Rozdział 1

Metryki na przestrzeni ciągów znaków

1.1 Podstawowe definicje

Definicja 1.1.1. Napisem *nazywamy skończone złączenie symboli (znaków) ze skończonego* alfabetu, *oznaczonego przez* Σ. *Produkt kartezjański rzędu* q, $\Sigma \times \ldots \times \Sigma$ *oznaczamy przez* Σ^q, *natomiast zbiór wszystkich skończonych napisów, które można utworzyć ze znaków z* Σ *oznaczamy przez* Σ^*. Pusty napis, *oznaczany* ε, *również należy do* Σ^*. *Napisy zwyczajowo będziemy oznaczać przez* s, t *oraz* u, *a ich* długość, *czyli liczbę znaków w napisie, przez* $|s|$. *Poprzez* s_i *rozumiemy* i*-ty znak z napisu* s, *dla każdego* $i \in \{1, \ldots, |s|\}$, *natomiast podnapisy od znaku* i*-tego do znaku* j*-tego oznaczamy przez* $s_{i:j}$. *Zakładamy również, że jeśli* $j < i$, *to* $s_{i:j} = \varepsilon$ *[6].*

Przykład 1.1.1. *Niech* Σ *będzie alfabetem złożonym z* 26 *małych liter alfabetu łacińskiego oraz niech* $s =' ala'$. *Wówczas mamy* $|s| = 3$, $s \in \Sigma^3$ *oraz* $s \in \Sigma$. *Co więcej, mamy* $s_1 =' a'$, $s_2 =' l'$, $s_3 =' a'$. *Podnapis* $1 : 2$ *napisu* s *to* $s_{1:2} =' al'$.

Odległość $d(s, t)$ pomiędzy dwoma napisami s i t to minimalny koszt ciągu operacji potrzebnego do przetransformowania s w t (i ∞, gdy taki ciąg nie istnieje). Koszt ciągu operacji jest sumą kosztów pojedynczych operacji. Przez operacje rozumiemy skończoną liczbę reguł w formie $\delta(x, y) = a$, gdzie x i y to różne podnapisy, a a to nieujemna liczba rzeczywista. Kiedy już, przy pomocy operacji, podnapis x zostanie przekształcony w napis y, żadne dalsze operacje nie mogą być wykonywane

na y [4].

Zauważmy w szczególności ostatnie ograniczenie, które nie pozwala wielokrotnie przekształcać tego samego podnapisu. DO POPRAWKI!!!: Gdyby pominąć to założenie, każdy system przekształcający napisy spełniałby definicję i stąd odległość między dwoma napisami nie byłaby, w ogólności, możliwa do policzenia [4].

Jeśli dla każdej operacji $\delta(x, y)$, istnieje odpowiednia operacja $\delta(y, x)$ o takim samym koszcie, to odległość jest symetryczna (tj. $d(s, t) = d(t, s)$). Zauważmy również, że:

- $d(s, t) \geq 0$ dla wszystkich napisów s, t,
- $d(s, s) = 0$,
- $d(s, u) \leq d(s, t) + d(t, u)$.

Stąd, jeśli odległość jest symetryczna, przestrzeń napisów tworzy przestrzeń metryczną [4].

Definicja 1.1.2. *Funkcję d nazywamy* metryką *na Σ^*, jeśli ma poniższe własności:*

- $d(s, t) \geq 0$
- $d(s, t) = 0$ *wtedy i tylko wtedy, gdy* $s = t$
- $d(s, t) = d(t, s)$
- $d(s, u) \leq d(s, t) + d(t, u)$,

gdzie s, t, u są napisami z Σ^*.

Odległości na napisach można podzielić na trzy grupy:

- oparte na operacjach edycyjnych (*edit operations*),
- oparte na q-gramach,
- miary heurystyczne.

1.2 Odległości na napisach oparte na operacjach edycyjnych

Metryki oparte na operacjach edycyjnych zliczają minimalną liczbę opercji potrzebnych do przetworzenia jednego napisu w drugi. Najczęściej wymieniamymi operacjami są:

- usunięcie znaku: $\delta(l, \varepsilon)$, tj. usunięcie litery $'l'$, np. $'ala' \rightarrow' aa'$
- wstawienie znaku: $\delta(\varepsilon, l)$, tj. wstawienie litery $'l'$, np. $'ala' \rightarrow' alka'$
- zamiana znaku: $\delta(a, e)$, tj. zamiana litery $'a'$ na $'e'$, np. $'ala' \rightarrow' ela'$
- transpozycja: $\delta(al, la)$, tj. przestawienie dwóch przylegających liter $'a'$ i $'l'$, np. $'ala' \rightarrow' laa'$

Koszt wszystkich powyższych operacji zazwyczaj wynosi 1. Dla wszystkich odległości, które dopuszczają więcej niż jedną operację edycyjną, może być znaczące nadanie wag poszczególnym operacjom, dając na przykład transpozycji mniejszy koszt niż operacji wstawienia znaku. Odległości, dla których takie wagi zostają nadane są zazwyczaj nazywane *uogólnionymi* odległościami [1].

Przykładowe odległości: Hamminga, najdłuższego wspólnego podnapisu (*longest common substring*), Levenshteina, optymalnego dopasowania napisów (*optimal string alignment*), Damareu-Levenshteina. Nie wszystkie z ww. odległości są metrykami.

Definicja 1.2.1. Odległością Hamminga *[2] na* Σ^* *nazywamy:*

$$d_{hamming}(s,t) = \begin{cases} \sum_{i=1}^{|s|}[1-\delta(s_i,t_i)], & \textit{gdy } |s| = |t|, \\ \infty, & \textit{w przeciwnym przypadku}, \end{cases}$$

gdzie

$$\delta(s_i,t_i) = \begin{cases} 1, & \textit{gdy } s_i = t_i, \\ 0, & \textit{w przeciwnym przypadku}. \end{cases}$$

Łatwo zauważyć, że odległość Hamminga spełnia definicję metryki. Intuicyjnie rzecz biorąc odległość Hamminga zlicza liczbę indeksów, na których dwa napisy mają różny znak.

[PIĘKNY RYSUNEK??]

Przykład 1.2.1. *Odległość Hamminga między słowami* koza *i* foka *wynosi* $d_{hamming}(koza, foka) = 2$.

Definicja 1.2.2. Odległością najdłuższego wspólnego podnapisu *[5] na* Σ^* *nazywamy:*

$$d_{lcs}(s,t) = \begin{cases} 0, & \textit{gdy } s = t = \varepsilon, \\ d_{lcs}(s_{1:|s|-1}, t_{1:|t|-1}), & \textit{gdy } s_{|s|} = t_{|t|}, \\ 1 + min\{d_{lcs}(s_{1:|s|-1}, t), d_{lcs}(s, t_{1:|t|-1})\}, & \textit{w przeciwnym przypadku}, \end{cases}$$

Odległość najdłuższego wspólnego podnapisu również spełnia definicję metryki. Przyjmuje wartości z przedziału $[0, |s| + |t|]$, przy czym maksimum jest osiągane, gdy s i t nie mają ani jednego wspólnego znaku. Odległość ta zlicza liczbę usunięć i wstawień, potrzebnych do przetworzenia jednego napisu w drugi. Np. $d_{lsc}('koza','foka') = 4$, bo $koza \xrightarrow{us.\ k} oza \xrightarrow{us.\ z} oa \xrightarrow{wst.\ f} foa \xrightarrow{wst.\ k} foka$.

Powyższy przykład pokazuje, że w ogólności nie ma unikalnej najkrótszej drogi transformacji jednego napisu w drugi, gdyż można zamienić kolejność usuwania (lub wstawiania) znaków i również uzyskać odlegość równą 4.

Jak sugeruje nazwa, odległość najdłuższego wspólnego podnapisu, ma też inną interpretację. Poprzez wyrażenie *najdłuższy wspólny podnapis* rozumiemy najdłuższy ciąg utworzony przez sparowanie znaków z s i t nie zmieniając ich porządku. Wówczas odległość ta jest rozumiana jako liczba niesparowanych znaków z obu napisów. W powyższym przykładzie może to być zwizualizowane następująco:

[PIĘKNY RYSUNEK??]

Jak widać na rysunku, litery $'k'$, $'z'$, $'f'$ i $'k'$ pozostają bez pary, dając odległość równą 4.

Definicja 1.2.3. *Uogólnioną* odległością Levenshteina *[3] na* Σ^* *nazywamy:*

$$d_{lv}(s,t) = \begin{cases} 0, & \text{gdy } s = t = \varepsilon, \\ min\{ & \\ \quad d_{lv}(s, t_{1:|t|-1}) + w_1, & \\ \quad d_{lv}(s_{1:|s|-1}, t) + w_2, & \\ \quad d_{lv}(s_{1:|s|-1}, t_{1:|t|-1}) + [1 - \delta(s_{|s|}, t_{|t|})]w_3 & \\ \quad \}, & \text{w przeciwnym przypadku}, \end{cases}$$

gdzie w_1, w_2 *i* w_3 *to niezerowe liczby rzeczywiste, oznaczające kary za usunięcie, wstawienie oraz zamianę znaku.*

Odległość ta zlicza ważoną sumę usunięć, wstawień oraz zamian znaków, potrzebnych do przetworzenia jednego napisu w drugi. Gdy za wagi przyjmuje się 1 mamy do czynienia ze zwykłą odległością Levenshteina, np. $d_{lv}('koza', 'foka') = 2$, bo $koza \xrightarrow{zm.\ k\ na\ f} foza \xrightarrow{zm.\ z\ na\ k} foka$. Powyższy przykład ilustruje, że dodatkowa elastyczność w odniesieniu do odległości najdłuższego wspólnego podnapisu, daje mniejszą wartość odległości między napisami, jako że potrzebujemy jedynie dwóch zamian znaków [6].

Gdy za wagi przyjmiemy np. $(0.1, 1, 0.3)$, to $d_{lv}('koza', 'foka') = 0.6$, bo $koza \xrightarrow[0.3]{zm.\ k\ na\ f} foza \xrightarrow[0.3]{zm.\ z\ na\ k} foka$.

Uogólniona odległość Levenshteina spełnia definicję metryki, gdy $w_1 = w_2$. W przeciwnym przypadku nie spełnia ona założenia o symetrii. Jednakowoż, symetria zostaje zachowana przy jednoczesnej zamianie s i t oraz w_1 i w_2, jako że liczba usunięć znaków przy przetwarzaniu napisu s w napis t jest równa liczbie wstawień znaków przy przetwarzaniu napisu t w napis s [6]. Dobrze obrazuje to następujący przykład.

Przykład 1.2.2. *Przyjmijmy za* $(w_1, w_2, w_3) = (0.1, 1, 0.3)$. *Wówczas uogólniona odległość Levenshteina dla napisów* $'koza'$ *i* $'foczka'$ *wynosi:*

$$d_{lv}('koza', 'foczka') = 0.5, \tag{1.1}$$

bo

$$koza \xrightarrow[0.3]{zm.\ k\ na\ f} foza \xrightarrow[0.1]{wst.c} focza \xrightarrow[0.1]{wst.k} foczka,$$

natomiast

$$d_{lv}('foczka','koza') = 2.3, \tag{1.2}$$

bo

$$foczka \xrightarrow[0.3]{zm.\ f\ na\ k} koczka \xrightarrow[1]{us.c} kozka \xrightarrow[1]{us.k} koza.$$

Gdy za wagi (w_1, w_2, w_3) *przyjmiemy* $(1, 0.1, 0.3)$*, to uogólniona odległość Levenshteina wynosi:*

$$d_{lv}('koza','foczka') = 2.3,$$

bo

$$koza \xrightarrow[0.3]{zm.\ k\ na\ f} foza \xrightarrow[1]{wst.c} focza \xrightarrow[1]{wst.k} foczka,$$

czyli analogicznie, jak w przypadku 1.2. Natomiast

$$d_{lv}('foczka','koza') = 0.5,$$

bo

$$foczka \xrightarrow[0.3]{zm.\ f\ na\ k} koczka \xrightarrow[0.1]{us.c} kozka \xrightarrow[0.1]{us.k} koza,$$

czyli analogicznie, jak w przypadku 1.1.

Definicja 1.2.4. *Odległością optymalnego dopasowania napisów na* Σ^* *nazywamy:*

$$d_{osa}(s,t) = \begin{cases} 0, & gdy\ s = t = \varepsilon, \\ min\{ & \\ \quad d_{osa}(s, t_{1:|t|-1}) + w_1, & \\ \quad d_{osa}(s_{1:|s|-1}, t) + w_2, & \\ \quad d_{osa}(s_{1:|s|-1}, t_{1:|t|-1}) + [1 - \delta(s_{|s|}, t_{|t|})]w_3 & \\ \quad d_{osa}(s_{1:|s|-2}, t_{1:|t|-2}) + w_4, gdy\ s_{|s|} = t_{|t|-1}, s_{|s|-1} = t_{|t|} & \\ \quad \}, & w\ przeciwnym\ przypadku, \end{cases}$$

gdzie w_1, w_2*,* w_3 *iw*$_4$ *to niezerowe liczby rzeczywiste, oznaczające kary za odpowiednio usunięcie, wstawienie, zamianę oraz transpozycję znaków.*

Metryka **optymalnego dopasowania napisów**, ozn. d_{osa}, zlicza liczbę usunięć, wstawień, zamian oraz transpozycji przylegających znaków, potrzebnych do przetworzenia jednego napisu w drugi. Np. $d_{osa}('leia','leela') = 2$, bo $leela \xrightarrow{us.\ e} lela \xrightarrow{zm.\ l\ na\ i}$

$leia$.

Metryka ta nie spełnia nierówności trójkąta: $2 = d_{osa}('ba','ab') + d_{osa}('ab','acb') \leq d_{osa}('ba','acb') = 3$

Bibliografia

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

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

[3] V. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission*, 1:8–17, 1965.

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

[5] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, March 1970.

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

Warszawa, dnia .....

Oświadczenie

Oświadczam, że pracę licencjacką pod tytułem: „Automatyczna kategoryzacja tematyczna tekstów przy użyciu metryk w przestrzeni ciągów znaków", której promotorem jest dr hab. Marek Gągolewski, wykonałem/am samodzielnie, co poświadczam własnoręcznym podpisem.

.....