Birla Institute of Technology & Science

A Report on : Vector Space IR model

Group No. :- 17

PREPARED BY

Utkarsh Kumar (2017B2A71008)

Supratik Bhattacharya (2017B2A70745P)

Kshitij gupta (2017B3A70601)

Shivankur (2017B4A71013P)

Tanmay Khandelwal(2017B3A70725P)

SUBMITTED TO

Dr Vinti Agarwal

Dr Abhishek

OVERVIEW

This project gives an glimpse of implementation of vector space information retrieval method useful for computing the score between the document and query using the the lnc.ltc scoring scheme and it goes on increasing the efficiency and time complexity using phrasal queries and champion list.It not only describe the advantages of using such improvements query wise but also covers the disadvantages and shortcomings of using fast methods such as champion lists which reduces the result quality , but at the same time decrease the time complexity which could cater to the practical industrial needs of such complex and huge models.

Outline

1. Vector Space IR model
2. Improvements
    a. Phrasal queries
    b. Champion List
3. Innovation
4. References

Part 1

Vector Space IR model

The Vector space IR model uses the inverted index posting list inorder to calculate the score of the document and the query .The inverted index posting list is created using the hashed index data structure and the tokenizer function of the nltk library.The list is then used to calculate the cosine similarity score, using the lnc.ltc scoring scheme.

Evaluation of the model w/o improvement with 10 queries and their result

Query 1: Service Provider

Top 10 Doc id:	Score	Relevant To Doc
47720641	0.1856654691	Yes
47742098	0.1624664051	No
47740903	0.1593743243	Yes
47754754	0.1199893922	No
47760651	0.1078378222	No
47780142	0.09899016305	No
47757151	0.09513058596	No
47763680	0.09380002272	No
47720961	0.08912961199	Yes
47724329	0.08898651491	No

Query 2: American Actor and French Director

Top 10 Doc id:	Score	Relevant To Doc
47739722	0.3019584204	Yes
47746817	0.22809294	Yes
47727187	0.1938089898	Yes
47773457	0.1935052754	Yes
47728192	0.1874528596	No
47740272	0.1862371056	No
47754896	0.1791942269	No
47724614	0.1668207466	Yes
47726317	0.1662226397	No
47780658	0.1641269449	No

Query 3: mindspark crazy

Top 10 Doc id:	Score	Relevant To Doc
47734460	0.09759000729	yes
47757452	0.05792844464	no
47729297	0.04032389193	yes
47725278	0.04022589934	no
47737903	0.03901371573	no
47784879	0	No
47784877	0	No
47784865	0	No
47784857	0	No

47784850	0	No

Query 4: American band

Top 10 Doc id:	Score	Relevant To Doc
47770942	0.2569778881	Yes
47767194	0.2506226425	No
47770919	0.209821567	No
47770714	0.1864692131	No
47734368	0.1864313546	No
47739722	0.1842888087	No
47748758	0.1757124847	No
47760305	0.1732544568	No
47738998	0.1616319775	No
47783496	0.1563917623	No

Query 5: American actor Charlie Gharaee

Top 10 Doc id:	Score	Relevant To Doc
47739722	0.2180906543	No
47773457	0.1780702735	No
47720177	0.1778772452	Yes
47746817	0.1430607701	No
47772524	0.1220026346	No
47743606	0.1220026346	No

47726317	0.1200549539	No
47740272	0.1168086296	No
47734136	0.1114034832	No
47734848	0.1104344956	No

Query 6: Maroon 5

Top 10 Doc id:	Score	Relevant To Doc
47772524	0.1414966313	No
47740272	0.1354727105	No
47738453	0.1173228088	No
47763310	0.1138871254	Yes
47784422	0.09785398691	No
47739694	0.09785398691	No
47760720	0.09659533473	Yes
47773009	0.09579367223	No
47765935	0.09579367223	No
47765695	0.09579367223	No

Query 7: Bits Pilani

Top 10 Doc id:	Score	Relevant To Doc
47778472	0.04608476701	No
47784169	0.04241570049	No
47751162	0.03809550187	No
47738673	0.03514956543	Yes

47774416	0.01432728805	No
47784879	0	No
47784877	0	NO
47784865	0	No
47784857	0	No
47784850	0	No

Query 8: American Comedian

Top 10 Doc id	Score	Relevant To Doc
47734848	0.2189070786	Yes
47756440	0.2096917636	No
47727187	0.1605487151	No
47739722	0.1589022318	No
47739896	0.1578502251	No
47748758	0.1515073334	No
47761535	0.1402536056	No
47738998	0.1393664767	No
47760579	0.1327819909	No
47773457	0.129743129	No

Query 9: mild stroke

Top 10 Doc id	Score	Relevant To Doc
47729044	0.05773502692	No
47720197	0.0535671584	Yes

47741688	0.04902903378	No
47744911	0.04508348173	No
47764552	0.04393747752	No
47752401	0.04256282654	No
47756506	0.03834824944	No
47755695	0.0308313208	No
47760958	0.0306858206	No
47749084	0.03001501126	No

Query 10: English politician

Top 10 Doc id	Score	Relevant To Doc
47740029	0.3159551721	Yes
47740133	0.2946297172	Yes
47739853	0.2825988971	Yes
47722195	0.2257144787	No
47748498	0.2141315561	No
47740486	0.2124219007	No
47720731	0.2041664279	No
47731261	0.2018563551	Yes
47771690	0.1954744726	No
47738956	0.1784700942	No

Proposed Improvements

1. Phrasal queries :

A query may contain phrases but the vector space IR system does not handle phrasal queries.So to improve this a bigram index to handle two word phrasal queries is created. It is similar to the vector space IR system but the difference is that instead of one word, two words act as an index of Hashmap.

It uses the bigram inverted index posting list inorder to calculate the score of the document and the query .The bigram inverted index posting list is created using the hashed index data structure and the ngram function of the nltk library.The list is then used to calculate the cosine similarity score, using the lnc.ltc scoring scheme.

1. What is the issue with the IR system built in part 1?

The issue with the IR system is that it does not handle bigram or ngram phrase query well.

2. What improvement are you proposing?

We have created a bigram-inverted index to facilitate bigram phrasal queries.

3. How will the proposed improvement address that issue?

Queries where a bigram (2 word phrases) is present are handled much better as shown in results.

4. A corner case (if any) where this improvement might not work or can have an adverse effect.

This improvement is only for bigram (2 word phrases) and not for n-gram. So if a query which has 3 or 4 word phrases, then the IR system may not give most accurate results.

5. Demonstrate the actual impact of the improvement. Give three queries, where the improvement yields better results compared to the part 1 implementation.

Query 1 : American comedian

Without Improvement		Improvement	
Doc id	Score	Doc id	Score
47734848	0.2189070786	47734848	1
47756440	0.2096917636	47778286	1
47727187	0.1605487151	47756440	0.2096917636
47739722	0.1589022318	47727187	0.1605487151
47739896	0.1578502251	47739722	0.1589022318
47748758	0.1515073334	47739896	0.1578502251
47761535	0.1402536056	47748758	0.1515073334
47738998	0.1393664767	47761535	0.1402536056
47760579	0.1327819909	47738998	0.1393664767

47773457	0.129743129	47760579	0.1327819909

Improvements :

- Score of Doc id 47734848 increases
- A relevant document with Doc id 47778286 is return which was not present in earlier IR system

Query 2 : mild stroke

Without Improvement		Improvement	
Doc id	Score	Doc id	Score
47729044	0.05773502692	47720197	1
47720197	0.0535671584	47729044	0.05773502692
47741688	0.04902903378	47741688	0.04902903378
47744911	0.04508348173	47744911	0.04508348173
47764552	0.04393747752	47764552	0.04393747752
47752401	0.04256282654	47752401	0.04256282654
47756506	0.03834824944	47756506	0.03834824944
47755695	0.0308313208	47755695	0.0308313208
47760958	0.0306858206	47760958	0.0306858206
47749084	0.03001501126	47749084	0.03001501126

Improvements :

- Score of Doc id 4772019 increases

Query 3 : English politician

Without Improvement		Improvement	
Doc id	Score	Doc id	Score
47740029	0.3159551721	47763400	1
47740133	0.2946297172	47740029	1
47739853	0.2825988971	47739853	1
47722195	0.2257144787	47731261	1
47748498	0.2141315561	47737714	1
47740486	0.2124219007	47730962	1
47720731	0.2041664279	47740133	1
47731261	0.2018563551	47722195	0.2257144787
47771690	0.1954744726	47748498	0.2141315561
47738956	0.1784700942	47740486	0.2124219007

Improvements :

- Four relevant Documents with doc id 47763400,47731261,47737714,47730962 are returned which were not present in the earlier IR system.
- Score of Doc id 47740029,47739853,47740133 increases

2. Champion List

It is an improvement in the vector space model decreasing the time complexity at the cost of some degree of loss in the top K documents and relative ordering.

For every term (t), store a list of r documents that have the highest score for term t.

i) The score can be tf score.

ii) r is fixed at the index creation time, thus it's possible that r < K.But we have taken r comparable to K. The set of r documents are called the champion list for a term t.

1. What is the issue with the IR system built in part 1?

Solution 1: Major issue with the above IR system is HIGH LATENCY ,in the above model for each query we need to find the score of each document in the corpus and then return the list of top K highest scoring document lists. However, score computation is a large fraction of the CPU work on a query. Having a tight budget on latency of order of 250ms this could be reduced by order of 100s.

2. What improvement are you proposing

Solution 2: Proposed solution is to look at ways of cutting CPU usage for scoring ,without compromising the quality of results. Basic idea is to avoid calculating scores for the documents which would not make to top K in the final list.One of generic approach is to somehow reduce the document of contenders from set of N to A,where $K < |A| << |N|$ where N is the set of the entire corpus while A is its subset. We can consider A as a set of pruning non-contenders.
In order to achieve best results we need to consider two aspects in pruning

- Consider high idf query terms so as to allow only high scoring documents.
- Consider docs containing many queries i.e high tf values.

Later can be taken care of in the champion list method.

3. How will the proposed improvement address that issue

Using this we ensure that only high scoring documents are allowed in the top K list using the high term frequency documents ordered and pruning the rest.This allows only documents which have high probability of carrying the entire query to be there in the contenders list for top scoring docs.This ranking system would yield top K documents with almost same relative ordering and low CPU processing time.

4. A corner case (if any) where this improvement might not work or can have an adverse effect

• Safe ranking: It is guaranteed that the top K documents returned by an algorithm are the K absolute highest scoring documents.

• Non-safe: The top K documents returned are closer to the K absolute highest scoring documents.

This method is based on non -safe RANKING FUNCTION where we get list of top K docs which are close to the top scoring documents however are just a approximation of them.However it might be possible that the approximation of
These documents differ with the actual ranking by large amounts.

For example:

Query :American actor Charlie Gharaee

Champion list result

Doc Id	Score
47720177	0.1778772452
47735643	0.1074578426
47766004	0.1051953188
47780362	0.1011592406
47758494	0.08681296066
47768248	0.06766873196
47751576	0.0673042371
47721446	0.0626966549

47784531	0.05780528036
47765771	0.05589752415

Vector space model result

Doc Id	Score
47739722	0.2180906543
47773457	0.1780702735
47720177	0.1778772452
47746817	0.1430607701
47772524	0.1220026346
47743606	0.1220026346
47726317	0.1200549539
47740272	0.1168086296
47734136	0.1114034832
47734848	0.1104344956

We noticed in many queries that there are very few Doc Id that are common in the top K relevant doc Ids of the vector space model and champion list model.
Like the example above for the same queries we have overlap of just one Doc Id 47720177.

5. Demonstrate the actual impact of the improvement. Give three queries, where the improvement yields better results compared to the part 1 implementation.

The major advantage of this improvement is to reduce the computation time when compared to the vector space model by order of 100s while maintaining the top K relevant documents along with their relative positions irrespective of magnitude of their scores. And we have the result as

Query: American actor
Time taken to process query = 0.0199041110000002445 using champion list
Time taken to process query = 4.833670964999996 using vector space model
% decrease of time is 253%

Query: American actor and french director
Time taken to process query = 0.038306992999999907 using champion list
Time taken to process query = 4.230438964999998 using vector space model

% decrease of time is 110%

Query: mild stroke
Time taken to process query = 0.008697234999999637 using champion list
Time taken to process query = 3.8317121710000066 using vector space model

% decrease of time is 439%

Query: American actor Charlie Gharaee
Time taken to process query = 0.0304490700000003076 using champion list

Time taken to process query = 3.919483245000002 using vector space model

% decrease of time is 129%

In all the above cases the order of execution time is reduced by order of 100s which is an achievement of this improvement but at the cost of loss of some relevant documents with precision of abou 60% and loss in their relative ordering.

Innovations

We use data structures such as hashindex and sets for making an inverted index and bigram index which offers O(1) time complexity on retrieval of the terms.

References

https://nlp.stanford.edu/IR-book/
https://www.nltk.org/
https://en.wikipedia.org/wiki/Vector_space_model