TF-IDF merupakan suatu algoritma yang digunakan untuk mencari bobot dari suatu term pada suatu dokumen dengan memperhatikan kemunculannya pada dokumen-dokumen lainnya. Sehingga nantinya, suatu kata terkait atau term, dapat dicari tingkat relevansinya pada dokumen tertentu terhadap dokumen-dokumen lainnya yang pada pada kumpulan dokumen.

Cara menghitung bobot dengan TF-IDF

Dalam menghitung nilai TF-IDF, pertama-tama perlu dicari terlebih dahulu nilai TF (Term Frequency, tingkat kemunculan suatu kata) pada dokumen tertentu. Hal ini bisa dilakukan dengan beberapa cara, pertama adalah dengan menggunakan TF biner, di mana tingkat kemunculan hanya dilihat apakah kata tersebut terkandung dalam dokumen, sehingga hanya menghasilkan nilai biner 1 (ada) dan 0 (tidak ada). Cara kedua adalah dengan TF murni, yaitu dengan menghitung jumlah total kemunculan dari kata pada dokumen. Ketiga adalah TF logaritmik, yang mana digunakan jika suatu dokumen hanya memiliki total kata yang sedikit tetapi frekuensinya cukup banyak. Sementara cara terakhir adalah dengan TF normalisasi, yaitu dengan membandingkan jumlah kemunculan suatu kata dengan total keseluruhan dari kata yang tersedia.

Selanjutnya adalah menghitung nilai IDF (Inverse Document Frequency), di mana di sini akan dicari nilai ketersebaran suatu kata pada sekumpulan dokumen yang ada. Untuk menghitungnya digunakan rumus berikut,

$$IDFj = log(D/dfj)$$

D = total dokumen
dfj = jumlah kemunculan kata pada kumpulan dokumen

Tahap terakhir adalah dengan menghitung bobot dari kata yang dicari dengan mengalikan nilai TF pada dokumen dengan nilai IDFnya.

$$wij = TFij \times IDFj$$

Untuk menghindari nilai 0 apabila kata yang dicari berada pada setiap dokumen yang terkumpul (log 1 = 0), maka dapat digunakan rumus berikut untuk mengatasinya,

$$wij = TFij \times (IDFj + 1)$$

Tujuan dan Manfaat penggunaan TF-IDF dalam PBA

Pada dasarnya, tujuan utama TF-IDF adalah mencari tingkat relevansi antara suatu dokumen dengan dokumen lainnya. Di antara manfaat dari mengetahui tingkat relevansi tersebut adalah:

- Mencari dokumen yang paling relevan dengan kata kunci yang dimasukkan user ke dalam mesin pencari dokumen
- Mengetahui dokumen lain yang isinya mirip atau berkaitan dengan sebuah dokumen, sehingga memudahkan dalam mencari referensi lain yang bertema sama

referensi:

https://informatikalogi.com/term-weighting-tf-idf/
https://www.elephate.com/blog/what-is-tf-idf/
http://www.tfidf.com/

Contoh Implementasi Penggunaan TF-IDF

Contoh implementasi penggunaan TF-IDF dapat kita temukan pada Sistem Monitoring Diskusi Online. Dalam sebuah forum diskusi online, admin tentu tidak bisa mengontrol apa isi komentar yang dituliskan oleh user. Dengan kondisi ini, sangat mungkin muncul banyak komentar *spam* atau komentar lain yang tidak berhubungan dengan tema. Oleh karena itu, TF-IDF dibutuhkan untuk menentukan bobot kesesuaian komentar dan tema diskusi. Jika bobot sebuah komentar adalah 0, maka komentar tersebut kemungkinan besar tidak layak dan direkomendasikan untuk dihapus dari forum.

Sebagai contoh, sebuah forum memiliki tema “Penerapan Sistem Pakar dalam Dunia Medis”. Dari forum itu kemudian muncul 20 komentar. Langkah pertama adalah melakukan Analisa pemecahan kalimat/komentar. Setelah semua kalimat/komentar terpecah, kemudian dilakukan ***case folding*** untuk mengubah teks menjadi huruf kecil, menghilangkan angka serta tanda baca dan simbol lainnya. Setelah itu, dilakukan *filtering* dengan menghilangkan kata tidak penting menggunakan algoritma ***stopword***. Kemudian hasil *filtering* dipotong dengan menggunakan ***tokenization***. Setelah itu dilakukan ***stemming*** untuk merubah seluruh kata menjadi bentuk kata dasarnya. Dengan demikian, data siap dihitung bobotnya sesuai dengan tema “Penerapan Sistem Pakar dalam Dunia Medis” tersebut. Sehingga data dengan bobot **0** dapat direkomendasikan untuk dihapus dari forum, sehingga komentar dalam forum terjaga dari *spam* maupun diskusi tidak sehat.

Rujukan Sistem : http://ejournal.uin-suska.ac.id/index.php/sitekin/article/viewFile/1399/1408