Przetwarzanie Języka Naturalnego
Lab 6 – Wektorowa i grafowa reprezentacja tekstu

Aleksander Smywiński-Pohl

Wydział IEiT
Katedra Informatyki

05.04.2017

Reprezentacja wektorowa tekstu (VSM – Vector-Space Model)

- każda składowa wektora odpowiada jednemu słowu (tokenowi)
- wymaga ustalenia pewnego słownika (bazy przestrzeni)
- umożliwia matematyczne traktowanie tekstu
- zachowuje informację o częstotliwości występowania słów
- traci informację o kolejności słów oraz o gramatyce
- „Ala ma kota” i „Kot ma Alę” (po sprowadzeniu do form podstawowych) mają taką samą reprezentację
- stosowane głównie w przypadku kolekcji dokumentów

Reprezentacja „bag of words”

- najprostsza reprezentacja wektorowa tekstu
- wartość składowej jest równa liczbie wystąpień danego słowa w tekście
- największą wagę mają słowa występujące najczęściej ALE niosą najmniej informacji

Funkcje tłumiące (spłaszczające)

- zmniejszają wpływ bardzo częstych termów na reprezentację wektorową
- najczęściej stosowane: pierwiastek, logarytm
- może to być dowolna (monotoniczna?) funkcja o pochodnej z przedziału $[0; 1]$

Inne reprezentacje wektorowe

W obliczaniu składowych wektora można posłużyć się następującymi współczynnikami:

- tf – *term frequency*, liczba wystąpień termu w tekście
- df – *document frequency*, liczba dokumentów, w których występuje term
- cf – *collection frequency*, liczba wystąpień termu w całym korpusie

Schemat wagowy tf-idf

Term frequency - inversed document frequency:
t – term (słowo)
d – dokument
N – liczba wszystkich dokumentów
w – waga (składowa wektora)
$w(t,d) = tf(t,d) * log(\frac{N}{df(t)})$

Graf odległości (*distance graph*)

$\mathcal{C}$ – korpus

D – dokument ($\in \mathcal{C}$)

k – rząd ($\in \mathbb{N}$)

$G(\mathcal{C}, D, k) = (N(\mathcal{C}), A(D, k))$

Graf odległości (c.d.)

$N(\mathcal{C})$ – zbiór wierzchołków – każdemu słowu z $\mathcal{C}$ odpowiada jeden wierzchołek

$A(D, k)$ – zbiór krawędzi – wierzchołki (słowa) x i y łączy krawędź skierowana $:\Leftrightarrow$ w dokumencie D x poprzedza y o co najwyżej k słów. Może zawierać krawędzie wielokrotne, a także każdy wierzchołek zawiera krawędź do siebie samego.

Budowa grafu

1. zastosowanie stoplisty
2. utworzenie wierzchołka dla każdego słowa występującego w tekście
3. utworzenie okna przesuwnego zawierającego $k + 1$ słów
   1. dodanie krawędzi od pierwszego słowa w oknie, do każdego (łącznie z nim samym)
   2. przesunięcie okna o 1 i powtórzenie operacji

Twierdzenie

Jeśli tekst *B* zawiera się w tekście *A*, to graf utworzony z tekstu *B* jest podgrafem grafu utworzonego z tekstu *A*.

Cechy modelu grafowego

Zalety:

- zachowuje informacje o następstwie słów
- konwertowalny na VSM, więc wszystkie algorytmy są nadal stosowalne
- czytelny dla człowieka

Wady:

- większy koszt pamięciowy i obliczeniowy niż VSM

Konwersja na VSM (nie „bag of words”)

$w_1, w_2, ..., w_n$ – słowa

Bag of words: $x = [f(w_1), f(w_2), ..., f(w_n)]$
$f(w)$ – liczba wystąpień w lub jej funkcja

VSM z reprezentacji grafowej: $x =$
$[F(w_1, w_1), F(w_1, w_2), ..., F(w_1, w_n), F(w_2, w_1), ..., F(w_n, w_n)]$
$F(w_a, w_b)$ – liczność krawędzi od w_a do w_b lub jej funkcja

Zastosowania

- klasteryzacja
- klasyfikacja (Bayesa, kNN, centroid, regułowa)
- indeksowanie i wyszukiwanie
- wykrywanie plagiatów

Zadania

1. napisać program wyszukujący w korpusie PAP notatki podobne do wybranej:
   - korzystając z modelu tf-idf (1 pkt.)
   - korzystając z modelu grafowego (1 pkt.)
2. porównać wyniki powyższych algorytmów (dla modelu grafowego przetestować różne rzędy) (1 pkt.)

Materiały: `http://apohllo.pl/text/lab6.tar.gz`

Schemat oceny algorytmu

- wybrać notatkę wzorcową
- uruchomić wszystkie (minimum 4) algorytmy, celem wyszukania notatek podobnych
- wszystkie wyniki tych algorytmów połączyć w jeden zbiór wynikowy
- ręcznie przejrzeć zbiór wynikowy i usunąć z niego notatki niepodobne do wzorcowej
- powstały zbiór jest wynikiem wzorcowym dla klasyfikacji binarnej
- ocenić wszystkie zastosowane algorytmu na podstawie Precision@10, Recall@10 i F1@10