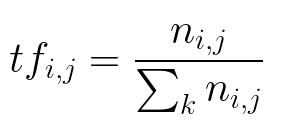
Nama : Sofia Dewi Rahmadiyanti (15523207)

Siwi Cahyaningtyas (15523223)

**TF – IDF**

TF-IDF (Term Frequency – Inverse Data Frequency).

TF atau *Term Frequency* digunakan untuk memberikan frekuensi setiap kata pada dolumen yang ada di dalam corpus. Dalam perhitungannya banyaknya kata yang ingin diketahui frekuensi dibandingkan dengan jumlah keseluruhan kata yang ada di dalam dokumen tersebut. Pada setiap dokumen memiliki tf nya masing-masing. Semakin besar jumlah kemunculan suatu term (TF tinggi) dalam dokumen, semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar. Berikut rumus untuk perhitungan TF:



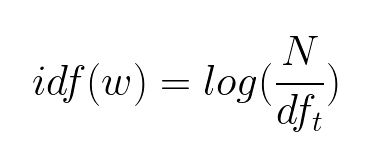
Adapun jenis TF yang lain seperti:

1. TF biner (binary TF): mencari tahu apakah suatu kata ata term terdapat didalam dokumen atau tidak, akan bernilai 1 kalau ada, kalau tidak maka bernilai 0
2. TF murni: tf yang diberikan sesuai dengan jumlah kemunculan term atau kata, jika muncul 2 maka akan bernilai 5 kata tersebut
3. TF logaritmik: untuk menghindari suatu dokumen yang memiliki sedikit term namun memiliki frekuensi tinggi dalam query
4. TF normalisasi: menggunakan perbandingan antara frekuensi sebuah term dengan nilai maksimum dari keseluruhan term pada dokumen. Normalisasi ini digunakan untuk menghindari adanya bias. Berikut perhitungannya:

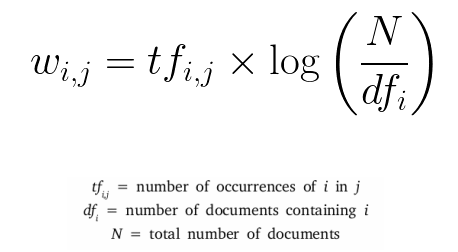


Dimana a adalah nilai antara 0 dan 1 dan umumnya ditetapkan dengan 0,4, meskipun beberapa biasa diawali dengan menggunakan nilai 0,5.

IDF atau Inverse Data Frequency merupakan suatu perhitungan dari bagaimana term didistribusikan secara luas pada koleksi dokumen yang bersangkutan. IDF menunjukkan hubungan ketersediaan sebuah term dalam seluruh dokumen. Semakin sedikit jumlah dokumen yang mengandung term yang dimaksud, maka nilai IDF semakin besar. Berikut perhitungannya:



Kombinasi dari tf dan idf akan menghasilkan sebuah nilai untuk sebuah kata dalam sebuah dokumen di dalam corpus. Berikut perhitungannya:



Contoh perhitungan dari penjelasan diatas (kalimat berikut berada diberbeda dokumen):

Kalimat 1: Adik pergi ke sekolah naik sepeda.

Kalimat 2: Kakak pergi ke sekolah naik motor.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kata** | **TF** | | **IDF** | **TF\*IDF** | |
| **Kalimat 1** | **Kalimat 2** | **Kalimat 1** | **Kalimat 2** |
| Adik | 1/6 | 0 | Log(2/1) = 0.3 | 0.05 | 0 |
| Kakak | 0 | 1/6 | Log(2/1) = 0.3 | 0 | 0.05 |
| Pergi | 1/6 | 1/6 | Log(2/2) = 0 | 0 | 0 |
| Ke | 1/6 | 1/6 | Log(2/2) = 0 | 0 | 0 |
| Sekolah | 1/6 | 1/6 | Log(2/2) = 0 | 0 | 0 |
| Naik | 1/6 | 1/6 | Log(2/2) = 0 | 0 | 0 |
| Sepeda | 1/6 | 0 | Log(2/1) = 0.3 | 0.05 | 0 |
| Motor | 0 | 1/6 | Log(2/1) = 0.3 | 0 | 0.05 |

Dari kesimpulan perhitungan diatas menunjukan bahwa kata yang bernilai 0 tidak signifikan. Dan kata lainnya yang bernilai tidak 0 lebih signifikan. Semakin besar nilai perhitungan bobot yang diperoleh maka semakin tinggi tingkat similaritas dokumen terhadap query.

Tujuan dan manfaat penggunaan TF-IDF di dalam pengolahan bahasa alami:

* Untuk menilai penting atau tidaknya sebuah term yang muncul dalam beberapa dokumen.
* Untuk memodelkan setiap dokumen ke dalam ruang vektor, mengabaikan urutan yang tepat dari kata-kata dalam dokumen sambil menyimpan informasi tentang kemunculan setiap kata.
* Akan mengurangi istilah yang sering terjadi sementara meningkatkan istilah yang jarang terjadi. misalnya, istilah yang terjadi 10 kali lebih banyak dari yang lain tidak 10 kali lebih penting daripada itu.

Contoh implementasi penggunaan TF-IDF di dalam pengolahan bahasa alami :

* Pengklasifikasian berita online sesuai kategorinya.
* Pengklasifikasi mood lirik lagu.

Referensi:

<https://informatikalogi.com/term-weighting-tf-idf/>

<https://medium.freecodecamp.org/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3>

https://nlp.stanford.edu/IR-book/html/htmledition/maximum-tf-normalization-1.html