CS5560 Knowledge Discovery and Management

Problem Set 3
June 19 (T), 2017

Name: Yalamanchili Sowmya
Class ID: 30

Information Retrieval (Text Mining) with TF-IDF
Consider the following three short documents

Doc #1:
The researchers will focus on computational phenotyping and will produce disease prediction models from machine learning and statistical tools.

Doc #2:
The researchers will develop tools that use Bayesian statistical information to generate causal models from large and complex phenotyping datasets.

Doc #3:
The researchers will build a computational information engine that uses machine learning to combine gene function and gene interaction information from disparate genomic data sources.

a) First remove stop words and punctuation; detect manually multi-word terms (using N-Gram or POS Tagging/Chunking); parse manually the documents and select the terms from the given 3 documents and created the dictionary (list of terms).

b) Create the document vectors by computing TF-IDF weights. Show how to compute the TF-IDF weights for terms. For each form of weighting list the document vectors in the following format:

	Term1	Term2	Term3	Term4	Term5	Term6	Term7	Term8 ...
DOC1	0	3	1	0	0	2	1	0 ....
DOC2	5	0	0	0	3	0	0	2 ....
DOC3	3	0	4	3	4	0	0	5 ....

Name: Yalamanchili Soumya
ClassID: 30

(a) Stopwords:

Stopwords are words which do not contain important significan to be used in search queries. Usually these words are filtered ou from search queries because they return vast amount of unnecessa information.

Removal of stopwords and punctuation:

Doc1: After removing stopwords and punctuation the output will be as follows.

O/p: researchers focus computational phenotyping produce disease prediction models machine learning statistical tools

Doc2

researchers develop tools Bayesian statistical information generate causalmodels large complex phenotyping datasets

Doc3

researchers build computational information engine uses machir learning combine gene function gene interaction information disparate genomic data sources.

N-Gram

N-gram is a contiguous sequence of n items from a given sequ of text or speech. The items can be phonemes, syllables, lett words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. When the it are words, n-grams may also be called shingles.

An n-gram of size 1 is referred to as a "unigram", size-2 is a "bigram", size-3 is a trigram. Larger sizes are sometimes referred to by the value of n in modern language, eg. "four-gram", "five-gram", and so on.

Doc1 o/p :- After applying N-gram the result of doc1 is as follows:-

The researchers will
researchers will focus
will focus on
focus on computational
on computational phenotyping
computational phenotyping and
phenotyping and will
and will produce
will produce disease
produce disease prediction
disease prediction models
prediction models from
models from machine
from machine learning
machine learning and
learning and statistical
and statistical tools

let N=3

Doc2 o/p :- The researchers The output for N=3 after remov

stopwords for doc 1 is

researchers focus computational
focus computational phenotyping
computational phenotyping produce
phenotyping produce disease
produce disease prediction
disease prediction models
Prediction models machine

Take N=3.

Name: yaramanchili Sowmya

class ID: 30

models machine learning
machine learning statistical
learning statistical tools

Doc 2

researchers develop tools
develop tools Bayesian
tools Bayesian statistical
Bayesian statistical information
Statistical information generate
information generate casual
generate casual models
casual models large
models large complex
large complex phenotyping
complex phenotyping datasets.

N=3

Doc 3

researchers build computational
build computational information
computational information engine
information engine uses
engine uses machine
uses machine learning
machine learning combine
learning combine gene
combine gene function
gene function gene
function gene interaction
gene interaction information
interaction information disparate
information disparate genomic
disparate genomic data

genomic data sources

Here also N=3

list of items from the 3 documents are as shown below

	D_1	D_2	D_3	count in 3 docs
researchers,	1	1	1	3
focus	1	0	0	1
computational	1	0	1	2
phenotyping	1	1	0	2
produce	1	0	0	1
disease	1	0	0	1
prediction	1	0	0	1
models	1	1	0	2
machine	1	0	1	2
learning	1	0	1	2
statistical	1	1	0	2
tools	1	1	0	2
develop	0	1	0	1
Bayesian	0	1	0	1
nformation	0	1	1	2
generate	0	1	0	1
casual	0	1	0	1
large	0	1	0	1
complex	0	1	0	1
datasets	0	1	0	1
build	0	0	1	1
sources	0	0	1	1
engine	0	0	1	1
data	0	0	1	1
Uses	0	0	1	1
ombine	0	0	1	1
gene	0	0	1	1
unction	0	0	1	1
nteraction	0	0	1	1
disparate	0	0	1	1
genomic	0	0	1	1

Name: Yalamanchili Soumya

Rollno: 30

(b) TF: It is the Term frequency which measures how frequently a term occurs in a document since every document is differen in length, It is possible that a term would appear much more times in a long document.

$$TF(t) = \frac{\text{No of times term t appears in a document}}{\text{Total no of terms in the document.}}$$

IDF: which measures how important a term is, while computing TF, c terms are considered equally important.

$$IDF(t) = \log_e(\text{Total no of documents} / \text{No of documents with term t in it}).$$

TF-IDF: Tf-Idf stands for term frequency- inverse document frequen and the tf-idf weight is a weight often used in informatio retrieval and text mining.

	D1	D2	D3
researchers	1	1	1
focus	1	0	0
computational	1	0	1
phenotyping	1	1	0
produce	1	0	0
disease	1	0	0
prediction	1	0	0
models	1	1	0
machine	1	0	1
learning	1	0	1
statistical	1	1	0
tools	1	1	0
develop	0	1	0

	D1	D2	D3
Bayesian	0	1	0
information	0	1	2
generate	0	1	0
casual	0	1	0
large	0	1	0
complex	0	1	0
datasets	0	1	0
build	0	0	1
sources	0	0	1
engine	0	0	1
data	0	0	1
uses	0	0	1
combine	0	0	1
gene	0	0	2
function	0	0	1
interaction	0	0	1
disparate	0	0	1
genomic	0	0	1

This matrix shows how many times a term occurs in a document.

Now we are calculating TF-IDF values for each term in D

For researchers

$TF = \frac{1}{12} = 0.083$ $\quad IDF = \log_e\left(\frac{3}{2}\right) = 0.176$ $\quad TF\text{-}IDF = \frac{1}{12} \times \log_e\left(\frac{3}{2}\right) = 0.0083 \times 0.176 = 0.0146$

for focus

$TF = \frac{1}{12} \quad IDF = \log_e\left(\frac{3}{1}\right) = 0.477 \quad TF\text{-}IDF = \frac{1}{12} \times 0.477 = 0.039$

For computational

$TF = \frac{1}{12} \quad IDF = \log_e\left(\frac{3}{2}\right) = 0.176 \quad TF\text{-}IDF = 0.0146$

For Phenotyping

$TF = \frac{1}{12} \quad IDF = \log_e\left(\frac{3}{2}\right) = 0.176 \quad TF\text{-}IDF = 0.0146$

Name :- Yalamanchili Sowmya
Roll no :- 30

For produce
$TF = \frac{1}{12}$ $IDF = \log_e\left(\frac{3}{1}\right) = 3$ $TF\text{-}IDF = \frac{1}{12} \times 0.477 = 0.039$

For disease
$TF = \frac{1}{12}$ $IDF = \log_e\left(\frac{3}{1}\right) = 3$ $TF\text{-}IDF = \frac{1}{12} \times 0.477 = 0.039$

For prediction
$TF = \frac{1}{12}$ $IDF = \log_e\left(\frac{3}{1}\right) = 0.477$ $TF\text{-}IDF = \frac{1}{12} \times 0.477 = 0.039$

For models
$TF = \frac{1}{12}$ $IDF = \log_e\left(\frac{3}{2}\right) = 0.176$ $TF\text{-}IDF = 0.0146$

For machine
$TF = \frac{1}{12}$ $IDF = \log\left(\frac{3}{2}\right) = 0.176$ $TF\text{-}IDF = 0.0146$

For learning
$TF = \frac{1}{12}$ $IDF = \log\left(\frac{3}{2}\right) = 0.176$ $TF\text{-}IDF = 0.0146$

For statistical
$TF = \frac{1}{12}$ $IDF = \log\left(\frac{3}{2}\right) = 0.176$ $TF\text{-}IDF = 0.0146$

For tools
$TF = \frac{1}{12}$ $IDF = \log\left(\frac{3}{2}\right) = 0.176$ $TF\text{-}IDF = 0.0146$

For remaining all terms which are not present in D_1, $TF = 0$ then
$TF\text{-}IDF = 0$

Now we are calculating TF-IDF values for each term in D_2

For develop
$TF = \frac{1}{13}$ $IDF = \log\left(\frac{3}{1}\right) = 0.477$ $TF\text{-}IDF = 0.036$

For Bayesian
$TF = \frac{1}{13}$ $IDF = \log\left(\frac{3}{1}\right) = 0.477$ $TF\text{-}IDF = 0.036$

For information
$TF = \frac{1}{13}$ $IDF = \log\left(\frac{3}{2}\right) = 0.176$ $TF\text{-}IDF = 0.0135$

for researchers

$TF = \frac{1}{13}$ $IDF = \log\left(\frac{3}{3}\right)$ $TF * IDF = 0$

For tools

$TF = \frac{1}{13}$ $IDF = \log\left(\frac{3}{2}\right)$ $TF * IDF = 0.0135$

For statistical

$TF = \frac{1}{13}$ $IDF = \log\left(\frac{3}{2}\right)$ $TF * IDF = 0.0135$

For generate, large, datasets, casual, complex

$TF = \frac{1}{13}$, $IDF = \log\left(\frac{3}{1}\right)$ $TF.IDF = 0.1036$

For phenotyping, models

$TF = \frac{1}{13}$, $IDF = \log\left(\frac{3}{2}\right)$ $TF * IDF = 0.0135$

Now we are calculating TF-IDF for each term in a D3

for researchers

$TF = \frac{1}{18}$ $IDF = \log\left(\frac{3}{3}\right)$ $TF * IDF = 0$

For build

$TF = \frac{1}{18}$ $IDF = \log\left(\frac{3}{1}\right)$ $TF * IDF = 0.026$

For computational

$TF = \frac{1}{18}$ $IDF = \log\left(\frac{3}{2}\right)$ $TF * IDF = 0.026\ 0.1978$

For information

$TF = \frac{2}{18}$ $IDF = \log\left(\frac{3}{2}\right)$ $TF.IDF = 0.1095$

For engine, uses, machine, combine, function, interaction, data, sources, genomic, disparate

$TF = \frac{1}{18}$ $IDF = \log\left(\frac{3}{1}\right)$ $TF * IDF = 0.026$

For gene

$TF = \frac{2}{18}$ $IDF = \log\left(\frac{3}{1}\right)$ $TF\ IDF = 0.0529$

For learning

$TF = \frac{1}{18}$ $IDF = \log\left(\frac{3}{2}\right)$ $TF.IDF = 0.1978$