Project Report

On

Query Expansion and Pseudo-Relevance feedback using Firefly Algorithm

Submitted by

15IT141 Shamitha S Udupa
15IT239 Sanjana B

Under the Guidance of

Dr Sowmya Kamath S

Dept. of Information Technology,

NITK, Surathkal

Date of Submission: May 4, 2018

Dept. of Information Technology
National Institute of Technology Karnataka, Surathkal.
2017-2018

Department of Information Technology NITK Surathkal

End Semester Evaluation(May 2018)

Course Code : IT362 **Course Title :** Information Retrieval **Project Title :** Query Expansion and Pseudo-Relevance feedback using Firefly Algorithm

Project Group:

Name of the student	Register Number	Signature with date
Shamitha S Udupa	15IT141	
Sanjana B	15IT239	

Place:

Date: *(Name and Signature of Mini Project Guide)*

Abstract

The hardship in finding keywords used in the search queries which are incomplete or not precise has resulted in search engines failing to retrieve the relevant information. A promising technique to escape this hurdle and improvise the search engine's performance is Query Expansion, wherein the original query of user is altered by adding new terms that characterize the user's information needs in the best way and produce a better query. To enhance the effectiveness of retrieval of query expansion along with low calculation cost,an evolutionary algorithm based on Firefly behaviour is used. In contrast to the traditional methods, the Firefly Algorithm finds the best expanded query from among fireflies representing expanded query solutions instead of selecting the expansion terms which are best. In this approach the length of the expanded query is determined using experimental evaluation. Experiments have been performed on the MEDLINE dataset, the online medical database.The results indicate that the firefly approach is more effecient when compared to existing architectures.

Declaration

We hereby declare that the project entitled ***"Query Expansion and Pseudo-Relevance feedback using Firefly Algorithm"*** was carried out by us during the even semester of the academic year 2017 – 2018. We declare that this is our own work and it has been completed successfully according to the direction of our guide Prof Dr Sowmya Kamath S and as per the specifications of Department of Information Technology, NITK Surathkal.

Place: Surathkal,Nitk
Date: May 4, 2018

Signature of Shamitha S Udupa

Signature of Sanjana B

Acknowledgement

We would like to express our gratitude to our mini project guide Prof Dr Sowmya Kamath S whose scholarly guidance and suggestions have helped us to carry out the present research work. We thank the Department of Information Technology for granting us with necesarry department server access to carry out the experimental evaluation of our project.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

In recent times,the amount of available information online has been growing tremendously. Amount of user generated media is increasing tremendously on social media platforms. The unprecedented growth of information has led to addition of new words, first occurences of names,acronyms,abbreviations,emoticons and so on.[1] have indicated that most of the query keywords are not in general vocabulary,or are abbreviated speeches(eg: BRB),proper names or wrong spellings or foreign words.

The hardship of finding these new imprecise,ambiguous words results in poor retrieval of information in search engines. Query expansion is a promising technique to overcome this problem. In QE,query is improvised by adding new key words to it that capture the user's information need better or produce a more useful query.

Currently, QE is a very promising technique to increase the effectiveness of retrieval. In recent times several new query techniques have been proposed nonetheless the results from these works are not significantly different from state-of-art techniques.

The main reason for the above drawback is that these techniques rely on traditional methods which search for best suited query terms and not best expanded query.We can try to tackle this problem by thinking of selection of the best expanded query as an optimisation problem.

Complete optimisation problems are np-hard and no polynomial time algorithms exist. For practical purposes, these methods might not be optimal. Thus approximate solutions to solve combinatorial optimisation problems have received more attention.Evolutionary algorithms are such algorithms which might give up the guarantee of finding the best solution but always give good solutions with less computation.

Firefly algorithms in 2008 [2],[3] is based on the phenomenon of bioluminescent communication and the flashing behavior of fireflies. An original approach for finding pseudo-relevance and query expansion based on Firefly algorithm is proposed. We evaluate this approach using Medline Dataset and query expansion techniques, rocchio[4] and Robertson/Spark Jones[5] technique are used as the baseline for comparison.

2 Literature Review

2.1 Background

The increase in size of new queries such as first occurences from proper names, abbreviations, misspelling words and use of ambiguous and imprecise words to describe the information need have caused failure of information retrieval matching the corresponding need.Several methods such as including search result clustering, concept based lattice IR and contextual document ranking modeled as basis vectors have been proposed to overcome this. Expansion of the user's query by appending extra relevant keywords is still a promising technique to enhance the effectiveness of the IR system's retrieval.

2.2 Outcome of Literature Review

One well-known approach is Pseudo-Relevance Feedback(PRF).Pseudo-relevant documents i.e. the initial set of documents retrieved w.r.t original query are used to choose the terms that can be used as expansion keywords.[6] As shown in the Figure 1 the pseudo-relevance feedback performs a search for the original query. The documents are scored and ranked using Okapi BM25 [7]. The top-ranked documents are taken as pseudo-relevant documents. The vocabulary of the pseudo relevant documents are ranked using Robertson/Sparck Jones or Rocchio's to get best expansion keywords. Finally, the expansion query is got by adding the top-ranked keywords to original query. Pseudo-Relevance Feedback provides a more precise description of the query but it doesnot look for a suitable expanded query. Further, [4] proposes a Rocchio's model based on proximity for pseudo relevance feedback. [8] attempts to increase PRF by applying a supervised learning method for choosing expansion terms in order to segragate good expansion terms from the rest. They have used the C-SVM to select the best expansion keywords.

Further,in this method[9], overlapping clusters are formed to find the important documents for the original query's retrieval and repeatedly uses these documents to highlight the important topics of the query.

Particle swarm optimization algorithm can be used to improve PRF by choosing an appropriate combination of term weights in the query vector and a fitness function to measure the proximity between the re-weighted query terms and top ranked documents retrieved by original query. This method although popular,the results are similar to those of the existing techniques.

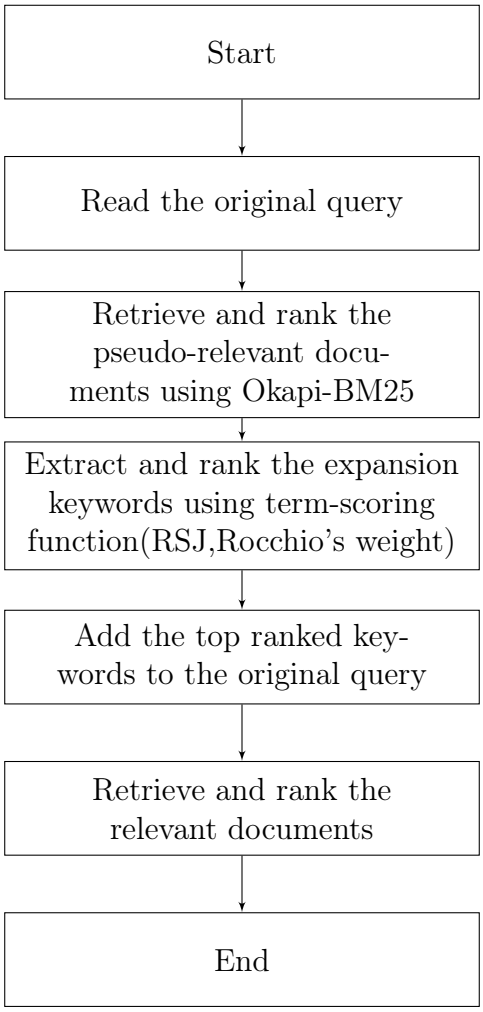

Figure 1: Pseudo-Relevance Feedback

2.3 Problem Statement

"To increase the retrieval effectiveness of pseudo-relevance feedback using firefly algorithm to find the best expanded query while maintaining low cost."

In this project, query expansion is looked from the prespective of expanded query rather than expansion keywords. Swarm optimisation based firefly algorithm is chosen to select the best expanded query from firefly query candidates. An Information Retrieval system is built on the expanded query and documents are ranked by Okapi BM25 model.

2.4 Objectives

1. To construct the inverted index for MEDLINE dataset

2. An Information Retrieval system for effective retrieval of documents using Okapi BM25 model

3. Implementation of Firefly algorithm to find the expanded query

4. Evaluation of the retrieval effectiveness using different measures like Precision and Mean Average Precision(MAP)

5. Comparison of our system with the PRF based on Rochhio's weight and Robertson/Sparck Jones weight to find best expansion keyword

3 Methodology

We will now discuss Firefly algorithm based approach. As shown in Figure 2 the procedure consists of getting the pseudo-relevant documents and respective terms to build p solution and executing the algorithm to derive the expanded query.

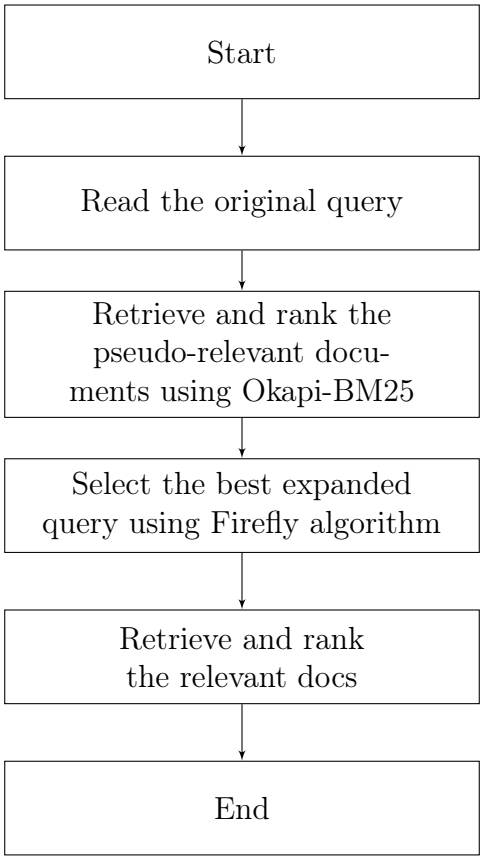

Figure 2: The proposed Firefly Algorithm for prf

3.1 Solutions' representation and initialisation of population

Each solution is represented by a firefly with the search space consisting of all of fireflies generated.Although a firefly in our problem would be the expanded query, we only consider the additional terms for optimisation concern.

$$\tilde{Q} =< \tilde{q_1}, \tilde{q_2}, ..., q_{|}\tilde{Q}_{|} > \tag{1}$$

$$EQ = Q + \tilde{Q} \tag{2}$$

Equation 1 is a vector representation of the terms that can be appended to the original query. The overall expanded query is given by Equation 2.

The search space has all potential fireflies vocabulary.The size of the search space consists of all possible combinations from vocabulary of pseudo relevant documents with $|\tilde{Q}|$(length of vector $\tilde{Q}$). Thus the solution space order is of the binomial of the coefficient of the above mentioned vocabulary and $|\tilde{Q}|$. This huge number describes the numerous possibilities that can be exploited to build a query expansion. Regarding the initialisation of population N fireflies are chosen randomly by selecting $|\tilde{Q}|$ terms from the vocabulary of pseudo relevant documents chosen. The size of pseudo-relevant documents considered has to be determined emperically.

3.2 Fitness Function

The objective function evaluates the solution's quality. Solution means an expanded query,therefore its performance can be assessed by considering the inverted indexes of its terms and then calculating the each document's score such that it belongs to R(pseudo relevant document corpus). Equation 3 calculated by pseudo-code in Algorithm 1 gives the fitness function score for every firefly.

$$f(EQ) = max(score_{bm25}(d1, EQ), score_{bm25}(d2, EQ), ...score_{bm25}(d_{\|R\|})) \tag{3}$$

Algorithm 1 initially generates the inverted index of every expansion term q̃i belongs

Algorithm 1 Fitness function pseudocode

```
1: procedure FITNESS FUNCTION F(EQ)
2:     Inverted index In(q̃_i)
3:     Initialise the vector of scores S ← [ ]
4:     for each keyword q̃i in Q̃ : do
5:         for each document d in R do
6:             if d E In(q̃i) and S[ d ] != 0 then
7:                 S[d] ← scorebm25(Q) + scorebm25(Q̃)
               end if
           end for
       end for
8:     f(EQ) ← max(S)
```

to $\tilde{Q}$ then iteratively calculates each document's score included in both In and R with respect to the expanded query EQ. The okapi score of $\tilde{Q}$ is calculated only as the score of Q will already have been calculated.Once the scores have been evaluated,max(S) is

the expanded query EQ's fitness value. As the QE issue is formulated as maximisation problem in Algorithm 1,the fitness function expresses the light intensity.

3.3 Update locations of fireflies

$$\beta = 1 \div (1 + \gamma * r_{ij}) \tag{4}$$

The solution differs across the iterations with its fitness function score increasing.The global is compared with the best solution of each iteration and is assigned the local best solution if it has a higher fitness value. There are two parts to movement of the fireflies. The first part is determined by Equation 4 where :
β belongs to [0,1], indicates the attractiveness of firefly j;
γ indicates the light absorption coefficient;
r_{ij} indicates the distance or how far firefly i,j are
The distance is calculated using Hamming distance shown in Equation 5.The Hamming distance is the number of elements which do not correspond with each other in the sequence between two fireflies.

$$\tilde{W}_1 =< \tilde{w_5}, \tilde{w_4}, \tilde{w_3} > \tilde{W}_2 =< \tilde{w_5}, \tilde{w_2}, \tilde{w_1} > \text{Then hamming distance is 2} \tag{5}$$

$$\tilde{Q}^{t+1} = \begin{cases} \tilde{Q}^t_{ik}, & \text{rand} > \beta \\ \tilde{Q}^t_{jl}, & \text{otherwise} \end{cases} \tag{6}$$

Equation 6 gets the firefly $\tilde{Q}$i nearer to a more suitable and attractive solution $\tilde{Q}$j. Next the distance between the two fireflies decreases and the attractiveness value is increased.Here

$\tilde{Q}^t_{ik}$ doesnot belong to $\tilde{Q}_j$, is a term of $\tilde{Q}_i$
$\tilde{Q}^t_{jl}$ doesnot belong to $\tilde{Q}_i$ is a random term of $\tilde{Q}_j$

Next Equation 7 is calculated.The randomization parameter α prevents the trapping of solution at local optima.The value is slowly decremented to prevent the algorithm's premature convergance.

$$\tilde{Q}^{t+1} = \begin{cases} \tilde{Q}^t_{ik}, & \text{rand} > \alpha \\ \tilde{q}_l, & \text{otherwise} \end{cases} \tag{7}$$

Based on the α's value,$\tilde{Q}_i$'position is updated by adjusting each of its element $\tilde{Q}_{ik}$ with $\tilde{q}_l$, a randomly selected keyword from the vocabulary of the pseudo relevant documents.

3.4 Firefly algorithm algorithm pseudocode

In Algorithm 2,the first step is to set the initial values of the γ, absorption coefficient and α, the randomisation parameter .The starting N populations are then created randomly from the search space.Each firefly $\tilde{Q}$ is initialised by selecting $|\tilde{Q}|$ terms from vocabulary of pseudo relevant documents. The intensity of light of each firefly is calculated by its fitness function value.The local best solution is the best solution of a given iteration.The global best solution will be the solution with the highest fitness value for all iterations prior to given iteration. Every iteration leads to the global best solution being updated if required. As shown in the Equation 6 and 7, fireflies move towards a more attractive firefly and provide increasingly better solutions.The termination conditions are that the optimised solution doesnot change for a pre-specified duration of iterations or maximum iteration limit is met.Finally when termination conditions are met,the terms from the global best solution are augmented to the Q to get the expanded query EQ.

Algorithm 2 Firefly algorithm pseudocode

```
1:  procedure FIREFLY ALGORITHM
2:      Fitness function f(Q̃)
3:      Define light absorption coefficientγ ,parameterα
4:      Light intensity I_i at Q_i is determined by f(Q̃)
5:      Generate an initial population of N fireflies
6:      while t < Maxiter and bestExpandedQuery still changes do
7:          for  i in 1 to N do
8:              for i in 1 to N do
9:                  if f (Q̃i) < f (Q̃j) then
10:                     Move query firefly Q̃i to Q̃j
                    end if
11:                 vary attractiveness
12:                 Evaluate new solutions
                end for
            end for
13:         Rank the fireflies and find the current best firefly
        end while
```

4 Implementation

4.1 Work Done

4.1.1 Dataset

In order to test our IR system, we have used MEDLINE dataset. It is the online medical database accessed via the PubMed interface.It contains articles with title, abstract, author names, date of publication, MeSH(Medical Subject Headings).

4.1.2 Preprocessing

We have extracted the document title, abstract and MeSH terms and preprocessed them with NLP operations like tokenizing,punctuation removal, stemming and stopword removal. The preprocessed document is used to construct an inverted index containing the index terms and postings list.

4.1.3 Information Retrieval System

We have created an Information Retrieval system using Okapi BM25 model to generate rankings to the query. The similarity score between the query and documents is calculated using the formula -

$$BM25(Q, d_j) = \sum_{t_i \epsilon Q} log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right) * \left(\frac{tf}{tf + k\Big((1-b) + b\frac{dl}{avgdl}\Big)}\right) \tag{8}$$

where Q is the query, d_j refers to the jth document in the corpus, N(total documents in the collection), n_i (the document frequency of the term t_i), tf is the term frequency of the term t_i, dl and avgdl refers to the document length of d_j and average length of the documents in the corpus respectively, k and b are constants such that 1.2<k<2 and 0.5<b<0.8 .

4.1.4 Firefly Algorithm

Firefly algorithm is implemented to find the query terms to give best expanded query.

Figure 3: Firefly algorithm execution

4.1.5 Performance Evaluation

We have used precision and Mean Average Precision(MAP) evaluation metrics to measure the retrieval effectiveness of our IR system. The results are discussed in the next section.

4.2 Result and Analysis

4.2.1 Fixing firefly algorithm parameters

In the firefly algorithm we need to emperically find the following parameters - $|\tilde{Q}|$, N and T (the query length, population size and the number of iterations). In these primary experiments, these parameters are varied while keeping others constant.The size of |R|(pseudo-relevant documents) are set as [10,30,50].The parameters α_0,γ and θ, are tuned at 1,1 and 0.95.

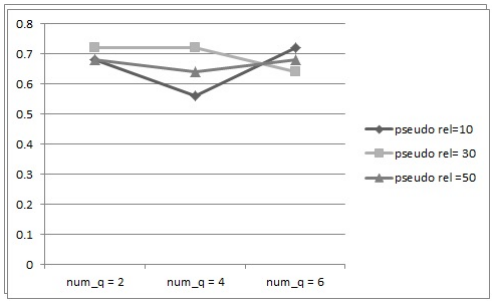

Figure 4: P@5 vs varying query length

Fig 4 and Fig 5 give the P@5 and MAP@10 results after varying $|\tilde{Q}|$ wrt number of query terms [2,4,6]. N and T are kept constant at value 5 and 10 respectively. From the graph we can see that $|\tilde{Q}| = 2 or 4$ gives better results. Fig 6 and Fig 7 give the P@5

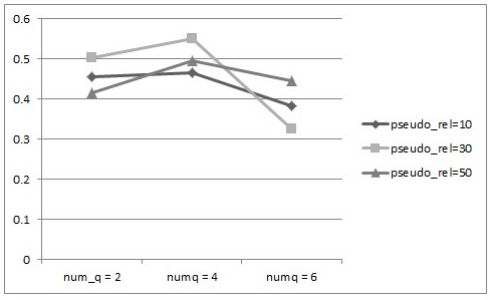

Figure 5: MAP@10 vs varying query length

and MAP@10 results for varying population length,N = [5,10,15,20,25]. $|\tilde{Q}|$ and T have values 4 and 10 respectively. Fig 8 and Fig 9 give the P@5 and MAP@10 results for varying iterations T =[10,20,30,40,50] while $|\tilde{Q}| = 4$ and N = 10

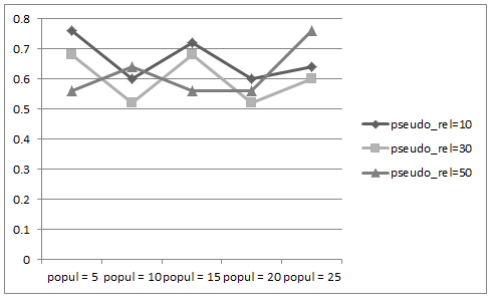

Figure 6: P@5 vs population size

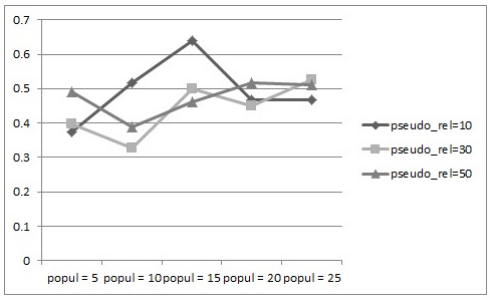

Figure 7: MAP@10 vs varying population size

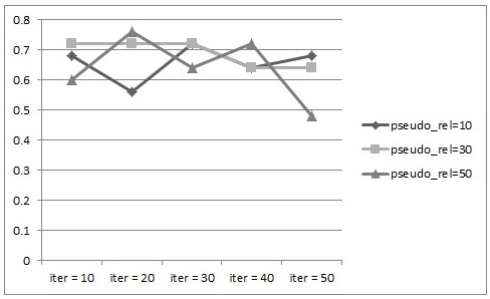

Figure 8: P@5 vs number of iterations

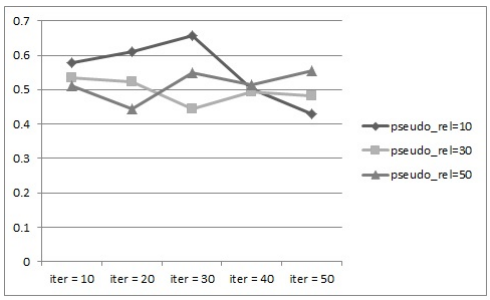

Figure 9: MAP@10 vs number of iterations

4.3 Comparison of Firefly algorithm with Rocchio and RSJ

Finally we compare the P@5 results of our firefly algorithm with that of Rocchio and RSJ algorithms. N = 10,$|\tilde{Q}|$ = 2, T = 30. The number of pseudo-relevant docu-

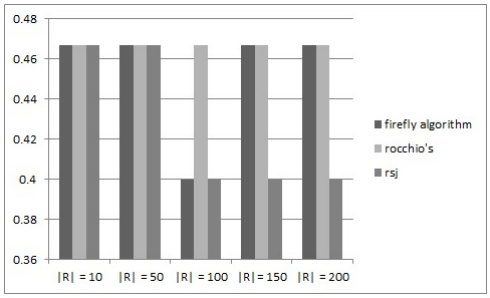

Figure 10: P@5 vs number of iterations

ments=[10,50,100,150,200]. Fig 10 gives the result of above computation.We can see that rocchio and firefly algorithms have almost similar performance and firefly algorithm performs better than rsj algorithm.

4.4 Innovative Work Done

We have developed a web interface which suggests the additional query words that can be added to the improvise the semantic meaning of the query. These additional terms are computed by the firefly algorithm. We have also refined the original query by removing the terms with low values of inverse document frequency(IDF) factor and replacing them with the terms of the global best firefly. In Figure 11, the terms in red color are the terms

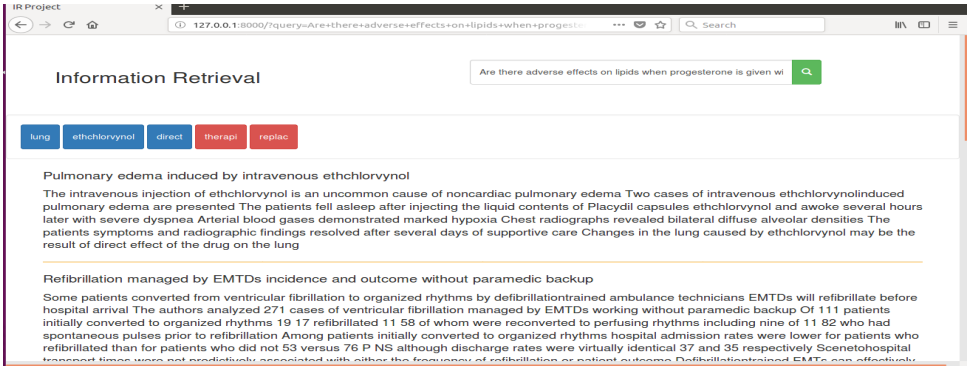

Figure 11: Web Interface for the firefly algorithm

of the original query which have low idf frequency and can be replaced with the terms colored blue calculated by firefly algorithm.

4.5 Individual Work Distribution

5 Future Work

We will try to improvise the firefly algorithm by designing a parallel variant of the same to increase both efficiency and effectiveness of our IR system. Further we will try to refine the query using automatically generated thesaurus and synsets from WordNet.

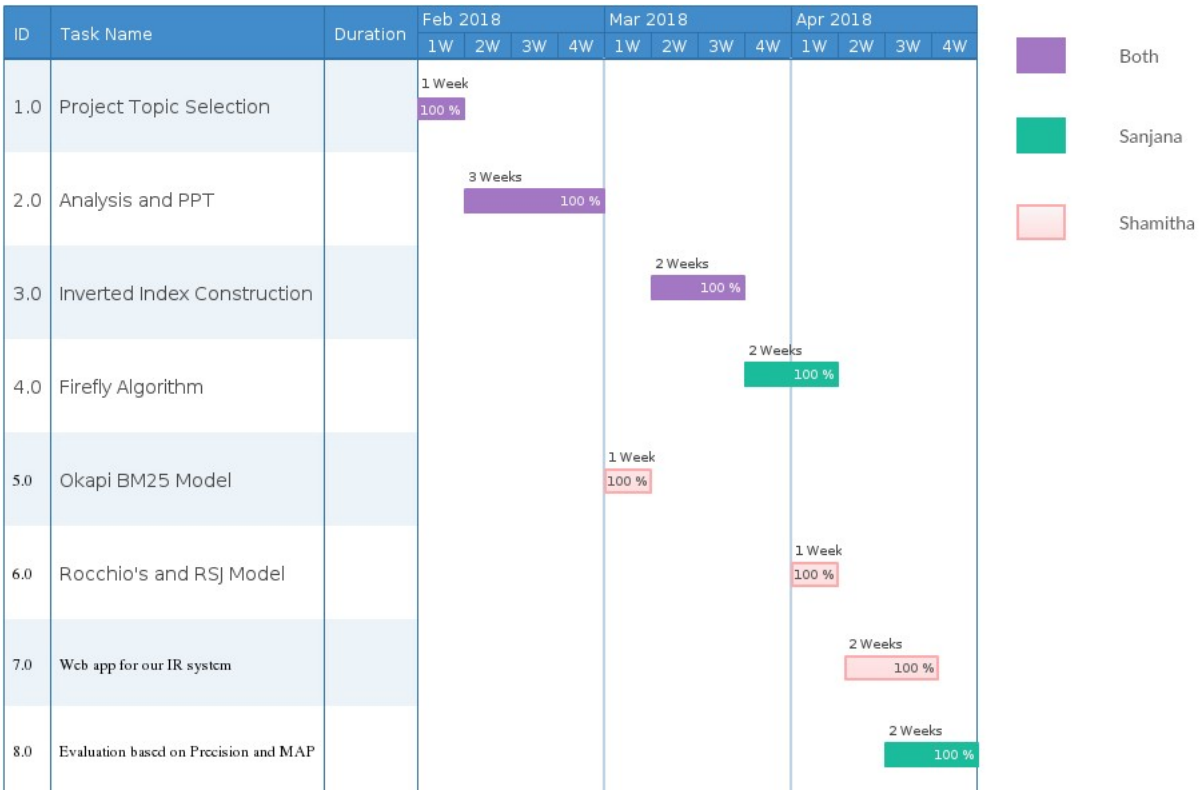

Figure 12: Gantt Chart

References

[1] Q. Chen, M. Li, and M. Zhou, "Improving query spelling correction using web search results." pp. 181–189, 01 2007.

[2] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 07 2010.

[3] X. S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, O. Watanabe and T. Zeugmann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 169–178.

[4] J. Miao, J. X. Huang, and Z. Ye, "Proximity-based rocchio's model for pseudo relevance," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '12. New York, NY, USA: ACM, 2012, pp. 535–544. [Online]. Available: http://doi.acm.org/10.1145/2348283.2348356

[5] S. Robertson and S. K Jones, "Relevance weighting of search terms," vol. 27, pp. 129–146, 05 1976.

[6] C. Carpineto and G. Romano, "A survey of automatic query expansion in information retrieval," *ACM Comput. Surv.*, vol. 44, no. 1, pp. 1:1–1:50, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2071389.2071390

[7] S. Robertson, S. Walker, M. Beaulieu, M. Gatford, and A. Payne, "Okapi at trec-4," in *In Proceedings of the 4th Text REtrieval Conference (TREC-4*, 1996, pp. 73–96.

[8] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson, "Selecting good expansion terms for pseudo-relevance feedback," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, ACM. ACM, 2008, p. 243–250.

[9] K. S. Lee and W. B. Croft, "A deterministic resampling method using overlapping document clusters for pseudo-relevance feedback," *Information Processing & Management*, vol. 49, no. 4, pp. 792 – 806, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030645731300006X