Natural-Language Processing

STT Cipasung

12/7/2022

Zulkaida Akbar

Apa yang sudah kita pelajari:

- Dasar dan pengenalan NLP: Menghitung dan plotting kata dari teks
- Sentiment analysis: Amazon review, twitter, dst..
- Topic modelling: Clustering topik dari Amazon review
- ChatBot AI: Simple ChatBot untuk pelayanan konsumen
- **Search Engine**

Search Engine:

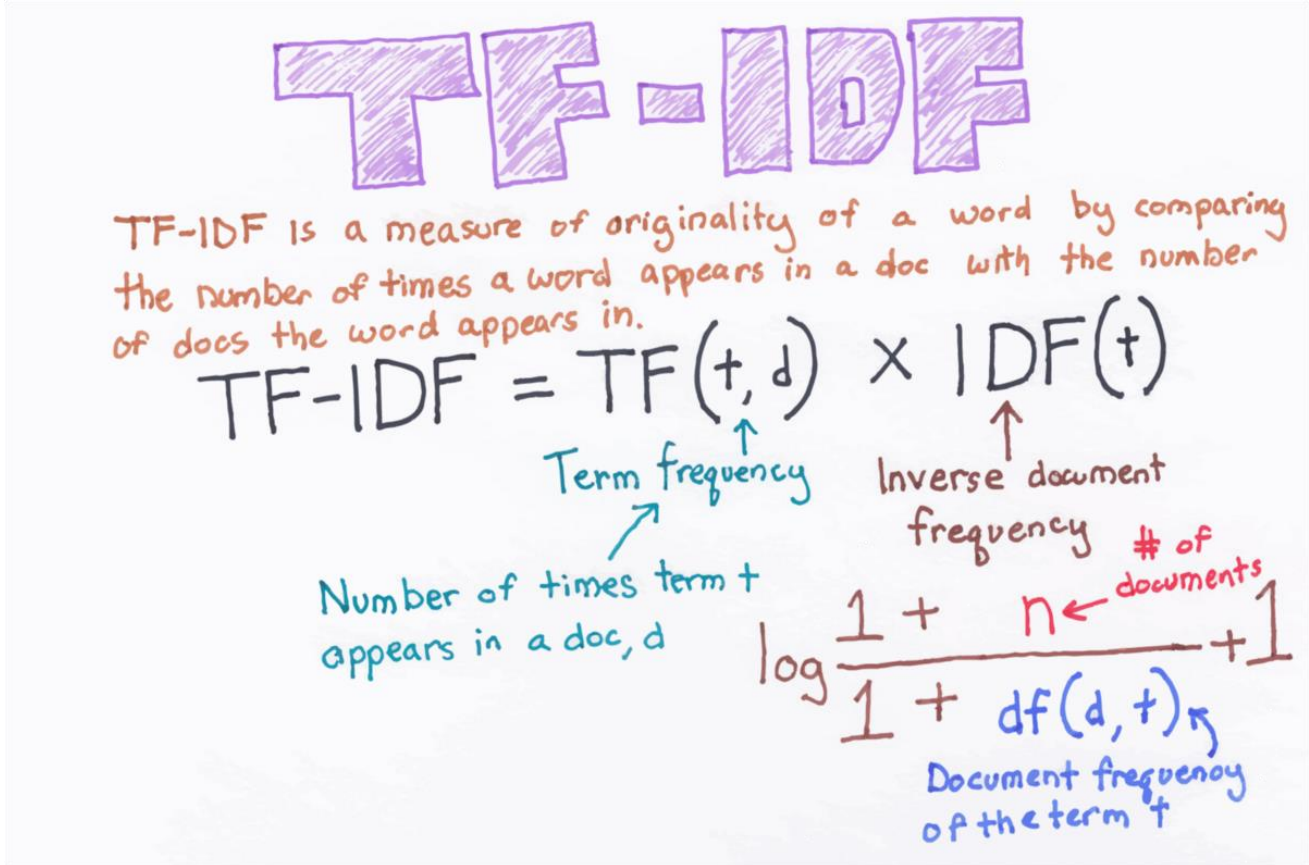
- Simple search engine untuk satu artikel wikipedia

```
Spar: My name is Spar. I will answer your queries about Chatbots. If you want to exit, type Bye!
hi
Spar: hello
hi
Spar: hi
what is chatbot
Spar: a virtual assistant chatbot

the 1966 eliza chatbot
a chatbot or chatterbot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct
contact with a live human agent.
what is chatbot application
```

- Search-Engine untuk pencarian dokumen terkait dengan “query” tertentu

```
QUERY: animals of zimbabwe

./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/Fauna_of_Zimbabwe 0.0962172268166385
./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/Cold_War_films 0.027295703295014084
./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/Brazilian_telenovelas 0.0
./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/Freeware_games 0.0
./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/International_law 0.0
./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/Landscape_painters 0.0
./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/Members_of_the_Chinese_Academy_of_Sciences 0.0
./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/Musicals_based_on_novels 0.0
./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/Particle_physics 0.0
./drive/MyDrive/Colab-Notebooks/Search-Engine/Categories/21st-century_women_mathematicians 0.0
```

- **Mini Google: Search engine untuk mencari dokumen relevan berdasarkan Query atau keyword dengan menggunakan metode TF-IDF dan Cosine simiarity**



$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

```
Document 1: The game of life is a game of everlasting learning
Document 2: The unexamined life is not worth living
Document 3: Never stop learning
```

Let us imagine that you are doing a search on these documents with the following query: **life learning**

The query is a free text query. It means a query in which the terms of the query are typed freeform into the search interface, without any connecting search operators.

Let us go over each step in detail to see how it all works.

Step 1: Term Frequency (TF)

Term Frequency also known as TF measures the number of times a term (word) occurs in a document. Given below are the terms and their frequency on each of the document.

TF for Document 1

Document1	the	game	of	life	is	a	everlasting	learning
Term Frequency	1	2	2	1	1	1	1	1

TF for Document 2

Document2	the	unexamined	life	is	not	worth	living
Term Frequency	1	1	1	1	1	1	1

In reality each document will be of different size. On a large document the frequency of the terms will be much higher than the smaller ones. Hence we need to **normalize** the document based on its size. A simple trick is to divide the term frequency by the total number of terms. For example in Document 1 the term **game** occurs **two** times. The total number of terms in the document is **10**. Hence the normalized term frequency is **2 / 10 = 0.2**. Given below are the normalized term frequency for all the documents.

Normalized TF for Document 1

Document1	the	game	of	life	is	a	everlasting	learning
Normalized TF	0.1	0.2	0.2	0.1	0.1	0.1	0.1	0.1

Normalized TF for Document 2

Document2	the	unexamined	life	is	not	worth	living
Normalized TF	0.142857	0.142857	0.142857	0.142857	0.142857	0.142857	0.142857

Step 2: Inverse Document Frequency (IDF)

The main purpose of doing a search is to find out **relevant documents** matching the query. In the first step all terms are considered equally important. In fact certain terms that occur too frequently have little power in determining the relevance. We need a way to **weigh down** the effects of too frequently occurring terms. Also the terms that occur less in the document can be more relevant. We need a way to **weigh up** the effects of less frequently occurring terms. Logarithms helps us to solve this problem.

Let us compute IDF for the term **game**

```
IDF(game) = 1 + log_e(Total Number Of Documents / Number Of Documents with term game in it)

There are 3 documents in all = Document1, Document2, Document3
The term game appears in Document1

IDF(game) = 1 + log_e(3 / 1)
          = 1 + 1.098726209
          = 2.098726209
```

Given below is the IDF for terms occurring in all the documents. Since the terms: **the, life, is, learning** occurs in 2 out of 3 documents they have a lower score compared to the other terms that appear in only one document.

Terms	IDF
the	1.405507153
game	2.098726209
of	2.098726209
life	1.405507153
is	1.405507153
a	2.098726209

Step 3: TF * IDF

Remember we are trying to find out relevant documents for the query: **life learning**

For each term in the query multiply its normalized term frequency with its IDF on each document. In Document1 for the term **life** the normalized term frequency is 0.1 and its IDF is 1.405507153. Multiplying them together we get **0.140550715** (0.1 * 1.405507153). Given below is TF * IDF calculations for **life and learning** in all the documents.

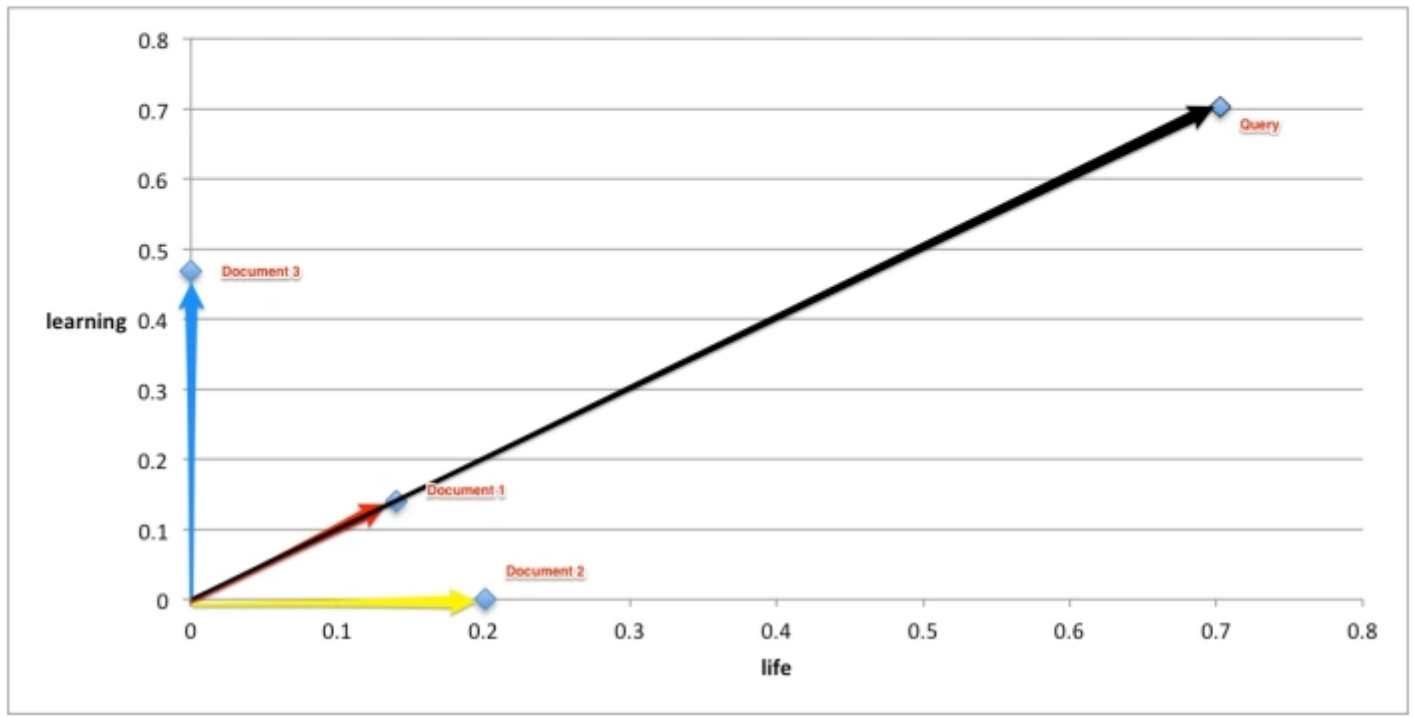
	Document1	Document2	Document3
life	0.140550715	0.200786736	0
learning	0.140550715	0	0.468502384

The query entered by the user can also be represented as a vector. We will calculate the TF*IDF for the query

	TF	IDF	TF*IDF
life	0.5	1.405507153	0.702753576
learning	0.5	1.405507153	0.702753576

	Document1	Document2	Document3
Cosine Similarity	1	0.707106781	0.707106781

I plotted vector values for the query and documents in 2-dimensional space of life and learning. Document1 has the highest score of 1. This is not surprising as it has both the terms life and learning.



https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/