Politechnika Warszawska
Wydział Matematyki i Nauk Informacyjnych

PRACA DYPLOMOWA MAGISTERSKA
NA KIERUNKU MATEMATYKA

AUTOMATYCZNA KATEGORYZACJA TEMATYCZNA TEKSTÓW PRZY UŻYCIU METRYK W PRZESTRZENI CIĄGÓW ZNAKÓW

Autor:
Natalia Potocka

Promotor:
dr Marek Gągolewski

Warszawa, grudzień 2015

.....

podpis promotora

.....

podpis autora

Spis treści

Rozdział 1

Metryki na przestrzeni ciągów znaków

1.1. Podstawowe definicje

CZY TO JAKO CIĄGŁY TEKST CZY WSZYSTKO PISAĆ W DEFINICJI???

Definicja 1.1. *Niech $\Sigma = \{\Sigma_i\}$ będzie skończonym uporządkowanym alfabetem o wielkości $|\Sigma|$.* Napisem *nazywamy skończony ciąg znaków z Σ. Zbiór wszystkich napisów o długości n nad Σ jest oznaczony przez Σ^n, podczas gdy przez $\Sigma^* = \bigcup_{n=1}^{\infty} \Sigma^n$ rozumiemy zbiór wszystkich napisów utworzonych ze znaków z Σ [1].*

O ile nie podano inaczej, używamy zmiennych s, t, u, v, w, x, y jako oznaczenie napisów oraz a, b, c do oznaczenia napisów jednoznakowych albo po prostu znaków*. Pusty napis jest oznaczany jako ε. Przez $|s|$, dla każdego napisu $s \in \Sigma^*$, rozumiemy jego długość, czyli liczbę znaków w napisie. Ciąg zmiennych oznaczających napisy i/lub znaki oznaczają ich złączenie [1].*

Poprzez s_i rozumiemy i-ty znak z napisu s, dla każdego $i \in \{1, \ldots, |s|\}$. Podciąg kolejnych przylegających do siebie znaków z napisu nazywamy podnapisem*. Podnapisem napisu s, który zaczyna się od i-tego znaku, a kończy na j-tym znaku, oznaczamy przez $s_{i:j}$, tj. $s_{i:j} = s_i s_{i+1} \ldots s_j$ dla $i < j$. Zakładamy również, że jeśli $j < i$, to $s_{i:j} = \varepsilon$ [1, 6].*

Załóżmy, że napis s jest resprezentacją złączenia trzech, być może pustych, podnapisów w, x i y, tj. $s = wxy$. Wówczas podnapis w nazywamy prefiksem*, natomiast podnapis y – sufiksem [1].*

Podnapis o ustalonej długości q jest nazywany q-gramem*. q-gramy o q równym jeden, dwa lub trzy mają specjalne nazwy:* unigram, bigram *i* trigram*. Jeśli $q > |s|$, to q-gramy napisu są naisami pustymi [1].*

Przykład 1.1. Niech Σ będzie alfabetem złożonym z 26 małych liter alfabetu łacińskiego oraz niech $s =' ela'$. Wówczas mamy $|s| = 3$, $s \in \Sigma^3$ oraz $s \in \Sigma$. Co więcej, mamy $s_1 ='$

e', $s_2 =' l'$, $s_3 =' a'$. Podnapis $1:2$ napisu s to $s_{1:2} =' el'$. W napisie tym mamy do czynienia jedynie z q-gramami o q równym jeden, dwa oraz trzy: $'e'$, $'l'$, $'a'$; $'el'$, $'la'$ oraz $'ela'$ odpowiednio.

[TU TRZEBA BARDZIEJ FORMALNIE! BOYTSOV STR. 4-7]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 1.2. *Funkcję d nazywamy* metryką *na Σ^*, jeśli ma poniższe własności:*

1. $d(s,t) \geq 0$
2. $d(s,t) = 0$ *wtedy i tylko wtedy, gdy* $s = t$
3. $d(s,t) = d(t,s)$
4. $d(s,u) \leq d(s,t) + d(t,u)$,

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

HISTORIA ODLEGLOSCI EDYCYJNYCH?

Odległość edycyjna $ED(s,t)$ pomiędzy dwoma napisami s i t to minimalna liczba operacji edycyjnych potrzebna do przetworzenia s w t (i ∞, gdy taki ciąg nie istnieje) [4]. *Ścisłą odległością edycyjną* nazywamy minimalną liczbę nie nakładających się operacji edycyjnych,

które pozwalają przekształcić jeden napis w drugi, i które nie przekształcają dwa razy tego samego podnapisu [1].

Napis może zostać przetworzony w drugi poprzez ciąg przekształcenia jego podnapisów. Ten ciąg nazywany jest *śladem edycji*, podczas gdy przekształcenia są nazywane *bazowymi* operacjami edycyjnymi. Bazowe operacje edycyjne, które polegają na mapowaniu napisu s w napis t, są oznaczane przez $s \to t$. Zbiór wszystkich bazowych operacji edycyjnych oznaczamy przez $\mathbb{B}$ [1].

Bazowe operacje edycyjne są zazwyczaj ograniczone do:

- usunięcie znaku: $l \to \varepsilon$, tj. usunięcie litery $'l'$, np. $'ela' \to' ea'$
- wstawienie znaku: $\varepsilon \to l$, tj. wstawienie litery $'l'$, np. $'ela' \to' elka'$
- zamiana znaku: $e \to a$, tj. zamiana litery $'e'$ na $'a'$, np. $'ala' \to' ela'$
- transpozycja: $el \to le$, tj. przestawienie dwóch przylegających liter $'e'$ i $'l'$, np. $'ela' \to' lea'$

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

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

gdzie

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

Łatwo zauważyć, że odległość Hamminga spełnia definicję metryki. Intuicyjnie rzecz biorąc odległość Hamminga zlicza liczbę indeksów, na których dwa napisy mają różny znak. Odległość ta przyjmuje wartości ze zbioru $\{0,\ldots,|s|\}$, gdy $|s| = |t|$, natomiast jest równa nieskończoności, gdy napisy mają różne długości.

[PIĘKNY RYSUNEK??]

Przykład 1.2. Odległość Hamminga między słowami *koza* i *foka* wynosi $d_{hamming}(koza, foka) = 2$, natomiast między słowami *koza* i *foczka* wynosi ona $d_{hamming}(koza, foczka) = \infty$.

Definicja 1.4. Odległością najdłuższego wspólnego podnapisu *[5] na* Σ^* *nazywamy:*

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

Odległość najdłuższego wspólnego podnapisu również spełnia definicję metryki. Przyjmuje wartości ze zbioru $\{0, |s| + |t|\}$, 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.

Przykład 1.3. Odległość najdłuższego wspólnego podnapisu między słowami *koza* i *foka* wynosi: $d_{lsc}('koza', 'foka') = 4$, bo $koza \xrightarrow[1]{us.\ k} oza \xrightarrow[1]{us.\ z} oa \xrightarrow[1]{wst.\ f} foa \xrightarrow[1]{wst.\ k} foka$.

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

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

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 przyjmie się 1 mamy do czynienia ze zwykłą odległością Levenshteina, np. $d_{lv}('koza', 'foka') = 2$, bo $koza \xrightarrow[1]{zm.\ k\ na\ f} foza \xrightarrow[1]{zm.\ z\ na\ k} foka$. Powyższy przykład ilustruje, że dodatkowa elastyczność w porównaniu 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, tj. podpunktu 3 definicji 1.2. 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 1.4. 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 1.6. 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.*

Odległość optymalnego dopasowania napisów jest bezpośrednim rozszerzeniem odległości Levenshteina, która zlicza również liczbę transpozycji przylegających znaków, potrzebnych do przetworzenia jednego napisu w drugi. W przeciwieństwie do wcześniej zaprezentowanych odległości, nie spełnia ona nierówności trójkąta, tj. podpunktu 4 z definicji 1.2 [6]:

$$2 = d_{osa}('ba','ab') + d_{osa}('ab','acb') \leq d_{osa}('ba','acb') = 3,$$

gdyż

$$ba \xrightarrow[1]{transp.\ b\ i\ a} ab + ab \xrightarrow[1]{wst.c} acb,$$

natomiast

$$ba \xrightarrow[1]{us.\ b} a \xrightarrow[1]{wst.\ c} ac \xrightarrow[1]{wst.\ b} acb.$$

W ostatnim przykładzie, zmniejszenie odległości poprzez zamianę liter $'a'$ i $'b'$, a następnie wstwienie litery $'c'$ spowodowałoby dwukrotne przekształcenie tego samego podnapisu. Z tego powodu odległość optymalnego dopasowania napisów bywa również nazywana *ścisłą odległością Damerau-Levenshteina* i jest często mylona z właściwą *odległością Damerau-Levenshteina*. Ta ostatnia pozwala na przekształcanie tego samego podnapisu wielokrotnie i jest metryką w rozumieniu definicjij 1.2, ale nie spełnia założenia o nie przekształcaniu wielokrotnie tego samego podnapisu [6].

[MIARA DAMERAU-LEVENSHTEINA??? WTEDY ZMIENIC DEFINICJE O NIEPRZERABIANIU 2 RAZY TEGO SAMEGO PODNAPISU]

W przypadku odległości Levenshteina i odległości optymalnego dopasowania napisów, maksymalna odległość między napisami s i t wynosi $max\{|s|, |t|\}$. Jednakowoż, gdy liczba dopuszczalnych operacji edycyjnych rośnie, to liczba dopuszczalnych ścieżek między napisami wzrasta, co pozwala ewentualnie zmiejszyć odległość między napisami. Dlatego relację między zaprezentowanymi powyżej odległościami można podsumować następująco [6]:

$$\left.\begin{array}{r} \infty \geq |s| \geq d_{hamming}(s,t) \\ |s| + |t| \geq d_{lcs}(s,t) \\ max\{|s|, |t|\} \end{array}\right\} \geq d_{lv}(s,t) \geq d_{osa}(s,t) \geq 0.$$

Literatura

[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.

Natalia Potocka
Nr albumu 237476

Warszawa, 4 września 2015

Oświadczenie

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

.....
Natalia Potocka