International Journal on Emerging Technologies 11(1): 280-285(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Bose Einstein 1 and Bose Einstein 2 Model for Optimal Query Expansion

Abhishek Kumar Shukla[1] and Sujoy Das[2]
[1]Research Scholar, Department of Mathematics and Computer Applications,
Maulana Azad National Institute of Technology, Bhopal (Madhya Pradesh), India.
[2]Professor, Department of Mathematics and Computer Applications,
Maulana Azad National Institute of Technology, Bhopal (Madhya Pradesh), India.

(Corresponding author: Abhishek Kumar Shukla)
(Received 29 October 2019, Revised 21 December 2019, Accepted 26 December 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Queries are generally short, vague and contains fewer information about the documents. Therefore retrieval of relevant documents from short query is a tedious task. Another issue in Information Retrieval (IR) is problem of term mismatch. Thus degrades the performance of any retrieval system. A number of methods have been developed in order to enhance the performance of IR system. One of these methods is query expansion. Query Expansion is used to reduce problem of word mismatch. The proposed method uses two query expansion model Bo1 and Bo2 based on pseudo relevance feedback and tries to find the optimum number of document and term for which the system perform better. The proposed model ascertains that retrieval system performs better when expansion documents and terms are 8 and 15 respectively.

Keywords: Query Expansion, Pseudo Relevance Feedback, FIRE, Terrier 3.5, Information Retrieval.

Abbreviations: TF-IDF, Term frequency Inverse document frequency; DFR, Divergence from Randomness; FIRE, Forum for Information Retrieval; Bo1, Bose Einstein 1; Bo2, Bose Einstein 2; P@10, Precision at 10 documents; P@20, Precision at 20 documents.

I. INTRODUCTION

The goal of IR models is to retrieve and rank the relevant documents from a large collection of documents. Boolean models, Vector Space Model, Probabilistic models and Language models have been developed to retrieve and rank the relevant documents. Vector space and probabilistic retrieval models give significantly good result as compared to Boolean retrieval models [1]. Normally user query uses on an average two to three words, and this makes it difficult to formulate user's need. User query containing the word "Alebert Einstein" does not provide the user intention whether he wants to search Albert Einstein biography, research or any other thing related to Albert Einstein. This may retrieve documents that might be irrelevant. Query expansion is a technique which improves the performance of IR system [2]. Query expansion technique suggests some additional words so that additional documents which are relevant but left out are to be retrieved. Polysemy and synonyms is another problem for retrieval systems [3]. For example "Chips price" does not describe whether "chips" refers to the computer chips or edible chips. These shortcomings of the retrieval model lead to use of various natural language processing techniques. Query expansion techniques is broadly divided into two categories Global and Local method.

In global method query expansion terms are expanded or reformulated independent of the original query. To reformulate the query new query words are added that have semantically same meaning as the original query words. Global methods include synonyms search, use of thesaurus like WordNet or query expansion using automatic generated thesaurus. Local methods include Pseudo relevance feedback and Indirect relevance feedback. According to 29.3% of the total searchers on the internet add one or more than one keyword and 32.5% repeat their query to find the relevant documents [4]. Several query expansion approach including relevance feedback, query filtering, interactive query filtering and result re-ranking and clustering techniques have been developed [5]. In this regard, *query expansion reduces the word mismatch problem by expanding the highly correlate terms or the terms that poses some statistical relationship.* To find such correlations between terms, a number of statistical measures, have been proposed, such as term Co-occurrence measures or lexical Co-occurrence measures [6, 7].

This paper is arranged as follows. Section II discusses related work regarding to query expansion. Section III discusses the material and methods for the query expansion. Section IV explores result discussion and finally Section V concludes the article and section VI discusses future scope of the article.

II. RELATED WORK

Query Expansion technique removes the ambiguity which exists in the user's need and hence it improves the performance [5]. Semantic approach is another technique for query expansion because TF-IDF measure does not resolve the problem of semantics of the query. Expansion terms are created based on the grammatical relation between original query terms and their meaning of the term in the information space [8, 9].

WordNet is used for implementing semantic approaches [10]. Ontology is another semantic based approach which deals with domain specific knowledge [11]. Suggested ontology heredity concept to find the expansion terms by calculating the similarity between ontology and query words [12]. Alqadah and Bhatnagar (2011) data mining technique along with ontology to find the pattern on the query terms [13].

These methods suffer from vocabulary mismatch problem. To minimize the vocabulary mismatch, researcher proposed Latent Semantic Indexing (LSI) [14], Probabilistic Latent Semantic Analysis (pLSI) [15] and Latent Dirichlet Allocation(LDA) [16] method to find the semantic term of the query term to expand the original query. To compute the semantic terms Sergienko *et al.,* (2016) suggested the word embedding concept [17]. Word embedding is used in GloVe and Word2vec [18-19]. A model was proposed which process a random walk on Markov chain between the terms and the documents [20]. If two terms occurs in the same documents then they might be strongly correlated to each other [21, 22]. Shaalan *et al.,* (2012) used expectation maximization algorithm for query expansion [23]. Bai *et al.*, (2007) proposed query expansion using association rule mining [24]. Probabilistic approach used probabilistic distribution of terms in which terms with highest probability terms are considered as expansion terms [25]. Metzler and Croft proposed query expansion using multi-term concept. In this approach they used Markov Random Field model to select both multi-term and single-term concepts [26]. Dalton *et al.*, (2014) proposed a technique using entity names, aliases and categories and a number of methods of linking these entities to the query for query expansion [27].

In this paper we are trying to find the optimum value of documents and terms for query expansion using pseudo relevance feedback based on Bo1 and Bo2 model. This paper gives an empirical result based on these models which finds the optimum value that can be maximized.

III. MATERIALS AND METHODS

A number of query expansion methods have been developed to enhance the performance of the IR system. Rocchio algorithm [28] is one of the oldest method for query expansion using relevance feedback. It was developed using vector space model. This algorithm adjusts the weights of the documents. In this approach coordinates for the modified vector is adjusted either closer, or farther away, from the centroid of the document collection. The modified vector coordinates is being closer to the centroid of relevant documents if the weight is increased for relevant documents. Pseudo relevance feedback based on Bo1 [29] model is one of the stable DFR term weighting model. In Bo1 model the weight of the term "*t*" is given by

$$we(t) = tf_{.}\ \log_2 \frac{1+Q_n}{Q_n} + \log_2(1+Q_n) \qquad (1)$$

where *we*(*t*) is the weight of the term "t" and *tf* is the term frequency of the query in top-ranked documents. Q_N is equal to F/N, where *F* represents frequency of the query term in the collection and *N* represents the number of documents in the collection.

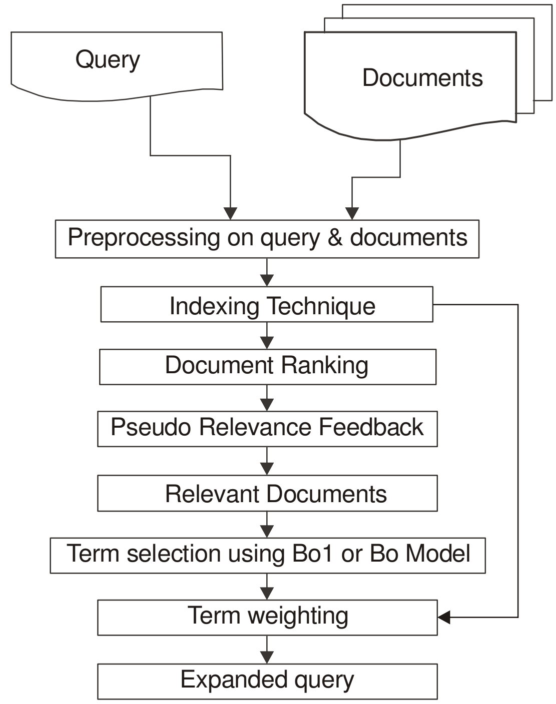

Fig. 1. Optimal Query Expansion Using Bo1 and Bo2 model.

While in Bo2 model [31] the weight of the term "*t* "is given by

$$we(t) = -\log_2 \frac{1}{1+\lambda} - f(t|top\_D).\log_2 \frac{\lambda}{1+\lambda} \qquad (2)$$

where $\lambda = Total\ Freq.(top\_D).\ \frac{Freq.(t|Collection)}{Tota\ lFreq.(Collection)}$

and top_*D* is the top ranked documents for the given query. The proposed method uses these two equations based on Bo1 and Bo2 model to find the optimum value of expansion documents and terms respectively.

Algorithm 1 Optimal QE using Bo1 and Bo2 model

1. Make a vector for user query as query vector.
2. Perform pre processing on the query and the document i.e. remove stop words.
3. Rank the documents containing query vector using Inverse Document Frequency model with Laplace and normalization of 2(InL2).
4. Apply Pseudo relevance feedback to extract top "*k*" ranked documents.
5. From Equation 1 and 2 use either Bo1 or Bo2 model to weight a term.
6. Expand the query by top "m" weighted terms.
7. Empirically observe the result and find optimum value of "k=d" and "m=optn".
8. Expand the query by adding "optn" number of terms using either Equation 1 or 2.
9. Return expanded query.

Optimal Query expansion using Bo1 and Bo2 model is shown in algorithm 1.

IV. RESULTS AND DISCUSSION

The performance of the retrieval system can be described by precision and recall. For a given collection of documents *C* precision and recall is given by

$$\text{Precision} = \frac{|A \cap B|}{|B|}$$

and $$\text{Recall} = \frac{|A \cap B|}{|A|}$$

where $A \subseteq C$ and $B \subseteq C$ are the collection of retrieved documents and is collection of relevant documents respectively. Whereas mean average precision(MAP) is given by

$$MAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{l_j} \sum_{k=1}^{l_j} \text{Precision}(R_{jk})$$

where (d_1, d_2, d_3,……. d_{l_j}) is the set of relevant documents for any given query $q_j \in Q$ and R_{jk} is the collection of ranked retrieval results from the top result. |Q| indicates total available queries. The experiment has been performed on Forum for Information Retrieval Evaluation (FIRE) collection [30] English dataset *using* Terrier 3.5 search engine. Terrier provides indexing, retrieval and evaluation on English and other language documents. The experiment has been performed on 50 queries. The topic file contains the following tag format

<topics> <top>
<num>26</num>
<title>Singur land dispute</title>
<desc>The land acquisition policies of the Left parties in Singur and the protest of Bhumi Ucched Protidrodh Committee against this policy.</desc>
<narr>Relevant documents should contain information regarding the acquisition of agricultural land for industrial growth in Singur, the territorial battle between the Left Parties an opposition parties ,the brutal killing of the innocent people and the protests and the criticism by people from different sections of society.</narr>
</top></topics>

In FIRE Collection Terrier processed the queries by Short queries (title only) and Long queries (title, description and narration). We have performed the experiment for Short and Long queries. In this experiment we have fixed the number of expansion terms at 10 and 15 and by varying number of top "k" documents at 3, 5, 8, 10, 50 and 100. The following tables show the performance of the system when query expansion is applied by using 10 terms and 15 terms on short and long queries. After analyzing the tables we observe that system performs well when expansion documents and expansion terms are 8 and 15 respectively in both Bo1 and Bo2 model. But Table 7 shows that system performs better when we processed the query using title field only i.e. on short query compared to the long query(title, description and narration). From Table 7 it is cleared that mean average precision for short queries is 0.5657 compared to long queries 0.5640 shown in Table 8. Bo2 model outperforms for short queries over long queries when we fix expansion documents and expansion terms to 8 and 15 respectively. But from Table 3 and 4 it is cleared that in case of Bo1 model long queries performs better than short queries when documents are 8 and expansion terms are 15. If we compare the result in Bo1 model and Bo2 model then we conclude that for short queries Bo2 model performs better than Bo1 model but for long queries Bo1 model perform better than Bo2 model in case(D=8 and T=15).If overall comparison is made then we find that Bo2 model better performs for short queries than Bo1 model for short queries and long queries.

Hence from the above discussion we can say that for this dataset optimum value of documents and expansion terms are 8 and 15 respectively for both Bo1 and Bo2 model.

Table 1: QE Using Bo1 on Short query when Expansion term = 10.

Performance of QE using Bo1 on Short Queries when Expansion Term=10						
	Exp.d=3	**Exp.d=5**	**Exp.d=8**	**Exp.d=10**	**Exp.d=50**	**Exp.d=100**
No. of Queries(N)	**50**	**50**	**50**	**50**	**50**	**50**
Retrieved	50,000	50,000	50,000	50,000	50,000	50,000
Relevant	3779	3779	3779	3779	3779	3779
Relevant Retrieved	3530	3536	3538	3535	3540	3536
MAP	0.5566	0.5549	0.5573	0.5537	0.5288	0.5180
R Precision	0.5429	0.5449	0.5425	0.5357	0.5124	0.5080
P@10	0.7300	0.7280	0.7120	0.7220	0.6880	0.6760
P@20	0.6790	0.6750	0.6740	0.6690	0.6520	0.6390

Table 2: QE Using Bo1 on Long query when Expansion term = 10.

Performance of QE using Bo1 on Long Queries when Expansion Term=10						
	Exp.d=3	**Exp.d=5**	**Exp.d=8**	**Exp.d=10**	**Exp.d=50**	**Exp.d=100**
No. of Queries(N)	**50**	**50**	**50**	**50**	**50**	**50**
Retrieved	50,000	50,000	50,000	50,000	50,000	50,000
Relevant	3779	3779	3779	3779	3779	3779
Relevant Retrieved	3560	3564	3579	3578	3584	3574
MAP	0.5333	0.5474	0.5557	0.5550	0.5326	0.5100
R Precision	0.5221	0.5310	0.5349	0.5355	0.5187	0.4991
P@10	0.7400	0.7420	0.7560	0.7520	0.7320	0.7100
P@20	0.6700	0.6910	0.7000	0.6880	0.6700	0.6480

Table 3: QE Using Bo1 on Short query when Expansion term = 15.

Performance of QE using Bo1 on Short Queries when Expansion Term=15						
	Exp.d=3	**Exp.d=5**	**Exp.d=8**	**Exp.d=10**	**Exp.d=50**	**Exp.d=100**
No. of Queries(N)	**50**	**50**	**50**	**50**	**50**	**50**
Retrieved	50,000	50,000	50,000	50,000	50,000	50,000
Relevant	3779	3779	3779	3779	3779	3779
Relevant Retrieved	3539	3542	3543	3547	3556	3544
MAP	0.5603	0.5638	0.5619	0.5586	0.5319	0.5195
R Precision	0.5455	0.5472	0.5442	0.5395	0.5154	0.5067
P@10	0.7240	0.7320	0.7200	0.7240	0.6820	0.6720
P@20	0.6850	0.6830	0.6800	0.6660	0.6510	0.6420

Table 4: QE Using Bo1 on Long query when Expansion term = 15.

Performance of QE using Bo1 on Long Queries when Expansion Term=15						
	Exp.d=3	**Exp.d=5**	**Exp.d=8**	**Exp.d=10**	**Exp.d=50**	**Exp.d=100**
No. of Queries(N)	**50**	**50**	**50**	**50**	**50**	**50**
Retrieved	50,000	50,000	50,000	50,000	50,000	50,000
Relevant	3779	3779	3779	3779	3779	3779
Relevant Retrieved	3560	3564	3579	3579	3586	3575
MAP	0.5335	0.5474	0.5558	0.5554	0.5330	0.5102
R Precision	0.5230	0.5309	0.5357	0.5354	0.5186	0.4989
P@10	0.7400	0.7420	0.7540	0.7520	0.7300	0.7120
P@20	0.6710	0.6910	0.6990	0.6880	0.6700	0.6480

Table 5: QE Using Bo2 on Short query when Expansion term = 10.

Performance of QE using Bo2 on Short Queries when Expansion Term=10						
	Exp.d=3	**Exp.d=5**	**Exp.d=8**	**Exp.d=10**	**Exp.d=50**	**Exp.d=100**
No. of Queries(N)	**50**	**50**	**50**	**50**	**50**	**50**
Retrieved	50,000	50,000	50,000	50,000	50,000	50,000
Relevant	3779	3779	3779	3779	3779	3779
Relevant Retrieved	3563	3567	3564	3557	3543	3520
MAP	0.5546	0.5497	0.5603	0.5594	0.5262	0.4925
R Precision	0.5407	0.5351	0.5467	0.5413	0.5128	0.4825
P@10	0.7200	0.7360	0.7360	0.7400	0.6940	0.6640
P@20	0.6690	0.6690	0.6710	0.6680	0.6510	0.6090

Table 6: QE Using Bo2 on Long query when Expansion term = 10.

Performance of QE using Bo2 on Long Queries when Expansion Term=10						
	Exp.d=3	**Exp.d=5**	**Exp.d=8**	**Exp.d=10**	**Exp.d=50**	**Exp.d=100**
No. of Queries(N)	**50**	**50**	**50**	**50**	**50**	**50**
Retrieved	50,000	50,000	50,000	50,000	50,000	50,000
Relevant	3779	3779	3779	3779	3779	3779
Relevant Retrieved	3583	3583	3593	3602	3576	3539
MAP	0.5420	0.5564	0.5632	0.5572	0.5251	0.4901
R Precision	0.5285	0.5382	0.5414	0.5394	0.5136	0.5811
P@10	0.7360	0.7400	0.7520	0.7360	0.7160	0.7000
P@20	0.6810	0.6940	0.6960	0.6900	0.6630	0.6390

Table 7: QE Using Bo2 on Short query when Expansion term = 15.

Performance of QE using Bo2 on Short Queries when Expansion Term=15						
	Exp.d=3	**Exp.d=5**	**Exp.d=8**	**Exp.d=10**	**Exp.d=50**	**Exp.d=100**
No. of Queries(N)	**50**	**50**	**50**	**50**	**50**	**50**
Retrieved	50,000	50,000	50,000	50,000	50,000	50,000
Relevant	3779	3779	3779	3779	3779	3779
Relevant Retrieved	3563	3571	3573	3569	3560	3524
MAP	0.5613	0.5594	0.5657	0.5644	0.5286	0.5020
R Precision	0.5463	0.5450	0.5502	0.5463	0.5073	0.4852
P@10	0.7280	0.7360	0.7320	0.7360	0.7180	0.6580
P@20	0.6710	0.6790	0.6700	0.6750	0.6500	0.6120

Table 8: QE Using Bo2 on Long query when Expansion term =15.

Performance of QE using Bo2 on Long Queries when Expansion Term=15						
	Exp.d=3	**Exp.d=5**	**Exp.d=8**	**Exp.d=10**	**Exp.d=50**	**Exp.d=100**
No. of Queries(N)	**50**	**50**	**50**	**50**	**50**	**50**
Retrieved	50,000	50,000	50,000	50,000	50,000	50,000
Relevant	3779	3779	3779	3779	3779	3779
Relevant Retrieved	3582	3583	3594	3602	3575	3540
MAP	0.5416	0.5571	0.5640	0.5579	0.5244	0.4906
R Precision	0.5280	0.5386	0.5412	0.5407	0.5138	0.4826
P@10	0.7360	0.7400	0.7540	0.7340	0.7180	0.7000
P@20	0.6810	0.6950	0.6970	0.6900	0.6610	0.6380

V. CONCLUSION

This paper explores query expansion techniques. We have tried to find the value of documents and terms respectively for which system performs well on this dataset. It is observed that Bo1 model and Bo2 model performs better when documents and expansion terms are 8 and 15 respectively. The mean average precision was 0.5657 in the case of Bo2 model when queries were processed using title field only and documents and terms were 8 and 15 respectively. Empirical results show that optimum number of query expansion documents and terms for both Bo1 and Bo2 model is 8 and 15 respectively for the dataset in study. However this value might be varied for other dataset.

VI. FUTURE SCOPE

In future we will try to implement query expansion technique using various machine learning techniques.

ACKNOWLEDGEMENTS

The authors sincerely thank FIRE for providing test collections relevance judgment files. The authors would also like to thank the Terrier team for the Terrier Retrieval Engine which was used for indexing, retrieval and evaluation of the experiments on English language. One of author is pursing Doctoral program from Department of Mathematics and Computer Applications Maulana Azad National Institute of Technology, Bhopal India as Full time research Scholar. He expresses his sincere gratitude toward the institute for giving opportunity and required facility to pursue his Doctoral program from the institute.

Conflict of Interest: Nil.

REFERENCES

[1]. Turtle, H. R., & Croft, W. B. (1992). A comparison of text retrieval models. *The computer journal, 35*(3), 279-290.

[2]. Ooi, J., Ma, X., Qin, H., & Liew, S. C. (2015). A survey of query expansion, query suggestion and query refinement techniques. In *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)* (pp. 112-117). IEEE.

[3]. Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, *44*(1), 1-50.

[4]. Spink, A., Wolfram, D., Jansen, M. B., & Saracevic, T. (2001). Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, *52*(3), 226-234.

[5]. Robertson, S. E., & Jones, K. S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information science*, *27*(3), 129-146.

[6]. Qiu, Y., & Frei, H. P. (1993). Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 160-169).

[7]. Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *SIGIR'94* (pp. 61-69). Springer, London.

[8]. Kassim, J. M., & Rahmany, M. (2009). Introduction to semantic search engine. In *2009 International Conference on Electrical Engineering and Informatics*, *2*, 380-386. IEEE.

[9]. Punyakanok, V., Roth, D., & Yih, W. T. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, *34*(2), 257-287.

[10]. Yokoyama, A., & Klyuev, V. (2010). Search Engine Query Expansion Using Japanese WordNet. In *2010 3rd International Conference on Human-Centric Computing* (pp. 1-5). IEEE.

[11]. Fensel, D., Van Harmelen, F., Horrocks, I., McGuinness, D. L., & Patel-Schneider, P. F. (2001). OIL: An ontology infrastructure for the semantic web. *IEEE intelligent systems*, *16*(2), 38-45.

[12]. Bhogal, J., MacFarlane, A., & Smith, P. (2007). A review of ontology based query expansion. *Information processing & management*, *43*(4), 866-886.

[13]. Alqadah, F., & Bhatnagar, R. (2011). Similarity measures in formal concept analysis. *Annals of Mathematics and Artificial Intelligence*, *61*(3), 245-256.

[14]. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391-407.

[15]. Hofmann, T. (2013). Probabilistic latent semantic analysis, 289-296.

[16]. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*, 993-1022.

[17]. Sergienko, R., Gasanova, T., Semenkin, E., & Minker, W. (2016). Collectives of Term Weighting Methods for Natural Language Call Routing. In *Informatics in Control, Automation and Robotics* (pp. 99-110). Springer, Cham.

[18]. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111-3119.

[19]. Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532-1543.

[20]. Lafferty, J., & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 111-119).

[21]. Attar, R., & Fraenkel, A. S. (1977). Local feedback in full-text retrieval systems. *Journal of the ACM (JACM)*, *24*(3), 397-417.

[22]. Bai, J., Song, D., Bruza, P., Nie, J. Y., & Cao, G. (2005, October). Query expansion using term relationships in language models for information retrieval. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, 688-695.

[23]. Shaalan, K., Al-Sheikh, S., & Oroumchian, F. (2012). Query expansion based-on similarity of terms for improving Arabic information retrieval. In *International Conference on Intelligent Information Processing*, 167-176. Springer, Berlin, Heidelberg.

[24]. Bai, J., Nie, J. Y., Cao, G., & Bouchard, H. (2007). Using query contexts in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval,*15-22.

[25]. Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, *44*(1), 1-50.

[26]. Metzler, D., & Croft, W. B. (2007). Latent concept expansion using markov random fields. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval,* 311-318.

[27]. Dalton, J., Dietz, L., & Allan, J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 365-374.

[28]. Rocchio, J. (1971). Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, 313-323.

[29]. Lu, S., He, B., & Xu, J. (2013). Hyper-geometric model for information retrieval revisited. In *Asia Information Retrieval Symposium*, 62-73. Springer, Berlin, Heidelberg.

[30]. McCreadie, R., Macdonald, C., Ounis, I., Peng, J., & Santos, R. L. (2009). *University of glasgow at trec 2009: Experiments with terrier*. GLASGOW UNIV (UNITED KINGDOM).

[31]. Amati, G. (2003). *Probabilistic Models for Information Retrieval based on Divergence from Randomness. University of Glasgow, UK* (Doctoral dissertation, Ph.D. Thesis).

How to cite this article: Shukla, Abhishek Kumar and Das, Sujoy (2020). Bose Einstein 1 and Bose Einstein 2 Model for Optimal Query Expansion. *International Journal on Emerging Technologies*, *11*(1): 280–285.