



Politechnika Warszawska
Wydział Matematyki i Nauk Informacyjnych



Automatyczna kategoryzacja tematyczna tekstów przy użyciu metryk w przestrzeni ciągów znaków

Natalia Potocka
Warszawa, 21.04.2014

- Cel pracy
- O metrykach słów kilka
- Postęp prac
- Co dalej?

Celem pracy jest skategoryzowanie tekstów z polskiej Wikipedii pod względem tematu na podstawie liczności słów występujących w tekście. Można się spodziewać, że jeśli w dwóch tekstach występuje dużo podobnych do siebie słów, to pochodzą one z tej samej kategorii tematycznej.

Celem pracy jest skategoryzowanie tekstów z polskiej Wikipedii pod względem tematu na podstawie liczności słów występujących w tekście. Można się spodziewać, że jeśli w dwóch tekstach występuje dużo podobnych do siebie słów, to pochodzą one z tej samej kategorii tematycznej.

A		B		C		D	
całka	10	całka	5	niewłaściwe	3	ułamek	4
po pochodna	5	po pochodna	15	powieść	7	mianownik	5
niewłaściwa	4	granica	7	granica	15	niewłaściwy	6

Co ze słowami podobnymi?

Przykładowo słowa *niewłaściwy* i *niewłaściwa* mają ten sam temat, różnią się tylko rodzajem (męski / żeński). W tekstach mogą też występować błędy ortograficzne, błędy spowodowane brakami znaków diaktrycznych (ą, ę, ł, ...) itd. Takie słowa również chcielibyśmy traktować jako „podobne”. W celu określenia jak bardzo dwa słowa są do siebie podobne, posłużą *metryki określone na napisach*.

DEFINICJA

Napisem nazywamy skończone złączenie symboli (znaków) ze skończonego *alfabetu*, oznaczonego przez Σ . Produkt kartezjański rzędu q , $\Sigma \times \dots \times \Sigma$ 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 $|s|$.

DEFINICJA

Napisem nazywamy skończone złączenie symboli (znaków) ze skończonego *alfabetu*, oznaczonego przez Σ . Produkt kartezjański rzędu q , $\Sigma \times \dots \times \Sigma$ 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 $|s|$.

Przykład. Niech Σ będzie alfabetem złożonym z 26 małych liter alfabetu łacińskiego oraz niech $s = 'ala'$. Wówczas mamy $|s| = 3$, $s \in \Sigma^3$ oraz $s \in \Sigma$. Pojedyncze znaki oznaczamy przez indeks dolny, stąd mamy $s_1 = 'a'$, $s_2 = 'l'$, $s_3 = 'a'$. [2]

DEFINICJA

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

- $d(s, t) \geq 0$
- $d(s, t) = 0$ wtw $s = t$
- $d(s, t) = d(t, s)$
- $d(s, u) \leq d(s, t) + d(t, u)$,

gdzie s, t, u są napisami.

DEFINICJA

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

- $d(s, t) \geq 0$
- $d(s, t) = 0$ wtw $s = t$
- $d(s, t) = d(t, s)$
- $d(s, u) \leq d(s, t) + d(t, u)$,

gdzie s, t, u są napisami.

Nie wszystkie metryki na napisach posiadają wszystkie z wyżej wymienionych własności.

DEFINICJA

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

- $d(s, t) \geq 0$
- $d(s, t) = 0$ wtw $s = t$
- $d(s, t) = d(t, s)$
- $d(s, u) \leq d(s, t) + d(t, u)$,

gdzie s, t, u są napisami.

Nie wszystkie metryki na napisach posiadają wszystkie z wyżej wymienionych własności.

Metryki na napisach można podzielić na trzy grupy:

- oparte na operacjach edytowania (*edit operations*)
- oparte na q -gramach
- miary heurystyczne

DEFINICJA

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

- $d(s, t) \geq 0$
- $d(s, t) = 0$ wtw $s = t$
- $d(s, t) = d(t, s)$
- $d(s, u) \leq d(s, t) + d(t, u)$,

gdzie s, t, u są napisami.

Nie wszystkie metryki na napisach posiadają wszystkie z wyżej wymienionych własności.

Metryki na napisach można podzielić na trzy grupy:

- **oparte na operacjach edytowania** (*edit operations*)
- oparte na q -gramach
- miary heurystyczne

Metryki oparte na operacjach edytowania zliczają liczbę operacji potrzebnych do przetworzenia jednego napisu w drugi. Najczęściej wymienianymi operacjami są:

- zamiana znaku, np. $'ala' \rightarrow 'ela'$
- usunięcie znaku, np. $'ala' \rightarrow 'aa'$
- wstawienie znaku, np. $'ala' \rightarrow 'alka'$
- transpozycja dwóch przylegających znaków, np. $'ala' \rightarrow 'laa'$

Metryki oparte na operacjach edytowania zliczają liczbę operacji potrzebnych do przetworzenia jednego napisu w drugi. Najczęściej wymienianymi operacjami są:

- zamiana znaku, np. $'ala' \rightarrow 'ela'$
- usunięcie znaku, np. $'ala' \rightarrow 'aa'$
- wstawienie znaku, np. $'ala' \rightarrow 'alka'$
- transpozycja dwóch przylegających znaków, np. $'ala' \rightarrow 'laa'$

Przykładowe metryki: Hamminga, najdłuższego wspólnego podnapisu (*longest common substring*), Levenshteina, optymalnego dopasowania napisów (*optimal string alignment*), Damareu-Levenshteina.

Metryka **najdłuższego wspólnego podnapisu**, ozn. d_{lcs} , zlicza liczbę usunięć i wstawień, potrzebnych do przetworzenia jednego napisu w drugi.

Np. $d_{lsc}('leia', 'leela') = 3$, bo $leela \xrightarrow{us. e} lela \xrightarrow{us. l} lea \xrightarrow{wst. i} leia$.

Metryka **najdłuższego wspólnego podnapisu**, ozn. d_{lcs} , zlicza liczbę usunięć i wstawień, potrzebnych do przetworzenia jednego napisu w drugi.

Np. $d_{lsc}('leia', 'leela') = 3$, bo $leela \xrightarrow{us. e} lela \xrightarrow{us. l} lea \xrightarrow{wst. i} leia$.

Uogólniona **odległość Levenshteina**, ozn. d_{lv} zlicza ważoną sumę usunięć, wstawień oraz zamian znaków, potrzebnych do przetworzenia jednego napisu w drugi.

Metryka **najdłuższego wspólnego podnapisu**, ozn. d_{lcs} , zlicza liczbę usunięć i wstawień, potrzebnych do przetworzenia jednego napisu w drugi.

Np. $d_{lsc}('leia', 'leela') = 3$, bo $leela \xrightarrow{us. e} lela \xrightarrow{us. l} lea \xrightarrow{wst. i} leia$.

Uogólniona **odległość Levenshteina**, ozn. d_{lv} 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.

$d_{lv}('leia', 'leela') = 2$, bo $leela \xrightarrow{us. e} lela \xrightarrow{zm. l \text{ na } i} leia$.

Metryka **najdłuższego wspólnego podnapisu**, ozn. d_{lcs} , zlicza liczbę usunięć i wstawień, potrzebnych do przetworzenia jednego napisu w drugi.

Np. $d_{lsc}('leia', 'leela') = 3$, bo $leela \xrightarrow{us. e} lela \xrightarrow{us. l} lea \xrightarrow{wst. i} leia$.

Uogólniona **odległość Levenshteina**, ozn. d_{lv} 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.

$d_{lv}('leia', 'leela') = 2$, bo $leela \xrightarrow{us. e} lela \xrightarrow{zm. l \ na \ i} leia$.

Gdy za wagi przyjmujemy np. $(0.1, 1, 1)$,

$d_{lv}('leia', 'leela') = 1.1$, bo $leela \xrightarrow[0.1]{us. e} lela \xrightarrow[1]{zm. l \ na \ i} leia$

Metryka **optymalnego dopasowania napisów**, ozn. d_{osa} , zlicza liczbę usunięć, wstawień, zamian oraz transpozycji przylegających znaków, potrzebnych do przetworzenia jednego napisu w drugi. Np.

$$d_{osa}('leia', 'leela') = 2, \text{ bo } leela \xrightarrow{us. e} lela \xrightarrow{zm. l na i} leia.$$

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:

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

Co zostało zrobione?

- wczytano 1 075 568 artykułów z polskiej Wikipedii

Co zostało zrobione?

- wczytano 1 075 568 artykułów z polskiej Wikipedii
- razem to 2 806 765 różnych słów...

Co zostało zrobione?

- wczytano 1 075 568 artykułów z polskiej Wikipedii
- razem to 2 806 765 różnych słów...
- ... z czego 49% wystąpiło tylko w **jednym** tekście
- ... a 44% wystąpiło tylko **jeden raz** we wszystkich tekstach

Co zostało zrobione?

- wczytano 1 075 568 artykułów z polskiej Wikipedii
- razem to 2 806 765 różnych słów...
- ... z czego 49% wystąpiło tylko w **jednym** tekście
- ... a 44% wystąpiło tylko **jeden raz** we wszystkich tekstach

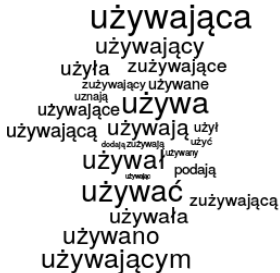
Po usunięciu tzw. *stopwords*, czyli słów nieistotnych w kontekście analizy, jak np. *a, bo, co, jak, to, w, z, że*, słów jednoliterowych oraz słów w językach obcych z niełacińskiego alfabetu, pozostało 2 805 858 słów do analizy.

Co zostało zrobione?

- wczytano 1 075 568 artykułów z polskiej Wikipedii
- razem to 2 806 765 różnych słów...
- ... z czego 49% wystąpiło tylko w **jednym** tekście
- ... a 44% wystąpiło tylko **jeden raz** we wszystkich tekstach

Po usunięciu tzw. *stopwords*, czyli słów nieistotnych w kontekście analizy, jak np. *a, bo, co, jak, to, w, z, że*, słów jednoliterowych oraz słów w językach obcych z niełacińskiego alfabetu, pozostało 2 805 858 słów do analizy.

Początkowy pomysł polegał na wykorzystaniu wcześniej wspomnianych metryk do klastrowania słów metodą k-medoidów, przy czym maksymalna odległość w klastrze miała nie przekraczać zadanej liczby.



RYSUNEK : Przykładowe klastry utworzone przy pomocy metryki *osa*.
Maksymalna odległość w klastrze to 7.

Z powodu słabej jakości klastrowania oraz braku możliwości obliczeniowych dokonano klastrowania przy pomocy tzw. *stemmingu*. Polega on na przyporządkowaniu do słowa jego rdzenia, a więc takiej jego części, która jest odporna na odmiany przez rodzaje, przyimki, przypadki itd. Przykładowo dla słowa *używająca* rdzeniem jest *żyw*.

Z powodu słabej jakości klastrowania oraz braku możliwości obliczeniowych dokonano klastrowania przy pomocy tzw. *stemmingu*. Polega on na przyporządkowaniu do słowa jego rdzenia, a więc takiej jego części, która jest odporna na odmiany przez rodzaje, przyimki, przypadki itd. Przykładowo dla słowa *używająca* rdzeniem jest *żyw*.

Do stemmingu użyto narzędzia Hunspell, które sprawdza pisownię dla wielu programów, takich jak: OpenOffice, Mozilla Firefox, Thunderbird czy Google Chrome.

Dzięki niemu udało się poklastrować 733 828 słów ($\approx 26\%$ wszystkich) z czego 89% stanowiły polskie słowa 5,5% - słowa angielskie, a po ponad 2% - słowa francuskie i niemieckie. Innych języków nie sprawdzano. Liczba uzyskanów klastrów to 186 942.

Z powodu słabej jakości klastrowania oraz braku możliwości obliczeniowych dokonano klastrowania przy pomocy tzw. *stemmingu*. Polega on na przyporządkowaniu do słowa jego rdzenia, a więc takiej jego części, która jest odporna na odmiany przez rodzaje, przyimki, przypadki itd. Przykładowo dla słowa *używająca* rdzeniem jest *żyw*.

Do stemmingu użyto narzędzia Hunspell, które sprawdza pisownię dla wielu programów, takich jak: OpenOffice, Mozilla Firefox, Thunderbird czy Google Chrome.

Dzięki niemu udało się poklastrować 733 828 słów ($\approx 26\%$ wszystkich) z czego 89% stanowiły polskie słowa 5,5% - słowa angielskie, a po ponad 2% - słowa francuskie i niemieckie. Innych języków nie sprawdzano. Liczba uzyskanów klastrów to 186 942.

Co z pozostałymi słowami?

Słowa, które wystąpiły więcej niż raz we wszystkich tekstach, dołączono do już istniejących klastrów przy pomocy metryk. Takich słów było 973 855, co dało łącznie poklastrowanych słów w liczbie 1 707 683.

Następnie dla próbki tekstów z trzech kategorii: matematyka, historia sztuki oraz wojny, dokonano klasteryzacji artykułów. Kryterium była liczność **grup słów** występujących w danym tekście. Do klastrowania użyto metody *sferycznych k -średnich*.

Następnie dla próbki tekstów z trzech kategorii: matematyka, historia sztuki oraz wojny, dokonano klasteryzacji artykułów. Kryterium była liczność **grup słów** występujących w danym tekście. Do klastrowania użyto metody *sferycznych k -średnich*.

PRZYPOMNIENIE

W metodzie k -średnich minimalizujemy

$$\sum_i d(x_i, p_{c(i)}),$$

gdzie x_i to zbiór wektorów cech, $c(i) \in \{1, \dots, k\}$ to identyfikator klastra, p_1, \dots, p_k to środek klastra, a d to odległość euklidesowa.

W metodzie k -średnich minimalizujemy

$$\sum_i d(x_i, p_{c(i)}),$$

gdzie x_i to zbiór wektorów cech, $c(i) \in \{1, \dots, k\}$ to identyfikator klastra, p_1, \dots, p_k to środek klastra, a d to odległość euklidesowa.

METODA SFERYCZNA

W metodzie sferycznych k -średnich minimalizujemy [1, 3]

$$\sum_i d(x_i, p_{c(i)}) = \sum_i 1 - \cos(x_i, p_{c(i)}) = \sum_i 1 - \frac{\langle x_i, p_{c(i)} \rangle}{\|x_i\| \cdot \|p_{c(i)}\|},$$

Opierając się na kategoriach z Wikipedii, poprawnie sklasyfikowanych zostało 61% z 59 403 artykułów.

tytuł	kat	id_kat	kl
kościół św. rocha w poznaniu	szt	1	1
portret	szt	1	2
quantum of solace (gra komputerowa)	szt	1	2
kurka wodna (seria gier)	szt	1	2
technika macierzy rzadkich	mat	2	2
kryterium walda	mat	2	2
generalized markup language	mat	2	2
czesław falkiewicz	woj	3	3
william goodenough	woj	3	3
kazimierz gallas	woj	3	3
wacław krzywiec	woj	3	3
fabian aleksandrowicz	woj	3	3

CO DALEJ?

- znaleźć metodę odpowiednią do poklastrowania wszystkich artykułów (SGD?), której jakość byłaby zadowalająca
- rozmyć klastrowanie
- napisać pracę :)

- [1] Martin Kober Christian Buchta Kurt Hornik, Ingo Feinerer. Spherical k-means clustering. *Journal of Statistical Software*, 50:1–22, 2012.
<http://bach-s49.wu-wien.ac.at/4000/1/paper.pdf>.
- [2] Mark P. J. van der Loo. The stringdist package for approximate string matching. *The R Journal*, 6:111–122, 2014.
<http://journal.r-project.org/archive/2014-1/loo.pdf>.
- [3] Stefan Wild. *Seeding Non-Negative Matrix Factorizations with the Spherical K-Means Clustering*. University of Colorado, Colorado, 2002.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.1623&rep=>

Dziękuję za uwagę.