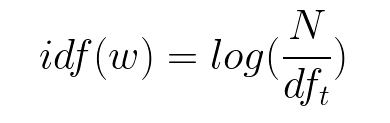
Model 2 =>TF-IDF APPROACH

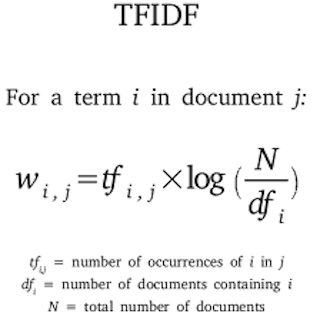
**REASON:**

1. **tf**-**idf** is a weighting scheme that assigns each term in a document a weight based on its term frequency (**tf**) and inverse document frequency (**idf**). The terms with higher weight scores are considered to be more important
2. use of word.net which helps in determining synonyms with reference to context which helps in determining meaning of content
3. two documents can be called similar If words occurrence are same in both the content
4. cosine formulation used to determine similarity function because it lies in 0 and 1 range with lemmas
5. Method used=clustering,unsupervised learning
6. Fast in process

explanation:

* TF(w) = (Number of times term w appears in a document) / (Total number of terms in the document)
* IDF(w) = log\_e(Total number of documents / Number of documents with term w in it)





Cosine Similarity (d1, d2) = Dot product(d1, d2) **/** ||d1|| \* ||d2||

Dot product (d1,d2) = d1[0] \* d2[0] + d1[1] \* d2[1] \* … \* d1[n] \* d2[n]

||d1|| = square root(d1[0]2 + d1[1]2 + ... + d1[n]2)

||d2|| = square root(d2[0]2 + d2[1]2 + ... + d2[n]2)

Model 2 =>Doc2Vec APPROACH

REASON

1. Easy analysis of documents
2. Less processing and code
3. Automated model available in nltk
4. More accurate similarity

and Explanation:

1. **Doc2vec** is an unsupervised computer algorithm to generate vectors for sentence/paragraphs/documents. The algorithm is an adaptation of word2vec which can generate vectors for words.
2. The vectors generated by doc2vec used for tasks like **finding similarity** between sentences
3. the paragraph vector was added to represent the missing information from the current context and to act as a memory of the topic of the paragraph
4. Uses cluster method but with some labeling to distinguish two texts and make comparison
5. Word embedding method is used to make data vectors

