Post 1

ال Bag of words المختلفة من Binary vectorization ل counts ل frequency فى الاخر بتعبر عن الكلمة جت فى ال document ده ولا لا زى ال binary ، والتانيين بيدوا اهمية للكلمات عن طريق ال counts وده عيوبة بتظهر سواء فى ال ranges المختلفة الى الكلمات ممكن تاخدها مع الكلمات الى ممكن تظهر كتير جدا فى document كبير بينما بالنسبة لكلمات تانية عددها قليل و ال ranges المختلفة بالنسبة للموديل بيكون فيها بعض المشاكل وبنحتاج زى range معين للارقام من 0 ل 1 مثلا وغير كده ال rare words رغم اهميتها بالنسبة لل document هنلاقيها ملهاش اى تاثير بالنسبة لكلمات تانية رغم عد اهميتها ولكن عددها كبير ، وده بشكل او باخر تم علاجة عن طريق ال frequency of words بالنسبة لعدد الكلمات فى ال data كلها ولكن برضه لسه معنديش اى معلومات عن اهمية بعض الكلمات من اخرى مش بس بالنسبة لل document الى هى فيه لا كمان بالنسبة لكل ال documents الاخري ، لان كلمات زى and مثلا او to هنلاقيها ذكرها كتير فى معظم ال documents بينما كلمات زى conjunction هنلاقيها ممكن تيجى فى عدد قليل جدا من ال documents واهميتها بالنسبة لل document الى بتيجى فيه بتكون كبيرة ! فهنلاقى انى ال Bag of words كان فى الاخر مجرد انه بيعمل counts سواء بقا one-grams or n-grams ، لكن دلوقت محتاجين نشوف ال score بتاع الكلمات مش بس بالنسبة لل document الى فيه لا بالنسبة كمان لعلاقتها مع ال documents الاخرى ويجى هنا دور ال TF-IDF وده البوست الجاى .

Post 2

ال TF-IDF هو Term Frequency - Inverse Document Frequency الجزء الاول هو انى بشوف كل كلمة جت فى ال document الى انا فيه كام مرة مقسومه على عدد الكلمات فى ال document نفسه ، وده لان ممكن نفس الكلمة تيجى فى two different documents وفى الاول رغم انها جت عدد اقل بكتير الا انها اهم كتير لل document بينما فى التانى رغم انها جت كتير جدا الا انها مش مهمة لذلك بعمل Normlized لل Terms عشان ققدر اشوف الكلمة بالنسبة لطول ال document نفسه .
الجزء التانى وهو الأهم والى بيربط علاقة الكلمة واهميتها بالنسبة لل documents الاخري وهو ال IDF بنروح في الاخر بقا نشوف عندى كام documents فى الداتا كلها وققسمه على عدد ظهور الكلمة ديه فى ال documents لى جت فيها بعد 1 لكل document الكلمة ديه جت فيه ومن ثم بضرب الاتنين في بعض . قارن ال two images عشان تفهم اكتر هنلاقى china لان ظهورها اقل فى ال documents فى ال idf بتاعها اعلى وهنا قدرنا نعالج ال rare words انها هيكون ليها weights اعلى بينما كلمات زى ال stopwords هنلاقي ال weights بتاعتها اقل كتير .

TF(t, d) = Count(t) / Count(d),
- T is the word it self
- d is the number of words in that document
IDF = All docs / Docs contain this term (count one for each document it appears in).

```
In [37]: doc_number = 2

kite_idf = 2 / 2 # as it appeared in the two documents

and_idf = 2 / 2 # as it appeared in the two documents

china_idf = 2 / 1 # as it appeared in the just one document ???????

print(kite_idf)
print(and_idf)
print(china_idf)
```

```
1.0
1.0
2.0
```

TF

Lets look again step by step for tf.

```
i]: intro_kite_tf = intro_tf['kite'] / intro_total # doc 1
history_kite_tf = history_tf['kite'] / history_total # doc 2

intro_and_tf = intro_tf['and'] / intro_total # doc 1
history_and_tf = history_tf['and'] / history_total # doc 2

intro_china_tf = intro_tf['china'] / intro_total # doc 1
history_china_tf = history_tf['china'] / history_total # doc 2

print(intro_kite_tf, history_kite_tf)
print("="*50)
print(intro_and_tf, history_and_tf)
print("="*50)
print(intro_china_tf, history_china_tf)
print("="*50)
```

```
0.0440771349862259 0.020202020202020204
==================================================
0.027548209366391185 0.030303030303030304
==================================================
0.0 0.010101010101010102
==================================================
```

Post 3

كل الخطوات الى فاتت من تحويل ال Text لارقام باختلافاتها فقدر من خلالها دلوقت انى اعمل search على ال documents الى ليها علاقة ببعض ولكن فى الكلمات المشتركة مش معانى الكلمات نفسها لان two document ممكن غير بعض فى الكلمات ولكنهم نفس المعنى ، ولكن حالياً انا وصلت فى الاخر للى انا عايزه وهو انى معايا ارقام بعبر بيها عن ال text لى معايا وقدر اعمل feed للموديل بالارقام ديه ، او انى استخدمها فى انى اعمل sum for two vector او اشوف ال cosin similarity between two vectors كل ده متاح ليا دلوقت انى اعمله واشوف علاقة document بالاخر وفى تشابه اد ايه ، ولكن عشان اعمل math operation انا محتاج ال vectors ديه تكون بتعمل share for common space ويكونوا ليهم نفس ال origin وكمان ليهم نفس ال scale ، وده الى وصلنا ليهم من خلال الاول ال TF وهو اننا عملنا Normalization عشان يكونوا على نفس ال Scale وفى نفس الوقت بحدد عدد الكلمات الى هستخدمها مثلا most common words او كل الكلمات وده ال Vocabulary بتاعتى الى بتخلى كل ال documents ليها نفس ال length ، ال Vector ده من الارقام بيعبر عندى عن اتجاة او طول ال document وكمان الفرق بين 2 documents اد ايه لما ارسم document بالنسبة لاخر وده بيساعدنى انى اعرف ال similarity of exact spelling of two documents لان بيعتمد في الاخر على التشابة فى الكلمات نفسها وال vector ده ال space بتاعة فى الاخر بيكون على حسب ال Vocabulary بتاعى وطبعا زيادة الكلمات بيحتاج computations اعلى لذلك بعض ال libraires بتساعد انك تحدد عدد الكلمات وده بالتالى بيساعد فى تقليل ال computation لانه بيعمل عندك diemntion reduction والى بيخلى ال computations عالية كلما زادت ال vocubalrary ديه هو انك بيبقا عندك d diemntion فهنا ال math لما تشتغل على d-2 غير لما تشتغل على d-1000 ببساطة انك تدور على حاجه على الارض غير انك تدور عليها على جبل .

```
# doc1 = doc1.reshape(-1,1)
cosine_sim(q, doc1)
```

```
Out[43]: 0.5697334438987307
```

```
In [44]: cosine_sim(q, doc2)
```

```
Out[44]: 0.0
```

```
In [45]: cosine_sim(q, doc3)
```

```
Out[45]: 0.2993266539431952
```