CS 6370 Final Project

Tapish Garg[MM17B034] and Srijan Kumar Upadhyay[NA17B034]

IIT Madras, Chennai
mm17b034@smail.iitm.ac.in
na17b034@smail.iitm.ac.in

Abstract. In order to improve the information retrieval system based on Vector space model, we have experimented with different methods in this project. In preprocessing, contractions and spell checks have been introduced to improve the retrieval. To address limitations of VSM model, bigram models are tested to add a sense of sequence and context, LSA is also experimented with to address the orthogonality limitation. Two other methods have been looked upon to increase the effectiveness of the retrieval system, the first one is the Query Expansion and second one is the BM25 model. All such changes were compared with original model through hypothesis testing using t-test.

Keywords: Information retrieval · Cranfield · Spell Check · VSM · Bigram · LSA · Query Expansion · BM25.

1 Introduction

Information retrieval (IR) is the process of obtaining information system resources that are relevant to an information need from a collection of those resources. In this project attempted information retrieval on Cranfield Dataset which is a collection of 1400 aerospace related documents and 225 queries. We started with basic Vector Space Model (VSM). As this model ignores the fact of co-occurrence and order of approach, we used VSM with bigram which will add local word to word coherence. VSM does take semantic information into account for which we implemented Latent Semantic Analysis (LSA) which will also reduce dimensional vectors. To capture the information of synonym, query expansion has been used to expand queries with similar words. Relevance of document is calculated by tfidf vector which takes care of frequency of words and occurrence of word in documents but this is not sufficient. We used Okapi BM25 to take document length as a parameter also which will help to retrieve short documents.

2 Problem Definition

We have implement the Vector space model that uses Cosine similarity between document vector and query vector to determine similarity between them. As this the basic model so it has many limitations that can be solved using other methods.

3 Motivation

The basic vector space model (VSM) model is enough for starter but it has some limitations which need to addressed to improve information retrieval.

1. The order in which terms appear in the document is lost in vector space representation.
2. Given two documents, one of a higher length than other and having same number of occurrence of the query terms, then the dot product of the query vector with both the doc vectors will be same but for the longer document the norm will be higher and hence, the cosine similarity will be low. Therefore, long documents are poorly represented.
3. The model assumes that the words are statistically independent.
4. The model doesn't also consider the semantic relatedness of the words.

4 Proposed Methodology

1. **Enlisting the limitations of VSM Model**
   The aim of the project is to overcome the limitations of the VSM model, therefore, the limitations of the VSM model are identified(the same have been described in the Motivation.
2. **Addresing the limitation of absence of coherence** Unigrams model lacks a sense of coherence. To address that both unigrams and bigrams were considered, bigrams add a sense of continuity and assist in word sense disambiguation as it considers the knowledge about the immediate neighbours.
3. **Latent Semantic Analysis**
   Vector space model assumes orthogonality of dimensions which is not true in general. LSA however, projects documents in a concept space, where similar words are situated close to each other.
4. **Query Expansion**
   The query provided by the user is often unstructured and incomplete. An incomplete query hinders a search engine from satisfying the user's information need and affects the performance of IR model.
5. **BM25**
   BM25 is an modified version of tf-idf which addresses the word saturation. It will solve saturate the term frequency and considers document length also to calculate relevance. This method reward documents that match more of the terms in a multi-term query over documents that have lots of matches for just one of the terms.

5 Experiments:

5.1 Vector Space Model

Vector space model represents documents and queries as vectors. Each unique term represents a dimension of this vector space. The tf-idf expresses the weight along each one of this dimensions. Vector space model is the first and base model implemented for this project.

Preprocessing Steps The following preprocessing steps are applied and in the same sequence as mentioned below.

1. **Converting words to lowercase**: All the texts have been converted to lowercase, this is to prevent different treatment of words due to capitalisation of some characters.
2. **Expanding Contractions**: Contractions are shortened forms of words or syllables. All the contractions such as 'won't', 'can't' has been expanded to 'would not' and 'can not'. This helps in text standardisation and prevent losing information after apostrophe is removed.
3. **Removing punctuations, special characters and extra spaces**: Only the lowercase characters are kept and rest all characters types are eliminated. This include punctuation, numerical and alphanumeric characters.
4. **Removing stopwords**: The stopwords have little to no significance when constructing meaningful features from the text. They include articles, conjunctions and prepositions. They have maximum frequency among all words and have very low tf-idf score.
5. **Inflection Reduction**: We have used lemmatization to reduce words to their base form. It is different from stemming in the way that the root word formed is always lexically correct.
6. **Spell Check**: Spell check is performed on queries, Norvig spell checker based on the levenshtein distance.

   Norvig Spell Check Procedure:
   (a) First a new concept: a simple edit to a word is a deletion (remove one letter), a transposition (swap two adjacent letters), a replacement (change one letter to another) or an insertion (add a letter). The function returns a set of all the edited strings (whether words or not) that can be made with one and two simple edits.
   (b) Probability of a word is estimated, P(word), by counting the number of times each word appears in a text file of about a million words, big.txt. It is a concatenation of public domain book excerpts from Project Gutenberg and lists of most frequent words from Wiktionary and the British National Corpus. The function words breaks text into words, then the variable WORDS holds a Counter of how often each word appears, and P estimates the probability of each word.

(c) The word with the highest probability is substituted in place of incorrect word.

Ranking and Evaluation Methodology:

1. **Ranking:** The cosine product between the preprocessed query and document vectors is calculated. The documents are ranked based on the cosine value. Higher the cosine product between a query and a document vectors, higher the rank of the document.
2. **Evaluation:** VSM model is evaluated based on multiple metrics such as MAP@k, nDCG@k, Precison@k,F-score@k and Recall@k.

Results and Evaluation:

The following table enlists the metrics at different values of k for Vector space model:

Table 1. Metrics and their values

k	MAP	nDCG	Precision	Recall	F score
1	0.6711	0.4933	0.6711	0.1156	0.1889
2	0.7088	0.3910	0.5422	0.1779	0.2514
3	0.7226	0.3784	0.4963	0.2356	0.2967
4	0.7222	0.3777	0.4511	0.2772	0.3176
5	0.7145	0.3788	0.4106	0.3056	0.3232
6	0.7090	0.3841	0.3763	0.3318	0.3250
7	0.6904	0.3915	0.3555	0.3612	0.3301
8	0.6836	0.3954	0.33	0.3798	0.3257
9	0.6713	0.3992	0.3091	0.3976	0.3209
10	0.6636	0.4025	0.2911	0.4131	0.3156

The below figure depicts the above tabular matrix in pictorial form.

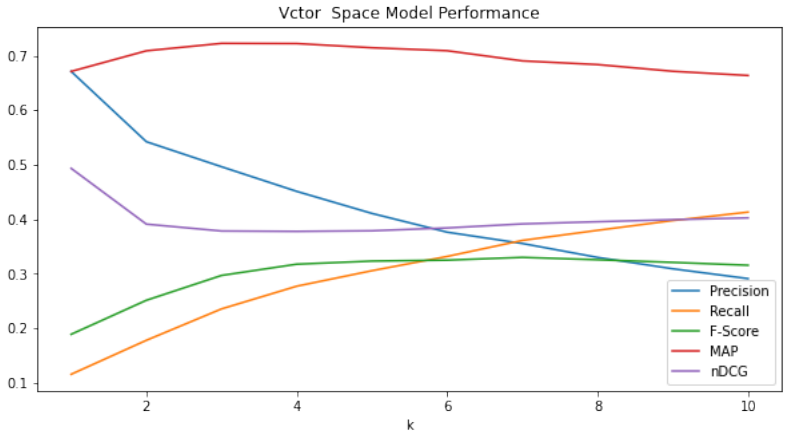


Fig. 1. Metrics

Observation:

From the **Fig. 1**, it can be observed that:

1. Precision decreases sharply with increase in k value, which is indicative of the fact that most relevant documents have already ranked at the beginning.
2. Recall increases with k as expected.
3. In systems that are higher in precision towards the left may favor precision over recall therefore, the given model is intended to be precise.

4. Though there is a decrease in the value of precision with the increase in K, the F-score shows an increase due to the steady increase in the value of recall. It is observed that with the value of precision drastically decreases the rate of increase of the F-score is also decreasing.
5. MAP decreases slowly as compared to precision. It is more smooth with lesser variations.
6. After an initial decrease in nDCG, the nDCG value keeps on increasing albeit slowly.

5.2 Bigram

In Vector space model in which only unigrams are considered, Documents and queries are viewed as bags of words, treating each term as independent of others terms. This approach ignores the fact that a large portion of meaning is carried in the co-occurrence and order of words.

To add an element of sequence, bigrams are also taken along with unigrams for constructing tf-idf matrix of the pre-processed text. The bigram add local word-to-word coherence.

Bigrams vs Unigrams VSM Model:

Table 2. Metrics and their values

	VSM with Unigrams					VSM with Unigrams and Bigrams				
k	**MAP**	**nDCG**	**Precision**	**Recall**	**F score**	**MAP**	**nDCG**	**Precision**	**Recall**	**F score**
1	0.6711	0.4933	0.6711	0.1156	0.1889	0.6977	0.5259	0.6977	0.1190	0.1953
2	0.7088	0.3910	0.5422	0.1779	0.2514	0.7466	0.4279	0.5977	0.1962	0.2781
3	0.7226	0.3784	0.4963	0.2356	0.2967	0.7533	0.3987	0.5170	0.2477	0.3121
4	0.7222	0.3777	0.4511	0.2772	0.3176	0.7535	0.3920	0.4511	0.2797	0.3199
5	0.7145	0.3788	0.4106	0.3056	0.3232	0.7348	0.3912	0.4115	0.3129	0.3282
6	0.7090	0.3841	0.3763	0.3318	0.3250	0.7270	0.3961	0.3748	0.3380	0.3282
7	0.6904	0.3915	0.3555	0.3612	0.3301	0.7134	0.4030	0.3536	0.3642	0.3305
8	0.6836	0.3954	0.3300	0.3798	0.3257	0.7061	0.4054	0.3255	0.3792	0.3226
9	0.6713	0.3992	0.3091	0.3976	0.3209	0.6989	0.4099	0.3022	0.3940	0.3151
10	0.6636	0.4025	0.2911	0.4131	0.3156	0.6868	0.4141	0.2849	0.4096	0.3103

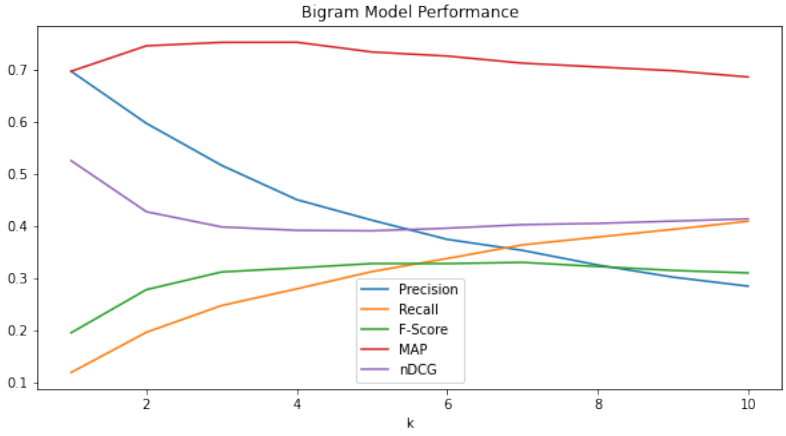


Fig. 2. Metrics

Sample:

Relevant = [32, 67, 164, 639, 715, 716, 719, 1379, 717, 499]
VSM = [1346, 706, 660, 1348, 286, 554, 1000, 1242, 944, 982]
VSM-bigarm = [1346, 660, 706, 1348, 554, 164, 163, 1000, 982, 1242]

Doc 164 = an approximate analytical method for studying entry into planetary atmospheres . the pair of motion equations for entry into a planetary atmosphere is reduced to a single, ordinary, nonlinear **differential equation** of second order by disregarding two relatively small terms and by introduring a certain mathematical transformation . the reduced equation includes various terms, certain of which represent the gravity force, the centrifugal acceleration, and the lift force . if these particular terms are disregarded, the **differential equation** is linear and yields precisely the solution of allen and eggers applicable to ballistic entry at relatively steep angles of descent . if all the other terms in the basic equation are disregarded (corresponding to negligible vertical acceleration and negligible vertical component of drag force), the resulting truncated **differential equation** yields the solution of sanger for equilibrium flight of glide vehicles with relatively large lift-drag ratios . a number of solutions for lifting and nonlifting vehicles entering at various initial angles also have been obtained from the complete nonlinear equation . these solutions are universal in the sense that a single solution determines the motion and heating of a vehicle of arbitrary weight, dimensions, and shape entering an arbitrary planetary atmosphere . one solution is required for each lift-drag ratio . these solutions are used to study the deceleration, heating rate, and total heat absorbed for entry into venus, earth, mars, and jupiter . from the equations developed for heating rates, and from available information on human tolerance limits to acceleration stress, approximate conditions for minimizing the aerodynamic heating of a trimmed vehicle with constant lift-drag ratio are established for several types of manned entry . a brief study is included of the process of atmosphere braking for slowing a vehicle from near escape velocity to near satellite velocity ."

Observation:

As we have taken Bigrams along with unigrams in this model, query and docs having co-occurring words are better captured and the results obtained are more relevant.

Hypothesis testing Bigram vs Unigram VSM:

- **Null Hypothesis:** Bigram and Unigram Model have similar values of precision, recall, fscore, MAP and nDCG.
- **Alternate Hypothesis:** Bigram and Unigram Models have different values of precision, recall, fscore, MAP and nDCG.
- **Approach:** A two sampled t-test is applied with 95% confidence value to determine, whether the values of the metrics of the two models are equal or not for the 225 queries in the Cranfield queries.
- **Results:**

Table 3. Metrics and their values

Metric	t-statistics	p-value
Precision	0.336127	0.73692
Recall	0.13395	0.839514
F-Score	0.29812	0.76571
nDCG	-0.506907	0.612469
Average Precision	-0.7865	0.4319

- **Conclusion:** All the p-values are much greater than 0.05 (required for 95%). Therefore, Null hypothesis cannot be rejected and conclusion can be derived that both bigram and unigram model performs for the given queries.

5.3 Latent Semantic Analysis

Two documents are considered similar if they have similar words and two words are similar if they occur in similar documents, this leads to a circularity.

LSA measures semantic information through co-occurrence analysis in the corpus. The document feature vectors are projected into the subspace spanned by the first k singular vectors of the feature space. This also reduces the amount of storage required for high dimensional document vectors.

LSA Procedure:

1. **Singular value decomposition (SVD)** decomposes the tf-idf matrix A into three matrices **U,V and** Σ, such that

$$A = U\Sigma V^T$$

where U is $m \times m$ and V is $n \times n$ square matrices, such that $UU^T = I$; $VV^T = I$ (orthonormal matrices), and Σ is an $m \times n$ diagonal matrix with singular values on the diagonal. In addition, the singular values are non-negative and are ordered from largest to smallest in the diagonal of Σ.

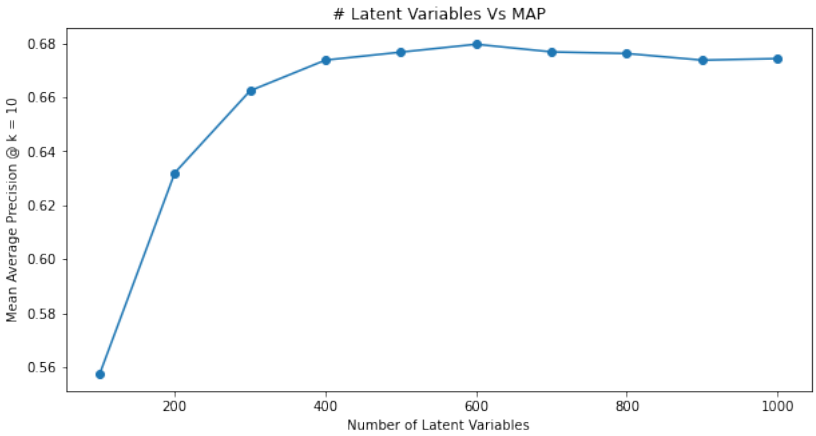
2. **Dimension Reduction:** By removing dimensions corresponding to small singular values and keeping the dimensions corresponding to larger singular values, the representation of each term is reduced to a smaller vector with only k dimensions. k is the number of concepts.

$$A_k = U_k \Sigma_{k \times k} V_k^T$$

The new term vectors (rows in the reduced U matrix, U_k) are no longer orthogonal, but the advantage of this is that only the most important dimensions that correspond to larger singular values are kept.

3. **Transformation:** Query and Documents are transformed into concept space. Cosine product is obtained between new query vector and new documents vectors. The documents with highest cosine product with the query is ranked first and so on.

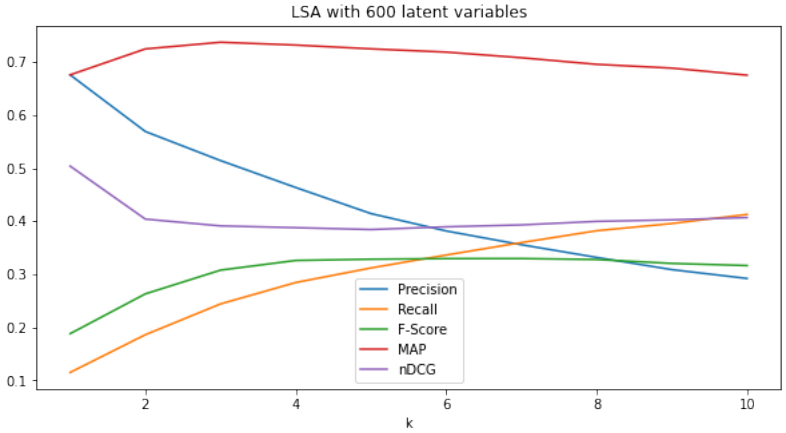
LSA Hyperparameter Tuning:



LSA with maximum MAP at k=10 has **600** latent variables.

LSA vs Unigram Model:

	VSM with Unigrams					LSA with 600 hyperparameters				
k	**MAP**	**nDCG**	**Precision**	**Recall**	**F score**	**MAP**	**nDCG**	**Precision**	**Recall**	**F score**
1	0.6711	0.4933	0.6711	0.1156	0.1889	0.6755	0.5037	0.6755	0.1149	0.1879
2	0.7088	0.3910	0.5422	0.1779	0.2514	0.7244	0.4038	0.5688	0.1859	0.2629
3	0.7226	0.3784	0.4963	0.2356	0.2967	0.7370	0.3910	0.5140	0.2440	0.3077
4	0.7222	0.3777	0.4511	0.2772	0.3176	0.7317	0.3876	0.4633	0.2843	0.3259
5	0.7145	0.3788	0.4106	0.3056	0.3232	0.7245	0.3841	0.4142	0.3117	0.3282
6	0.7090	0.3841	0.3763	0.3318	0.3250	0.7184	0.3893	0.3814	0.3361	0.3294
7	0.6904	0.3915	0.3555	0.3612	0.3301	0.7078	0.3929	0.3555	0.3595	0.3295
8	0.6836	0.3954	0.3300	0.3798	0.3257	0.6954	0.3995	0.3316	0.3819	0.3276
9	0.6713	0.3992	0.3091	0.3976	0.3209	0.6882	0.4024	0.3086	0.3955	0.3203
10	0.6636	0.4025	0.2911	0.4131	0.3156	0.6749	0.4067	0.2920	0.4126	0.3163



Hypothesis testing LSA vs VSM:

- **Null Hypothesis:** LSA and Unigram Vector Space Model have similar values of precision, recall, fscore, MAP and nDCG.
- **Alternate Hypothesis:** LSA and Unigram Vector Space Model have different values of precision, recall, fscore, MAP and nDCG.
- **Approach:** A two sampled t-test is applied with 95% confidence value to determine, whether the values of the metrics of the two models are equal or not for the 225 queries in the Cranfield queries.
- **Results:**

Table 4. Metrics and their values

Metric	t-statistics	p-value
Precision	-0.0479	0.9617
Recall	0.0164	0.9868
F-Score	-0.0372	0.9703
nDCG	-0.1866	0.8520
Average Precision	-0.3781	0.7055

- **Conclusion:** All the p-values are much greater than 0.05 (required for 95%). Therefore, Null hypothesis cannot be rejected and conclusion can be derived that both LSA and Unigram Vector Space Model performs for the given queries.

5.4 Query Expansion Model

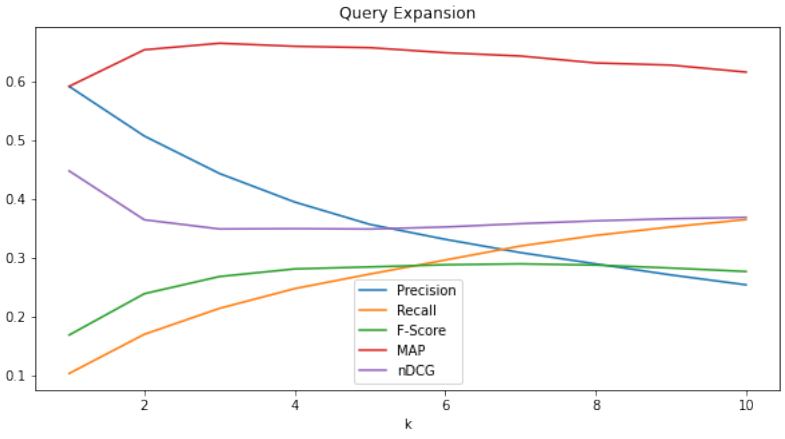
VSM works very well if the user is able to convey his information need in form of query. But query is seldom complete. The query provided by the user is often unstructured and incomplete. An incomplete query hinders a search engine from satisfying the user's information need.
In query expansion, the user's original query is expanded by adding terms that are synonymous with or related to the original terms. Query expansion is thus a technique for improving recall, perhaps at the expense of precision. In comparison to naive vector space model, use of query expansion will results in documents which are not even in query.

Procedure:

1. For each word in a query, three synonyms are generated from WordNet.
2. The generated synonyms are attached to the main word.
3. Doing this for all words results in an expanded query.

Query Expansion vs VSM Model:

	VSM with Unigrams					Query Expansion				
k	**MAP**	**nDCG**	**Precision**	**Recall**	**F score**	**MAP**	**nDCG**	**Precision**	**Recall**	**F score**
1	0.6711	0.4933	0.6711	0.1156	0.1889	0.5911	0.4474	0.5911	0.1031	0.1684
2	0.7088	0.3910	0.5422	0.1779	0.2514	0.6533	0.3643	0.5066	0.1697	0.2386
3	0.7226	0.3784	0.4963	0.2356	0.2967	0.6644	0.3487	0.4429	0.2138	0.2678
4	0.7222	0.3777	0.4511	0.2772	0.3176	0.6592	0.3492	0.3944	0.2473	0.2808
5	0.7145	0.3788	0.4106	0.3056	0.3232	0.6569	0.3485	0.3564	0.2721	0.2842
6	0.7090	0.3841	0.3763	0.3318	0.3250	0.6483	0.3520	0.3311	0.2959	0.2878
7	0.6904	0.3915	0.3555	0.3612	0.3301	0.6427	0.3577	0.3085	0.3197	0.2894
8	0.6836	0.3954	0.3300	0.3798	0.3257	0.6309	0.3625	0.2894	0.3377	0.2873
9	0.6713	0.3992	0.3091	0.3976	0.3209	0.6272	0.3661	0.2706	0.3522	0.2823
10	0.6636	0.4025	0.2911	0.4131	0.3156	0.6153	0.3684	0.2537	0.3647	0.2764



Hypothesis Testing:

- **Null Hypothesis:** Query Expansion and Unigram Vector Space Model have similar values of precision, recall, fscore, MAP and nDCG.
- **Alternate Hypothesis:** Query Expansion and Unigram Vector Space Model have different values of precision, recall, fscore, MAP and nDCG.
- **Approach:** A two sampled t-test is applied with 95% confidence value to determine, whether the values of the metrics of the two models are equal or not for the 225 queries in the Cranfield queries.

– **Results:**

Table 5. Metrics and their values

Metric	t-statistics	p-value
Precision	2.058	0.0401
Recall	1.8700	0.0621
F-Score	2.1979	0.0284
nDCG	1.4633	0.1440
Average Precision	-1.5478	0.122

– **Conclusion:** Some p-values are much less than 0.05 (required for 95%). Therefore, Null hypothesis can be rejected and conclusion can be derived that both Query Expansion and Unigram Vector Space Model performs differently for the given queries.

5.5 Okapi BM25 Model:

BM25 is the improvement of TF-IDF function. It keeps the structure the same and refines TF and IDF components.

Once a document is saturated with occurrences of a term, more occurrences shouldn't a have a significant impact on the score. So we'd like a way to control the contribution of TF to our score. One common way to tame TF is to take the square root of it, but that's still an unbounded quantity. A more sophisticated way is to:

$$TF_{new} = \frac{TF}{k + TF}$$

This put a bound on TF's contribution to the score, and also control how rapidly the contribution approaches that bound.

If a document happens to be really short and it contains a word say IIT once, that's a good indicator that IIT is important to the content. But if the document is really, really long and it mentions a IIT only once, the document is probably not about IIT. So we'd like to reward matches in short documents, while penalizing matches in long documents. To define short and long, we define short as smaller than average corpus length and long, longer than corpus length.

As we have modified TF to $\frac{TF}{TF+k}$ trick. Of course as k increases, the value of $\frac{TF}{TF+k}$ decreases. To penalize long documents, k should be adjusted up if the document is longer than average, and adjusted down if the document is shorter than average. The modified formula for the Term Frequency therfore is,

$$TF_{new} = \frac{TF}{k\frac{DL}{DL_{avg}} + TF}$$

Here, DL is the document's length, and DL_{avg} is the average document length across the corpus.

When a document is of average length, $\frac{DL}{DL_{avg}} = 1$, and our multiplier doesn't affect k at all. For a document that's shorter than average, we'll be multiplying k by a value between 0 and 1, thereby reducing it, and increasing $\frac{TF}{TF+k}$. For a document that's longer than average, we'll be multiplying k by a value greater than 1, thereby increasing it, and reducing $\frac{TF}{TF+k}$. The multiplier also puts us on a different TF saturation curve. Shorter documents will approach a TF saturation point more quickly while longer documents will approach it more gradually.

$$TF_{new} = \frac{TF}{k(1 - b + b\frac{DL}{DL_{avg}}) + TF}$$

Another point to consider is how much importance should we place on document length in any particular corpus? Might there be some collections of documents where length matters a lot and some where it doesn't? Therefore, document length can be treated as a second parameter

that can be experimented with.

The IDF is also modified as:

$$IDF_{new} = log\frac{1 + (N - DF + .5)}{DF + .5}$$

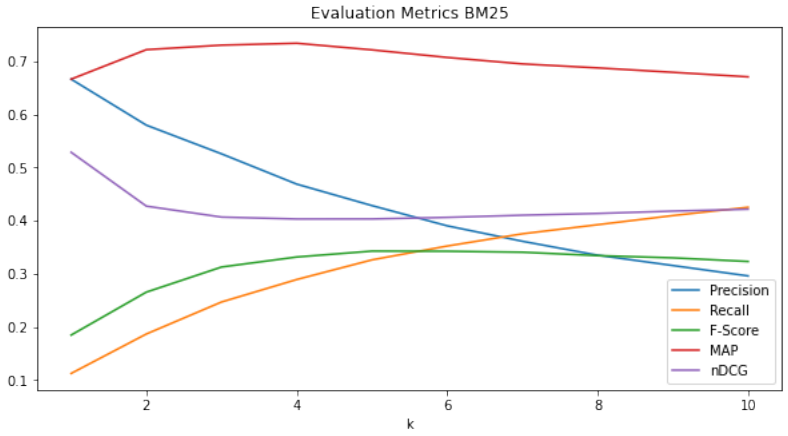
where N is the total number of documents and DF is the Document Frequency.

The final score is therefore, given by:

$$Score = \frac{TF}{k(1 - b + b\frac{DL}{DL_{avg}}) + TF} \times log\frac{1 + (N - DF + .5)}{DF + .5}$$

BM25 vs Unigram Model:

	VSM with Unigrams					BM25				
k	**MAP**	**nDCG**	**Precision**	**Recall**	**F score**	**MAP**	**nDCG**	**Precision**	**Recall**	**F score**
1	0.6711	0.4933	0.6711	0.1156	0.1889	0.6666	0.4288	0.6666	0.1122	0.1845
2	0.7088	0.3910	0.5422	0.1779	0.2514	0.7222	0.4274	0.5800	0.1865	0.2654
3	0.7226	0.3784	0.4963	0.2356	0.2967	0.7307	0.4068	0.5259	0.2468	0.3127
4	0.7222	0.3777	0.4511	0.2772	0.3176	0.7343	0.4032	0.4688	0.2892	0.3316
5	0.7145	0.3788	0.4106	0.3056	0.3232	0.7218	0.4033	0.4284	0.3262	0.3427
6	0.7090	0.3841	0.3763	0.3318	0.3250	0.7075	0.4062	0.3903	0.3522	0.3424
7	0.6904	0.3915	0.3555	0.3612	0.3301	0.6953	0.4103	0.3612	0.3753	0.3406
8	0.6836	0.3954	0.3300	0.3798	0.3257	0.6878	0.4136	0.3350	0.3924	0.3342
9	0.6713	0.3992	0.3091	0.3976	0.3209	0.6795	0.4180	0.3155	0.4096	0.3300
10	0.6636	0.4025	0.2911	0.4131	0.3156	0.6710	0.4216	0.2960	0.4253	0.3232



Sample:

Query = 185
Relevant = [858, 859, 857, 1008, 856, 15, 285, 894, 766, 948]
VSM = [856, 1008, 766, 857, 859, 858, 391, 948, 658, 864]
BM25 = [856 857 858 1008 766 390 859 948 391 864]

Doc 858 ="experimental investigation at mach numbers 3. 0 of the effects of thermal stress and buckling on the **flutter** of four-bay aluminium alloy panels with length-width ratios of 10 . skin-stiffener aluminum alloy panels consisting of four bays, each bay having a length-width ratio of 10, were tested at a mach number of 3.0 at dynamic pressures ranging from 1,500 psf to 5,000 psf and at stagnation temperatures from 300 f to 655 f . the panels were restrained by the

supporting structure in such a manner that partial thermal expansion of the skins could occur in both the longitudinal and lateral directions . a boundary faired through the experimental **flutter** points consisted of a flat-panel portion, a buckled-panel portion, and a transition point at the intersection of the two boundaries . in the region where a panel must be flat when **flutter** occurs, an increase in panel skin temperature (or midplane compressive stress) makes the panel more susceptible to **flutter** . in the region where a panel must be buckled when flutter occurs, the **flutter** trend is reversed . this reversal in trend is attributed to the panel postbuckling behavior ."

Observation:

As the document length is less that's why BM25 gave higher importance to the word flutter while assigning weight to the word. Whereas in VSM the word has a low weight assigned to it as the tf-idf value is less.

Hypothesis Testing:

- **Null Hypothesis:** BM25 and Unigram Vector Space Model have similar values of precision, recall, fscore, MAP and nDCG.
- **Alternate Hypothesis:** BM25 and Unigram Vector Space Model have different values of precision, recall, fscore, MAP and nDCG.
- **Approach:** A two sampled t-test is applied with 95% confidence value to determine, whether the values of the metrics of the two models are equal or not for the 225 queries in the Cranfield queries.
- **Results:**

Metric	**t-statistics**	**p-value**
Precision	-0.2673	0.7892
Recall	-0.4716	0.6374
F-Score	-0.4240	0.6717
nDCG	-0.8464	0.3977
Average Precision	-0.2474	0.8046

- **Conclusion:** Some p-values are much greater than 0.05 (required for 95%). Therefore, Null hypothesis cannot be rejected and conclusion can be derived that both BM25 and Unigram Vector Space Model performs differently for the given queries.

6 Conclusion:

Many preprocessing techniques were employed including, lowering the text, expanding the contractions, removing the punctuations and queries have been spell corrected using a model based on Norvig Spell Checker. Vector Space model considering only unigrams was deployed on the preprocessed text. However, the model itself has many limitations therefore, to address these limitations we used models which tackle one or more limitations of the VSM model. To add an element of sequence to the model, bigrams were taken along with unigrams. On applying hypothesis test on the metrics of Precision, recall, F-score, Average Precision and nDCG, we were unable to reject the null hypothesis(that both models perform similar in terms of the above mentioned metrics. Another important point to note is that VSM model takes the words to be orthogonal which maynot be true in real world. To address that LSA has been taken, surprising here again we were unable to reject the null hypothesis. Query expansion model despite taking the expanded queries did not improved the efficacy of the model. Okapi BM25 model improvises upon the tf-idf used in Unigram VSM model, it also takes document length into account. However, on hypothesis testing the smaller p values indicated that we couldn't reject the null hypothesis.

References:

1. https://norvig.com/spell-correct.html
2. https://trec.nist.gov/pubs/trec27/papers/JARIR-CC.pdf
3. https://kmwllc.com/index.php/2020/03/20/understanding-tf-idf-and-bm-25/
4. http://sifaka.cs.uiuc.edu/course/410/note/vs-old.html
5. https://core.ac.uk/download/pdf/41373831.pdf