibshirani, Jerome H. Friedman. *The elements of statistical learning : data mining, inference, and prediction.* Springer series in statistics. Springer, New York, 2009.

[7] Lawrence Hubert, Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[8] J. Koronacki, J. Ćwik. *Statystyczne systemy uczące się.* Wydawnictwa Naukowo-Techniczne, 2005.

[9] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[10] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[11] Andrew Rosenberg, Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning(EMNLP-CoNLL)*, strony 410–420, 2007.

[12] Peter Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987.

[13] D. Sculley. Web-scale k-means clustering. *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, strony 1177–1178, New York, NY, USA, 2010. ACM.

[14] Esko Ukkonen. Algorithms for approximate string matching. *Inf. Control*, 64(1-3):100–118, 1985.

[15] Esko Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 92(1):191 – 211, 1992.

[16] Mark P. J. van der Loo. The stringdist Package for Approximate String Matching. *The R Journal*, 6:111–122, 2014.

[17] Robert A. Wagner, Michael J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.

[18] Robert A. Wagner, Roy Lowrance. An extension of the string-to-string correction problem. *J. ACM*, 22(2):177–183, 1975.

[19] William E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *Proceedings of the Section on Survey Research*, strony 354–359, 1990.

[20] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, 2007.