Ostatnio poznaliśmy jak tworzy się słownik i jak przedstawi słowa w postaci zrozumiałej dla modeli

Mamy dwa dokumenty

Dokument 1:

Data mining is the process of discovering patterns in large data sets

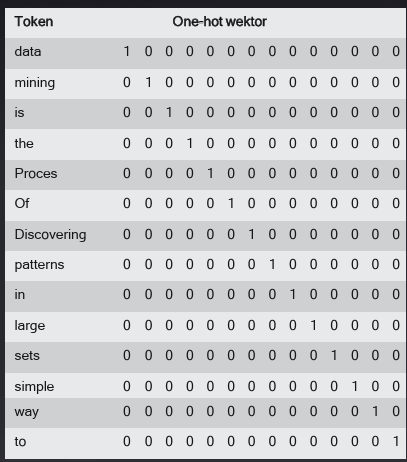
Dokument 2:

The simple way to process large data sets is the data minig

Tworząc słownik nic nie odrzucając dochodzimy do takiego słownika

{data, minig, is, the, process, of, discovering, patterns, in, large, sets, way, to}

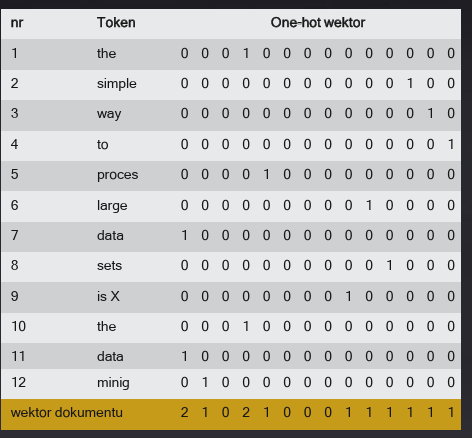
Wówczas możemy zakodować słowa metodą one-hot-encoding



Utworzyła nam się macierz i teraz aby utworzyć na jej bazie wektor charakterystyczne dla całego dokumentu wystarczy zsumować wszystkich wystąpień danego słowa



Utwórz wektor dla drugiego dokumentu



Dwa zdania zostały zastąpione wektorami:

Wektor dokumentu 1: [2 1 1 1 1 1 1 1 1 1 1 0 0 0]

Wektor dokumentu 2: [2 1 0 2 1 0 0 0 1 1 1 1 1 1]

Te wektory w rzeczywistości są dość duże więc lepiej zachować te wartości w słowniku czyli unikalne wystąpienie to będzie klucz a wartością ilość wystąpień. Oczywiście jest to tymczasowe rozwiązanie bo do modeli trzeba przekonwertować wektor BoW na odpowiedni.

BoW1:

{data:2, simple:1,…minig:1}

BoW2:

{the:2, simple:1,…,minigi:1}

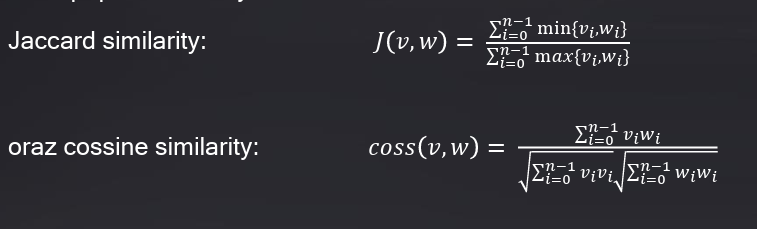
Jeżeli wystąpi nowe słowo to je pomijamy. Po całym procesie przychodzi dokument Data mining is a fun słowo a i fun jest pomijane bo nie wystepuje w słowniku. W taki sposób otrzymujemy algorytm Bag-of-Words opis w pdf.

Ma to podejście wady- konieczność przechowywania dużych wektorów, nie uwzględnia kolejności słów i współwystąpienia.

Można zamiast sumować użyć operacji OR czyli czy występuje słowo czy nie.

Wektorów używa się np. przy sentimential analysis czyli prosta klasyfikacja czy dany dokument jest spamem czy pasuje do wytycznych itp.

Aby porównać wektory stosuje się miary podobieństw dwie popularne to Jaccard similarity oraz cossine similarity.



Jak porównywać

BoW 1: [2 1 1 1 1 1 1 1 1 1 1 0 0 0]

BoW 2: [2 1 0 2 1 0 0 0 1 1 1 1 1 1]

Jaccard simiarity to: 8/16=0.5

Cossine similarity to: 11/sqrt14sqrt16 ~0.735

Dokumenty są tym bardziej podobne jak miara jest bliższa 1 ale rozbieżność w wynikach pokazuje że miary są wrażliwe na rozkład danych w szczególności Jaccard

Inne możliwości

TF-IDF-> to jest zlepek dwóch terminów Inverse Document Frequency to komponent który ma karać słowa które występują w każdym dokumencie czyli jeżeli występuje w każdym dokumencie to jest mało informacyjne dla zadania i jego waga powinna zostać zmniejszona.

Ti to ite słowo ze słownik a jego wartość to logarytm z liczby dokumentów w korpusie/ liczbę elementów dokumentów w którym ite słowo występuje.

Rozważmy dwa przypadki słowo występuje w każdym dokumencie wówczas mianownik jest równy ilości dokumentów w korpusie czyli mamy logarytm z 1 czyli IDF=0 a jeżeli słowo występuje tylko raz to mamy logarytm z ilości dokumentów czyli dużą liczbę. Jeżeli słowo jest powszechnie używane to ta wartość będzie mała i odwrotnie

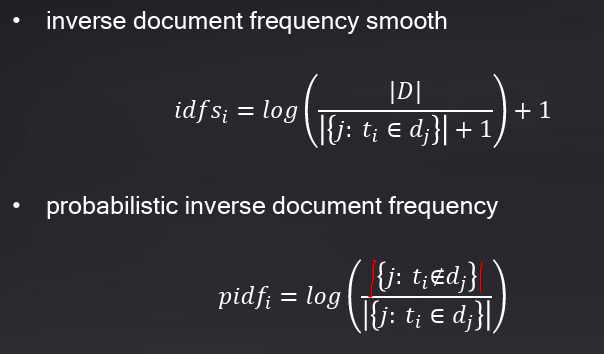
Ten czynnik jest obliczany dla danego słowa względem korpusu nie uwzględnia się dokumentu

TF- Term Frequency to liczba wystąpień w dokumencie

A TF-IDF to iloczyn tych dwóch -> TF-IDF oraz TF jest uzależniony od słowa i od dokumentu a IDF od słowa.

Inne wartości dla Term Frequency jest wartość binarna, słowa przez liczbę słów w dokumencie albo log z 1 +tf

Alternatywy da IDF miary wygładzającą logarytmiczną zależność liniową



W drugim wzorze rozważmy trzy przypadki-> element należy tylko do jednego dokumentu czyli log z D-1 czyli liczba stosunkowo duża, element należy do wszystkich dokumentów czyli wyjdzie log 1/d-1 czyli wartość bardzo mała. Co by się stało jeśli element należy do połowy dokumentów czyli log 1.

Zwiększamy słownik

n-gramy to są ciągi n kolejnych słów z dokumentu unigramy bigramy trigramy

mamy zdanie Data mining is a fun

data mining

mining is

is a

a fun

Robimy to po to aby wyszukać pary które występują ze sobą często jest to operacja kosztowna, można wykorzystać do tego metody często powtarzających się wzorców.

Zwróćmy uwagę na bigram data mining to co innego niż słowo data i mining.

Jest jeszcze że słowa nie muszą być jedno po drugim k-skip n-gram dla słów nie występujące po sobie

3-skip 2-gram

