Indian Statistical Institute, Kolkata

M. Tech. (Computer Science) Dissertation

# Automated Text Illustration Problem

A dissertation submitted in partial fulfillment of the requirements
for the award of Master of Technology
in
Computer Science

Author:
Swarnendu Chakraborty
Roll No: CS-1421

Supervisor:
Dr. Mandar Mitra
CVPR Unit, ISI

**M.Tech(CS) DISSERTATION THESIS COMPLETION CERTIFICATE**
**Student: Swarnendu Chakraborty (CS1421)**
**Topic: Automated Text Illustration Problem**
**Supervisor: Dr. Mandar Mitra**

This is to certify that the thesis titled "***Automated Text Illustration Problem***" submitted by **Swarnendu Chakraborty** in partial fulfillment for the award of the degree of Master of Technology is a bonafide record of work carried out by him under my supervision. The thesis has fulfilled all the requirements as per the regulations of this Institute and, in my opinion, has reached the standard needed for submission. The results contained in this thesis have not been submitted to any other university for the award of any degree or diploma.

Date: Mandar Mitra

# Dedication

**To my parents and my well wishers, without their help and encouragement it would not have been possible.**

# Acknowledgements

I would like to thank my dissertation supervisor Dr. Mandar Mitra for agreeing to guide me and for helping me to undertake work in the topic.

I would also like to thank Dr. Debasis Ganguly, currently post doctoral fellow of Dublin City University of Dublin, Ireland, who helped me as a co-guide of my dissertation thesis.Without his help, this work wouldn't get a shape.

Last but not the least, I am grateful to all my lab mates and seniors of information retrieval lab of CVPR Unit Indian Statistical Institute, Kolkata for helping me though out the project with their valuable suggestions.

# Abstract

Image and text are the two different ways of communication. The readability and comprehensibility of a large volume of text can be increased vastly using a sequence of images. It is a very important research question that how efficiently a query should be formulated from a segment of text to retrieve relevant images from a data set to illustrate the text. In this project, we proposed a number of system to counter this problem which can illustrate a text by most appropriate images. To construct a system like this, we used ImageCLEF 2010 and Wikipedia 2016 data set. In the first phase of this work a set of children stories has been illustrated by the images of ImageCLEF 2010 data set. In the second phase of this work, Wikipedia 2016 data dump was used to investigate how our query formulation method works in a vast amount of data set like Wikipedia. An image data set was made from this data dump and the textual information of Wikipedia page has been used as query. Some of the research challenges in this project was to develop an automated text illustrating system including techniques to automatically extract out the concepts to be illustrated from a full text page, explore how to use these extracted concepts for query representation in order to retrieve a ranked list of images per query and finally investigating how merge the ranked lists obtained from each individual concept to present a single ranked list of candidate relevant images per text page. In this work for query formulation segmentation of text, relevance feedback method, POS tag based technique has been used. It has been found from the subsequent experiments that in the ImageCLEF 2010 data set, the query formulation and expansion technique based on relevance feedback method performs better than all other approaches. On other side in Wikipedia 2016 data set, POS tag based method outperforms all other query formulation technique mainly because in this approach only noun phrases are used to formulate query. So instead of a verbose query, this method gives a concise and crisp query yet appropriate to describe the core content of a large text. In this method also the detailed performance analyses of various system has been reported with different standard metrics of information retrieval field.

# Contents

# Chapter 1

# Introduction

From the long past the medium of information is of textual form. Most of the data generated worldwide by systems today is in the form of raw text. This text comes as structured, semi-structured or unstructured data. The information hides in this huge amount of text. A very important research question is that how fast the information which is embedded in the text can be extracted or retrieved. One of the obvious way is to illustrate the text using images. An image is always more self-explaining than its textual form. So if a text can be illustrated by a sequence of images, then the information can be readable in a very fast way.

## 1.1 Background and Motivation

The problem of text illustration by image has been a very popular research field. A particular approach to solve the aforesaid problem is to build a system which can take the textual information and gives a sequence of images as output which can illustrate the textual information best in real time. In this study, we consider that Given a database $I$ of images and a text document or passage $T$, retrieve a ranked list of images $i_1, i_2, ..., i_k \in I$ that are appropriate as illustration for $T$.

In the proceedings of "*Forum for Information Retrieval Evaluation '2015*" a shared task has been proposed, named "*Automated Story Illustration Problem*" [2, 10, 16] which was the main motivation of this work. In the shared task, ImageCLEF 2010 [5] data set has been given. This data set was consist of meta data of 237,434 different images with image name, image description, image caption etc and a set of 27 short child stories taken from Aesop Fable have been given as query. The objective of the task was to illustrate the child stories by a sequence of images querying the image data set and retrieving a ranked list of relevant images. Later this problem has been extended to the general problem of text illustration on the context of more versatile data source using Wikipedia 2016 data dump [20].

## 1.2 Our Work

The work has been aimed at how neatly query can be formulated from a given text to illustrate as well as how the image data set can be constructed crisply and efficiently from the huge data dump. To formulate the query a number of methods has been tried also with some heuristics. Note that no image level features have been used to retrieve the relevant images, only textual meta data of the image i.e. image name, caption, description are used. This project also shows the detailed result about the retrieval and illustration performance in the subsequent sections. The contributions of this dissertation are:

- Creation of a significantly larger and more heterogeneous data set for text illustration by images from Wikipedia 2016 data dump [20].

- Comparison of various techniques for formulating or extracting keywords for the queries from a text that is to be illustrated.

- Comparison of standard retrieval model (BM25, LM, TF-IDF) for both "Automated Story Illustration" task of FIRE 2015 [2] and newly created Wikipedia data set.

# Chapter 2

# Related Work

As discussed in chapter 1, a very obvious way to increase the readability of a text segment can be done by converting the text to the sequence of images. It is no surprise that significant research effort has been put to develop such a system. The problem of "*Automated Story Illustration*" has been proposed as a shared task in "*Forum for Information Retrieval Evaluation ' 2015 (FIRE'15)*" conference. There, a number of techniques have been proposed for this problem.

In the overview paper [2] of this shared task in FIRE'15, two techniques were proposed. Firstly the whole text portion of the short story has been used as query to retrieved the relevant images. Another technique of query formulation has been introduced in this paper. The terms of the story text has been weighted by the tf-idf [12, 1, 8] score and then the terms have been used to retrieve images. A single query has been fired for every story.

A paper named "*Automated Story Illustration Using Word Embedding*" [10] has been published in *FIRE'16*. In this paper word2vec feature vector for each document has been used to expand the document as well as the query. In this method a new passage has been formed for every story with story text, story entities and story events. Now this new passage has been extended with Word Sense Disambiguation technique as well as using hypernyms of WordNet. On the other hand, from ImageCLEF 2010 [5] data set a retrieval has been perform by using the passage with story text, entity and events and relevant document has be retrieved. With the retrieved document a local image data set has been made. Now this database is used as a training data to train a model using Gensim. When the model is ready, the expanded query has been fired and relevant images weighted by tf-idf score for TREC evaluation has been generated. Most of the research paper user Apache Lucene [9] for the purpose of indexing.The paper [16] named "Automated Story Illustrator" used "Terriertool" as an indexing tool. For retrieval purpose BM25 and DFR relevance scoring function has been used.

The above related work [2, 10, 16] mainly focused on the state of the art technique of information retrieval field. A substantial amount of research has been done to illustrate text by the image using natural language processing technique. In a paper named “*Image illustration of Text using Natural Language Processing*” [21] a POS tagged ans machine learning based technique has been used in the first place.In the second phase query formulation and state of the art retrieval technique has been used. In some web based work, Goggle search engine has been used to online retrieval of images given a text segment.

# Chapter 3

# Our Work

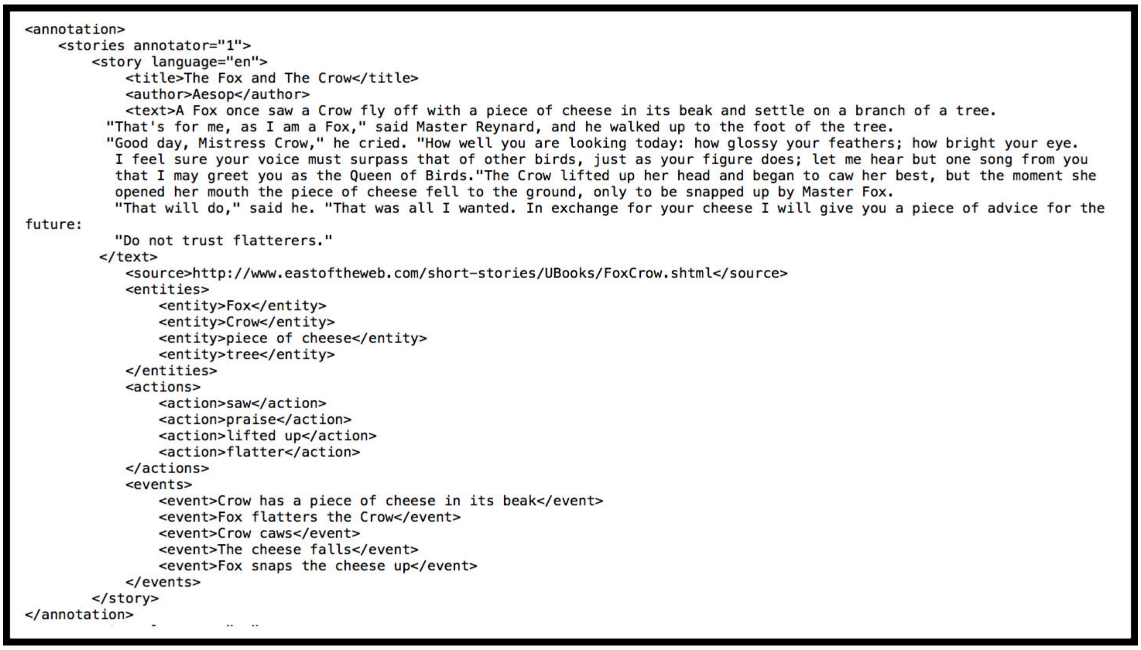
The field of information retrieval [8] being an area of research, the necessity of a standard test collection cannot be overemphasized. To build a test collection we first need at least a moderately large document collection. The document collection turns into a test collection once it is supplanted with a set of search queries and relevance judgments. As discussed in chapter 1, the main motivation of our work has come from a problem named "*Automated Story Illustration*" [2, 16, 10] which was offered as a shared task of the conference "*Forum For Information Retrieval Evaluation (FIRE)*". The main objective of this shared task was as follows given a short children's story, a system needs to retrieve images that are appropriate as illustration for the textual story. The organiser of this shared task has provided the necessary data set, short child story which needs to be illustrated in the form of XML file and relevance judgement file used by the TREC evaluation software.

## 3.1 Collection Overview

We first discuss a little bit about the data set used for this experiment. The static image collection that we use for this task is the ImageCLEF 2010 [5] data set. This collection consists of total 237,434 images. We have worked with the meta data of these images where each image was described as a XML file. Each XMl describes an image with its id, name, respective description, caption and comment about the image in 3 different languages, i.e English, French and German. Each image also contains a part named overall comment that basically describes the source of the image, photographer name and sometimes the description of the image. For our purpose we have used the caption, description and comment section in English only.

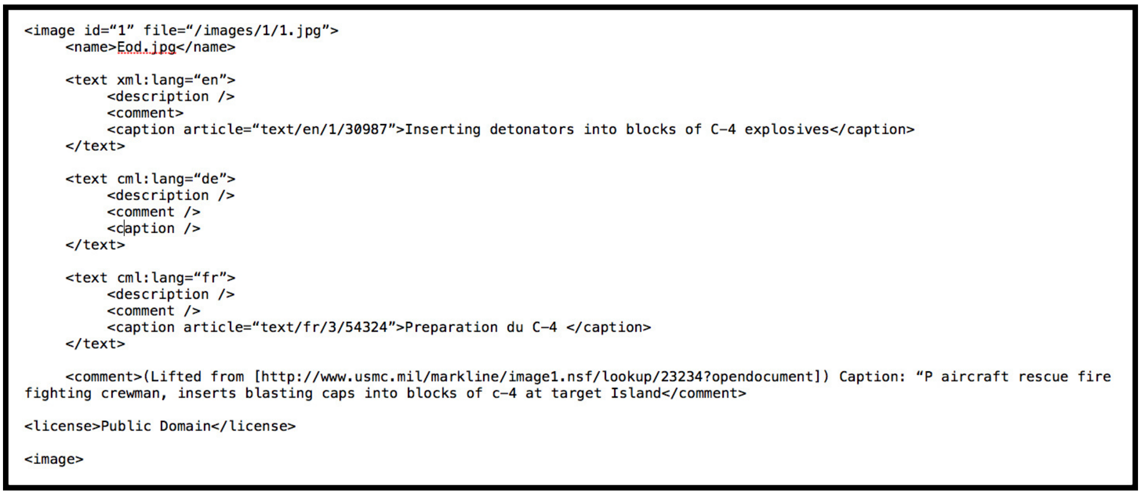
Next come to the query file that has been used to retrieve the images from the data set. In particular, we make use of 27 short stories collected from "Aesop's Fables". These query files have been formatted as XML. Each story contains the text portion where the story has been described. Next there is a list of entities which consists of the subject and the object of the story. For example, the story named "The Fox and The Crow" contains "Fox", "Crow", "Piece of Cheese" as different entities. In the "action" section of each story mainly the important verb phrase has been described, ex. “saw” , “praise” , “flatter”. In the last section, named “events”, the event happening in the story has been written, ex. “Crow has a piece of cheese in its beak”, “Fox snaps the cheese up”.

A relevance judgement file has been provided to test the retrieval performance by the TREC evaluation software. A short children’s story which has been used as query and meta data of an image has been depicted in the figure 3.1 and 3.2 respectively.



```
<annotation>
    <stories annotator="1">
        <story language="en">
            <title>The Fox and The Crow</title>
            <author>Aesop</author>
            <text>A Fox once saw a Crow fly off with a piece of cheese in its beak and settle on a branch of a tree.
          "That's for me, as I am a Fox," said Master Reynard, and he walked up to the foot of the tree.
          "Good day, Mistress Crow," he cried. "How well you are looking today: how glossy your feathers; how bright your eye.
           I feel sure your voice must surpass that of other birds, just as your figure does; let me hear but one song from you
           that I may greet you as the Queen of Birds."The Crow lifted up her head and began to caw her best, but the moment she
           opened her mouth the piece of cheese fell to the ground, only to be snapped up by Master Fox.
           "That will do," said he. "That was all I wanted. In exchange for your cheese I will give you a piece of advice for the
future:
           "Do not trust flatterers."
         </text>
            <source>http://www.eastoftheweb.com/short-stories/UBooks/FoxCrow.shtml</source>
            <entities>
                <entity>Fox</entity>
                <entity>Crow</entity>
                <entity>piece of cheese</entity>
                <entity>tree</entity>
            </entities>
            <actions>
                <action>saw</action>
                <action>praise</action>
                <action>lifted up</action>
                <action>flatter</action>
            </actions>
            <events>
                <event>Crow has a piece of cheese in its beak</event>
                <event>Fox flatters the Crow</event>
                <event>Crow caws</event>
                <event>The cheese falls</event>
                <event>Fox snaps the cheese up</event>
            </events>
        </story>
</annotation>
```

Figure 3.1: Example of Short Story Used as query

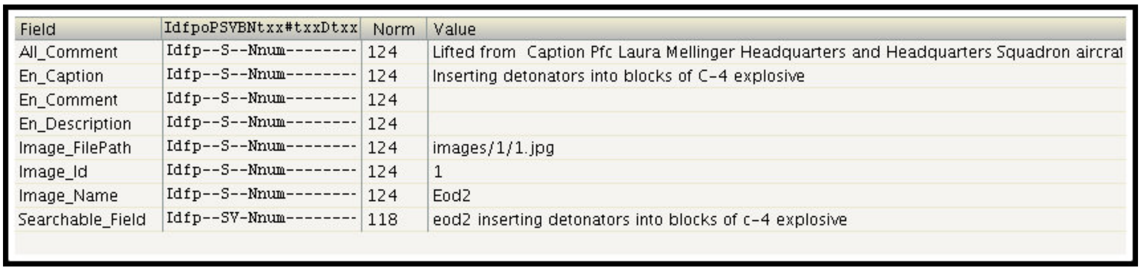


```
<image id=“1” file=“/images/1/1.jpg”>
     <name>Eod.jpg</name>

     <text xml:lang=“en”>
          <description />
          <comment>
          <caption article=“text/en/1/30987”>Inserting detonators into blocks of C-4 explosives</caption>
     </text>

     <text cml:lang=“de”>
          <description />
          <comment />
          <caption />
     </text>

     <text cml:lang=“fr”>
          <description />
          <comment />
          <caption article=“text/fr/3/54324”>Preparation du C-4 </caption>
     </text>

     <comment>(Lifted from [http://www.usmc.mil/markline/image1.nsf/lookup/23234?opendocument]) Caption: “P aircraft rescue fire
fighting crewman, inserts blasting caps into blocks of c-4 at target Island</comment>

<license>Public Domain</license>

<image>
```

Figure 3.2: Example of Image Meta data

## 3.2 Indexing

The first step of this project was to index the ImageClEF 2010 Wikipedia [5] data set. We used Apache Lucene 4.9 [9] for this purpose. For each image id, name, comment, description



| Field | IdfpoPSVBNtxx#txxDtxx | Norm | Value |
|---|---|---|---|
| All_Comment | Idfp--S--Nnum-------- | 124 | Lifted from  Caption Pfc Laura Mellinger Headquarters and Headquarters Squadron aircrat |
| En_Caption | Idfp--S--Nnum-------- | 124 | Inserting detonators into blocks of C-4 explosive |
| En_Comment | Idfp--S--Nnum-------- | 124 | |
| En_Description | Idfp--S--Nnum-------- | 124 | |
| Image_FilePath | Idfp--S--Nnum-------- | 124 | images/1/1.jpg |
| Image_Id | Idfp--S--Nnum-------- | 124 | 1 |
| Image_Name | Idfp--S--Nnum-------- | 124 | Eod2 |
| Searchable_Field | Idfp--SV-Nnum-------- | 118 | eod2 inserting detonators into blocks of c-4 explosive |

Figure 3.3: Example of an Indexed Document in Lucene shown by GUI Tool Luke

and caption about the image in English has been stored in different field. An extra field has been created, named “Searchable Field”. This field is created appending the image name, caption, comment and description. Before storing data in this field, all english stop word has been removed and also Apache Lucene 4.9 English Analyzer is used for stemming purpose. There were total 237,434 images that has been indexed. A screen shot of an indexed document in Luke software is depicted in the figure 3.3.

## 3.3 Query Formulation

As mentioned, to convert a document collection into an evaluation data set, we need to supply a set of standard queries along with relevance judgments. So Query formulation is one of the main step in the retrieval process. In this part various approach has been taken to formulate the appropriate query that describes a story best.

### 3.3.1 Full Text Based Approach

Here the full text of a story has been used to make the query. The text portion of each story has been taken and after stemming and stop word removal, the whole text has been fired to the retrieval system as a single query to retrieve top 100 relevant images.

### 3.3.2 Entity and Event Based Approach

In this experiment, the full text of the story has not been given because if the query itself is too long, there may be chance of drifting. On the other side, the entity of a story is the subject and object of the story which is very compact and crisp as query. Now a single story has more than one entity. In the first approach in this category, for each entity, 100 top relevant images have been retrieved. It may happen that an image has been retrieved more than once for different entity subject to same story. In this case the relevance score of the same images have been added and a final list of images have been created. Now this list of images have been sorted according to their score in descending order and final top 100 images are selected as the most relevant images that illustrate a story.

In the second approach in this query formulation technique, instead of using only entities of an images, each entity as well as event are used as query to retrieve top relevant document.

The relevant image documents are merged to create the final top relevant document using the same process stated earlier.

### 3.3.3 Relevance Feedback Based Approach to Query Expansion

In all the preceding approaches the MAP (Mean Average Precision) of the result was low. So Relevance Feedback [4, 19] approach has been taken to retrieve the most relevant images.

In this experiment, only the text section with the title was used as a primary query. Using this query a set of 100 relevant images was retrieved. With the relevance judgement file, this has been checked that which images are present in both list. The images which are present in both of the list for a specific query have been used to expand the actual query.

For the relevant and retrieved images, the value at the "Searchable Field" has been extracted. So for each query we have some value appended back to back, let call it as "expand part". Now for these part, TF-iDF score for every word has been calculated and a vector was formed where every components of that vector is a TF-idf score of a term. It is worth mentioning here that term-frequency (TF) is the number of time a word occurs in a document and it is calculated at a document level. The inverse document frequency (iDF) has been calculated over all the document set.

The *term-frequency (TF)* [12, 1] of a term ***t*** in a document ***d*** can be described by the formula 3.1

$$tf(t,d) = \frac{f_{t,d}}{max\{f_{t',d} \colon t' \in d\}} \tag{3.1}$$

The *inverse document frequency (idf)* [12, 1] of a term ***t*** in a full corpus ***D*** can be described by the formula 3.2

$$idf(t,D) = \log \frac{N}{|\{d \in D \colon t \in d\}|} \tag{3.2}$$

where :

- N is total no. of documents in the corpus
- $|\{d \in D : t \in d\}|$ : Number of documents where the term t appears. If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$

Besides a similar vector for the original text query has been created. After that vector addition was done and sorting was done according to the TF-iDF score. Now take the top k-terms with the highest score and formulate the query.

The Rocchioo relevance [4, 11, 19] procedure has been implemented to reformulate the query according to the formula 3.3.

$$\vec{Q}_m = \left(a \cdot \vec{Q}_0\right) + \left(b \cdot \frac{1}{|D_r|} \cdot \sum_{\vec{D}_j \in D_r} \vec{D}_j\right) - \left(c \cdot \frac{1}{|D_{nr}|} \cdot \sum_{\vec{D}_k \in D_{nr}} \vec{D}_k\right) \tag{3.3}$$

where :

$\vec{Q}_m$ : Modified Query Vector

$\vec{Q}_0$ : Initial Query Vector

$\vec{D}_j$ : Document Vector of relevant document retrieved by the initial query

$\vec{D}_k$ : Document Vector of non-relevant document retrieved by the initial query

$D_r$ : Set Of Relevant document retrieved by the initial query

$D_{nr}$ : Set Of Non-Relevant Document retrieved by the initial query

a : Original Query Weight

b : Related Document Weight

c : Non-Related Document Weight

In this relevance feedback method, an initial retrieval is done in the first iteration. Next the initial query is expanded using the terms from the top relevant and retrieved documents in the previous retrieval. That's how the query is modified in such a way that it comes closer to the relevant documents in the documents space and go further from the non-relevant documents. The above formulation can be better visualized by the figure 3.4.

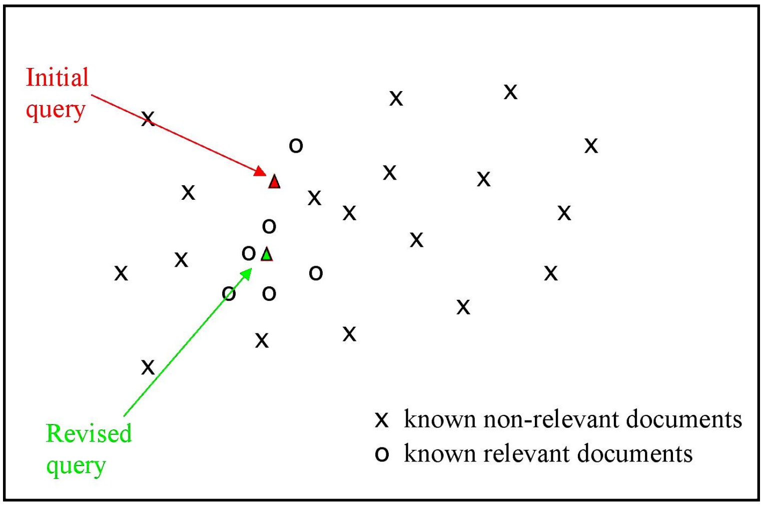


Figure 3.4: Rocchio Relevance Feedback Method

In our experiment, a = 1 , b = 1 and c = 0 has been taken. In other words only positive feedback has been considered to retrieve the relevant document. Many times in research only positive relevance feedback has been used. Positive relevance feedback is more likely to move the query closer to a user's information need. Negative feedback may help but in some cases it actually reduces the effectiveness of the query. Positive feedback moves the query to retrieve items similar to the items retrieved and thus in the direction of more relevant documents. Negative feedback moves the query away from Non-relevant document but non necessarily closer to the more relevant documents.That's why in our experiment only positive feedback has been used.

To evaluate the effectiveness of RF, the obvious first strategy is to start with an initial query $q_0$ and to compute a precision-recall graph. After one round of feedback from the user, we compute the modified query $q_m$ and again compute the precision-recall graph. In this method in both rounds we asses performance over all the documents in the collection, which makes comparison straightforward. If we do this, we find spectacular gains from RF which is the order of 50 % in MAP. But this is merely because the relevant documents are ranked higher now. Fairness demands that this evaluation should only be on the documents unseen by the user. A second idea is to use documents in the residual collection (the set of documents minus those assessed relevant) for the second round of evaluation. This is particularly the case if there are few relevant documents, and so a fair proportion of them have been judged by the user in the first round. This evaluation technique seems like a more realistic. Another approach to evaluate is to have two collection, one that is used for the initial query and relevance judgements, and the second that is then used for comparative evaluation. The performance of both $q_0$ and $q_m$ can be validly compared on the second collection.

# Chapter 4

# Results of Automated Story Illustration

As discussed, Apache Lucene 4.9 [9] was used to index the ImageCLEF 2010 [5] Wikipedia data set. Now one index has been created with this field after stemming and stop word removal. Beside another index had been created with the field appending with the image overall comment. The image overall comment generally consist of where the image was taken with the photographer name sometime with small description of the image. So it was assumed that this overall comment section will improve the retrieval performance of the system. In this regard two types of index has been made, one with the overall comment and another with excluding overall comment in the "Searchable field" of the index.

## 4.1 Retrieval Performance Using Two Method of Indexing

We have used two method of indexing. Once with including **overall comment section** to the "Searchable Field" of the indexing, and other with excluding this field.

The images are retrieved from both of this index using same queries and various query formulation technique mentioned in section 3.3. The performance of the retrieval has been evaluated by the TREC evaluation software using relevance judgement file. The table 4.1 and 4.2 shows the details metrics of performance without using overall comment and with using overall comment in searchable field respectively.

Table 4.1: Automated Story Illustration Results Without Overall Comment Section

| Model | Query Formulation | Rel-Ret | MAP | P@5 |
|---|---|---|---|---|
| BM25 (1.5, 0.75) | Story Title + Full Text | 134 | 0.0407 | 0.1909 |
| | Story Title + Entities | 167 | 0.0322 | 0.1289 |
| | Title + Entities + Events | 143 | 0.0608 | 0.1965 |
| | Relevance Feedback | **289** | **0.3213** | **0.6452** |
| LM - Jelinek-Mercer (0.4) | Story Title + Full Text | 122 | 0.0478 | 0.1643 |
| | Story Title + Entities | 178 | 0.0398 | 0.1457 |
| | Title + Entities + Events | 176 | 0.0629 | 0.2912 |
| | Relevance Feedback | 275 | 0.3124 | 0.5876 |

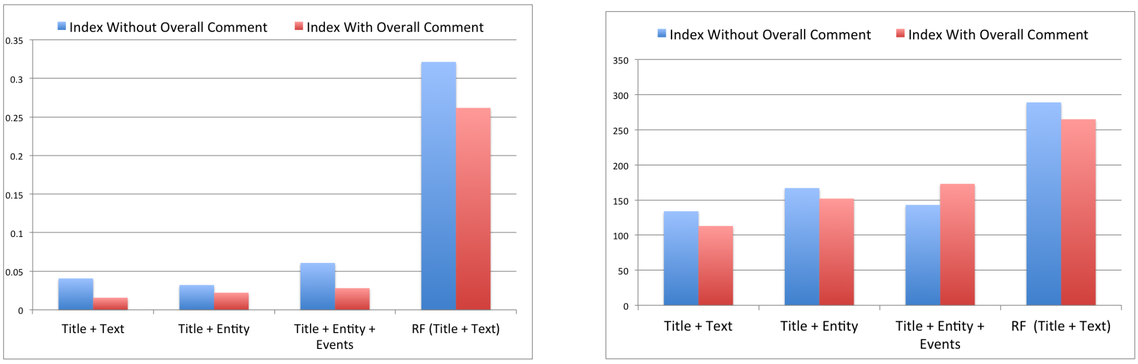
Table 4.2: Automated Story Illustration Results With Overall Comment Section

| Model | Query Formulation | Rel-Ret | MAP | P@5 |
|---|---|---|---|---|
| BM25 (1.5, 0.75) | Story Title + Full Text | 113 | 0.0157 | 0.0871 |
| | Story Title + Entities | 152 | 0.0222 | 0.0975 |
| | Title + Entities + Events | 173 | 0.0281 | 0.1090 |
| | Relevance Feedback | **265** | **0.2617** | 0.3567 |
| LM - Jelinek-Mercer (0.4) | Story Title + Full Text | 118 | 0.0183 | 0.0932 |
| | Story Title + Entities | 161 | 0.0318 | 0.1032 |
| | Title + Entities + Events | 166 | 0.0416 | 0.1671 |
| | Relevance Feedback | 242 | 0.2513 | **0.4211** |

## 4.2 Comparison of the Result

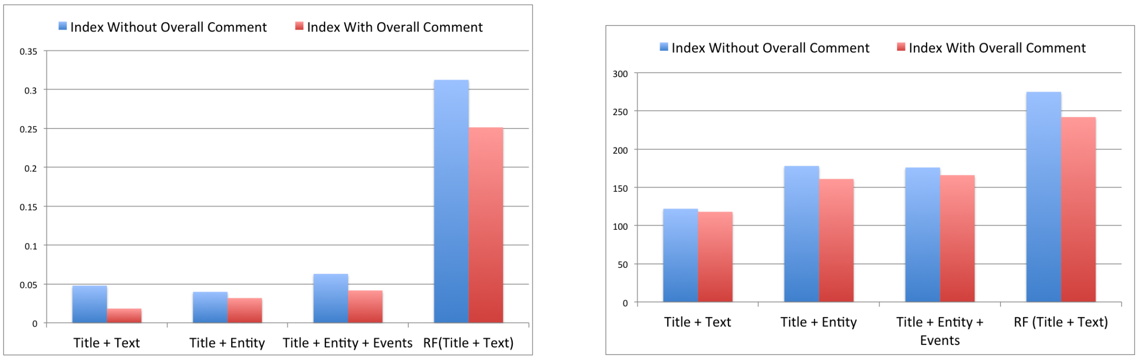
It is evident from Table 4.1 and Table 4.2 that the "Searchable Field" without the overall comment section of the image outperforms the retrieval performance of the "Searchable Field" with overall comment section. In the general context, comment section of the image should improve the result, but it has been evident from the research about the data set that the comment section generally holds http link and more over details about the photographer, where there is no correlation with the context of the image. Sometimes the language of the over all comment section is not English as this language highly dependent on the photographer's language. This has been seen that in many cases this language is German or French. So in one words the comment section contains noisy data. This is the cause of poor performance for this type of indexing.

The comparison between the result found in two way of indexing is presented through the figure 4.1 and figure 4.2 which is by mean average precision (MAP) [8] and by number of relevant images retrieved.



(a) MAP (b) No. of Relevant Retrieved

Figure 4.1: Comparison of Indexing Method Using BM25 Scoring



(a) MAP (b) No. of Relevant Retrieved

Figure 4.2: Comparison of Indexing Method Using LM Scoring

In the relevance judgement file there are total 447 relevant images for the queries. In BM25 (1.5, 0.75) model of Lucene, 289 relevant images were retrieved by relevance feedback method. The figure 4.3 shows the number of relevant images retrieved for each query as well as the total number of relevant images through a side by side comparison.

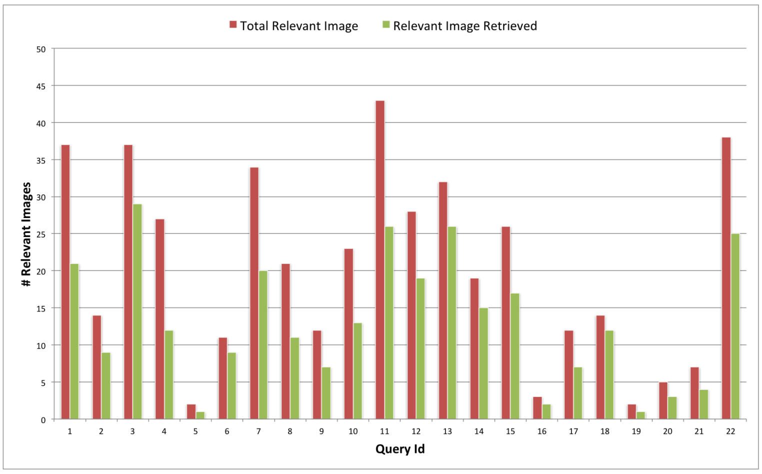


Figure 4.3: Comparison of Total Relevant Images Vs No. of Relevant Images Retrieved for Each Query

## 4.3 Limitation of the Experiment

In the preceding experiment, it is shown that the query expansion by relevance feedback method outperforms all the query formulation method and gives the better performance in all metrics of evaluation. It is stated in section 3.1 that the data set contain 234834 images and only 27 story stories as query. Though the number of images in the ImageCLEF 2010 data set [5] contains a substantial amount of images but it is very hard to establish a method's performance on the basis of only 27 queries. From this point of view, a large scale experiment of the method of query formulation has been taken into account. In this regard, all the query formulation method has been tested on the Wikipedia 2016 data dump [20]. The subsequent chapter shows the detailed data processing and experiment steps as well as the performance result.

# Chapter 5

# Automatic Illustration of Wikipedia Articles

In every data centric research problem, data set which is used to address a problem is utmost important. As discussed chapter 2, a substantial amount of work has been done on the ImageCLEF 2016 data set [5] with 27 short stories used as a query set. As discussed in the chapter 3, it is shown that query expansion by relevance feedback method outperforms all other query formulation technique but the primary question arises that depending on only 27 queries, is the claim of the performance enough. With that question in mind, an obvious solution is to apply the method in some versatile data set and query the system with substantial amount of diversified query. The subsequent section describes the process of data collection, data processing, data set generation, query generation, query formulation and relevance judgement.

## 5.1 Collection Overview

Wikipedia data dump 2016 [20] has been used in this experiment. Wikipedia dump was in the XML format. This dump consist of total 12,385,543 pages.

### 5.1.1 Data Cleaning and Generating the Data set

In this phase, the objective was to extract only textual information from each page of Wikipedia. We have started the work with the raw data dump of Wikipedia articles which consist of textual information, image caption, various file name that are linked to the page, http links and XML tags with style attributes. For our research purpose, we only need to extract only the textual information that describes the Wikipedia article. But Many pages and articles of Wikipedia do not contains any images. As the research interest is to illustrate a Wikipedia page with images, the pages which do not contain any images have filtered out in the first step of data cleaning. There were 875,957 number of pages containing one or more images.

In the next step, only the pages with images has been got. In this step, the main objective was to extract the image name, image caption or description for each image from each page. It is worth mentioning that every image in the Wikipedia page contains a caption which generally describes the image context and what is the image. So to do that the text portion of each page has been taken and with the help of regular expression the respective image and their caption has been identified. There are 875,957 pages in total which contains 2158163 images with caption. Each page has different number of images in it. The figure 5.1 shows a plot of logarithm of number of wikipedia pages with respect to the number of image it contains on the whole wikipedia data set. As the number of pages with lower number of images is much higher than the number of pages with higher number of images, so we take the logarithmic transformation of the number of pages that contains a number of images. It is evident from the figure 5.1 that number of pages that contains a specific number of images decreases as the number of images increases.

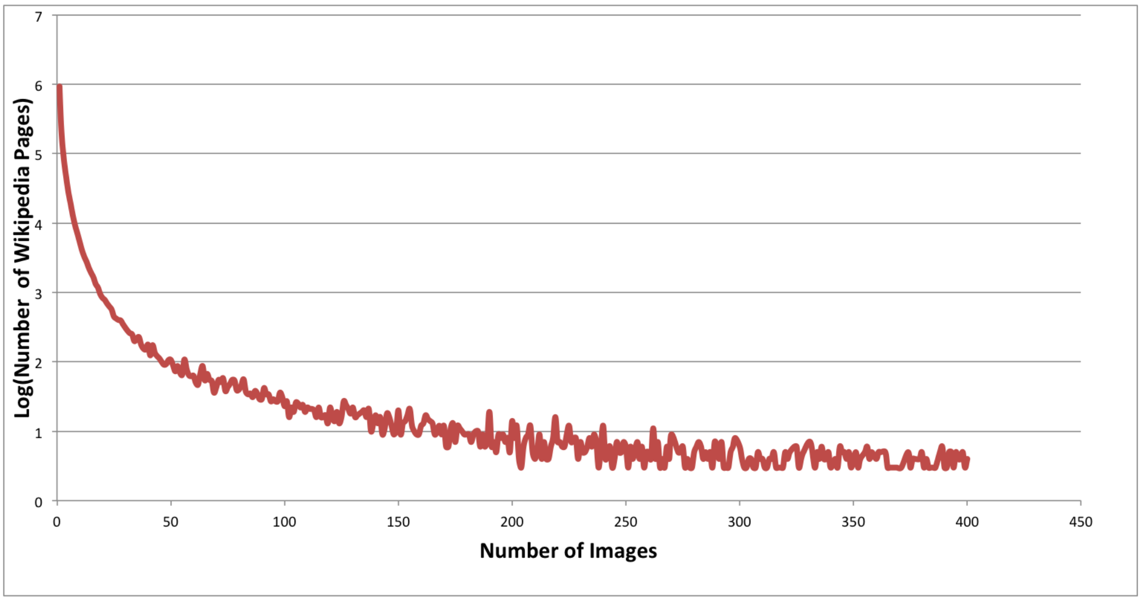
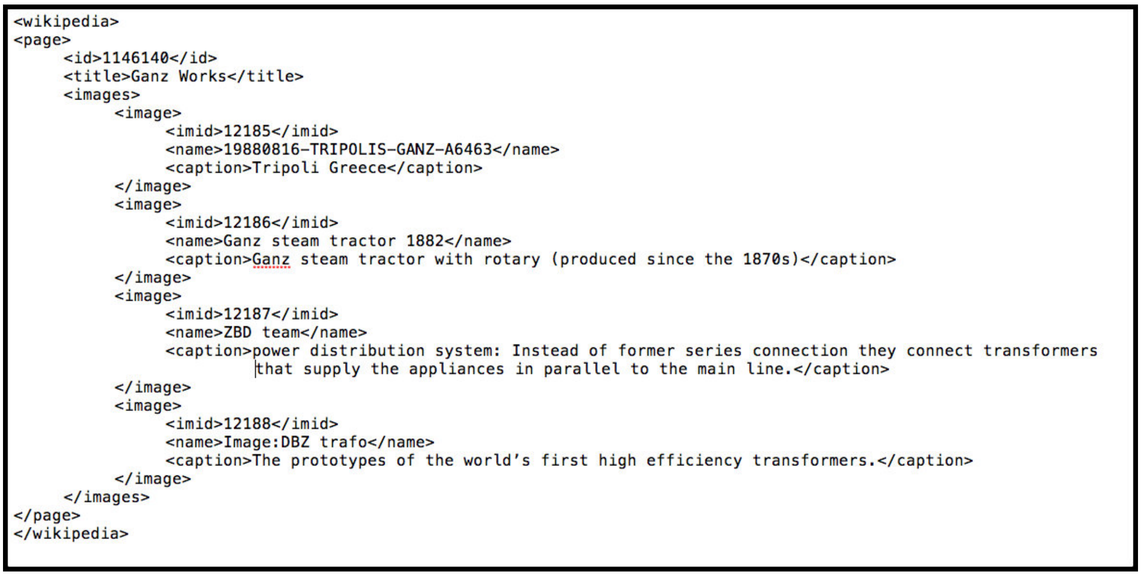


Figure 5.1: Plot Showing Number of Images per Wikipedia Page

By the above process, a well formatted XMl file has been constructed with each page identified by its page id and within each page all the images of that page with name, respective caption and program generated id. The program generated id has been set for every images to identify each images without the context of the page it associated to. This Wikipedia 2016 [20] image data set has been used to perform the experiment described in the subsequent sections. The figure 5.1 shows a Wikipedia page in image data set in XML format.

### 5.1.2 Generating Benchmark Queries for Evaluation

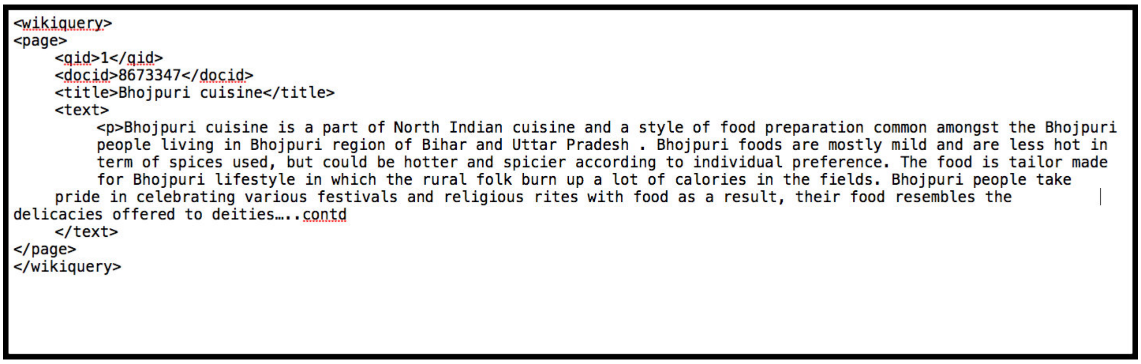
In the previous experiment with ImageCLEF 2010 data set [5], one of the main issue of that experiment was the number of queries to test the retrieval performance of the retrieval. In this context, a substantial number of queries has been generated from the Wikipedia data



```
<wikipedia>
<page>
     <id>1146140</id>
     <title>Ganz Works</title>
     <images>
          <image>
               <imid>12185</imid>
               <name>19880816-TRIPOLIS-GANZ-A6463</name>
               <caption>Tripoli Greece</caption>
          </image>
          <image>
               <imid>12186</imid>
               <name>Ganz steam tractor 1882</name>
               <caption>Ganz steam tractor with rotary (produced since the 1870s)</caption>
          </image>
          <image>
               <imid>12187</imid>
               <name>ZBD team</name>
               <caption>power distribution system: Instead of former series connection they connect transformers
                        that supply the appliances in parallel to the main line.</caption>
          </image>
          <image>
               <imid>12188</imid>
               <name>Image:DBZ trafo</name>
               <caption>The prototypes of the world's first high efficiency transformers.</caption>
          </image>
     </images>
</page>
</wikipedia>
```

Figure 5.2: Example of Wikipedia Image Dataset File in XML Format

dump. The text section of a page consist of textual description of the page with various image and formatting description. So that was full of noisy data. So by using regular expression only the textual information about the page has been extracted. Now for each page a query has been generated in XML format with page id, page title and textual information. The figure 5.2 shows a Wikipedia page in query data set in XML format.



```
<wikiquery>
<page>
     <qid>1</qid>
     <docid>8673347</docid>
     <title>Bhojpuri cuisine</title>
     <text>
          <p>Bhojpuri cuisine is a part of North Indian cuisine and a style of food preparation common amongst the Bhojpuri
          people living in Bhojpuri region of Bihar and Uttar Pradesh . Bhojpuri foods are mostly mild and are less hot in
          term of spices used, but could be hotter and spicier according to individual preference. The food is tailor made
          for Bhojpuri lifestyle in which the rural folk burn up a lot of calories in the fields. Bhojpuri people take
     pride in celebrating various festivals and religious rites with food as a result, their food resembles the
delicacies offered to deities…..contd
     </text>
</page>
</wikiquery>
```

Figure 5.3: Example of Wikipedia Query File in XML Format

There were total 300 different query that has been generated from the data set. The query pages have been chosen such that the topics are mostly diversified from different category of Wikipedia page. For example, there were page from personality, history, culture, science and technology, various cities of world, nature etc and also this has been taken into account that each such page has more than 10 images and less than 30 images to illustrate its textual information. The figure 5.4 shows a distribution of number of images over number of query that contains a specific number of images in it.

### 5.1.3 Generating Relevance Judgement File

In the last phase of data processing, a relevance judgement file in the standard TREC format has been populated. In this method the images which are presented in a Wikipedia page has

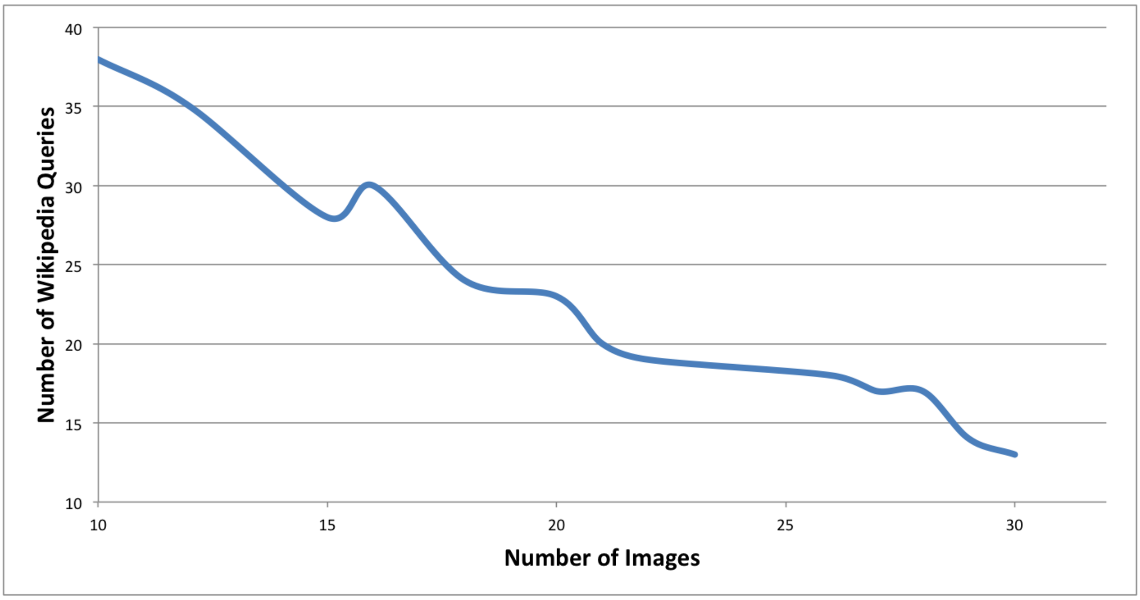


Figure 5.4: Plot Showing Number of Images per Query

been considered as relevant image to the page. So for 300 such queries a relevance judgement file in TREC format is made from the data set for evaluation purpose of the retrieval system.

## 5.2 Experimental Set up

The refined image data set, a set of substantial number of queries and a relevance judgement file in standard TREC format has been made in XMl format from the Wikipedia 2016 data dump which has been used in subsequent experiments.

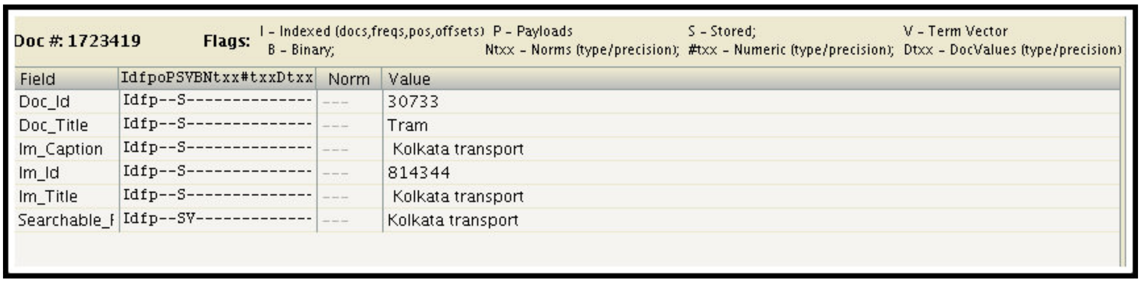
### 5.2.1 Indexing

The first step of any information retrieval experiment starts with the indexing of the data set. In this approach Apache Lucene has been used to index the XML format image data set. A standard java based XML parser has been used to parse the XML pages which has been made from the data set. For every image in the image data set, a single image has been made in the Lucene index. The field of the Lucene [9] index is as below :

- **Doc id** : This is the doc id of the Wikipedia page where the image is present.
- **Doc Title** : The title of the page where the image is present.
- **Image Caption** : This is the image caption which generally describes the image in lingual form.
- **Image Id** : This id has been generated automatically by program to every image in the data set to distinguish this from another image.
- **Image Title** : This is the name of the image given in the Wikipedia page.

- **Searchable Field** : This field has been created by concatenating image title, image caption and also analyzing by Lucene English Analyzer after stop word removed. “Searchable Field” has been used to calculate the relevance score of the image with the query.

There were total 2158163 number of images that has been indexed in this process.The figure 5.4 shows the indexed image in the luke software.



**Doc #: 1723419** **Flags:** I - Indexed (docs,freqs,pos,offsets) P - Payloads S - Stored; V - Term Vector B - Binary; Ntxx - Norms (type/precision); #txx - Numeric (type/precision); Dtxx - DocValues (type/precision)

| Field | IdfpoPSVBNtxx#txxDtxx | Norm | Value |
|---|---|---|---|
| Doc_Id | Idfp--S-------------- | --- | 30733 |
| Doc_Title | Idfp--S-------------- | --- | Tram |
| Im_Caption | Idfp--S-------------- | --- | Kolkata transport |
| Im_Id | Idfp--S-------------- | --- | 814344 |
| Im_Title | Idfp--S-------------- | --- | Kolkata transport |
| Searchable_F | Idfp--SV------------- | --- | Kolkata transport |

Figure 5.5: Example of an Indexed Image Document by Lucene Shown by GUI Tool Luke

### 5.2.2 Query Formulation

After indexing, query formulation is the next part that has been done. There has been number of ways that has been taken to formulate the query to judge the performance of the retrieval system. All the stop words from the query was removed and as well as the query was stemmed by standard English Analyzer of Lucene. The various query formulation techniques are as follows :

#### Single Query Method

A single Wikipedia page consists of a number of paragraphs. The assumption is that every such paragraph can be illustrated with one or more images.For example, a Wikipedia page about Indian Statistical Institute, kolkata consists of the introduction of ISI as well as academics in different paragraph. It is evident that in the “Introduction” paragraph there is the logo of the ISI and in “academics” paragraph there are various academic building of ISI is presented. So all these image illustrate the topic ISI, so this is very obvious that a single paragraph will be illustrated by some images which can be different from another paragraph in the same page. So for one paragraph one query has been fired to retrieve relevant images. In this approach for every page, and within the every paragraph of a page, a number of top relevant images have been extracted from index.

#### Multiple Query Method

Here the paragraph has been divided into two parts. The division is done in such a way that system calculates the total number of line in the paragraph. After that the paragraph is divided into two equal halves according to the number of lines. We have incorporated this heuristic technique because some time the length of a paragraph is huge. So if single query method is used, the verbosity of the query may harm the retrieval performance. It has been seen that a single paragraph consist of two or more diversified topic, in that case a single query fail to capture the two embedded topic in it.

#### Query Per Sentence

In this formulation, every sentence has been used to formulate a query and to retrieve images.

#### POS Tag Method with Single Query

It has been evident from a verbose query that not every term is important to retrieve a set of relevant images. That's why in this part a single query has been made for every paragraph of a page. Now in the previous method named "Single Query Method", the whole paragraph has been used for query after stop word removed and stemming. But it has been noted that not every term in a paragraph is important to its context. Besides the verbosity of the query may cause drifting from the actual context. So after stop word removal, the terms of the paragraph has been tagged according to their parts of speech. In this approach stanford NLP Maxent POStagger [17] has been used to tag the terms in the paragraph. In the testing phase the tagger learns a log linear conditional probability model from tagged text, using a maximum entropy method. The model assigns a probability for every tag t in the set of possible tags given a word and its context h, which is usually defined as the sequence of several words and tags preceding the word.

Now from the tagged paragraph only Noun phrases ("NN", "NNS", "NNP", "NNPS" ) are passed to the next phase of query formulation. The reason for choosing only noun phrases was that the nouns in a text best describe its main content in a concise and crisp way. These noun phrases has been passed to a standard Lucene stemmer and the stemmed noun phrases have been used to formulate the query and to retrieve the relevant images.

### 5.2.3 Aggregated Score Calculation

In the previous section different query formulation method has been stated. As there are more than one query that has been associated with a single wikipage, so same image will be retrieved with different score for different passage of the same wiki page. So the score given by the Lucene retrieval process of the same image can be added and a list of relevant images with the aggregated score for a whole wiki page can be found. Now the list of the retrieved images for a wikipage has been sorted according to the similarity score in descending order.

Now after aggregation of the score every query will contain one image document one times with its aggregated score. Now top 100 documents with highest score has been selected for each query and a final result file in TREC format has been generated which is used to evaluate the performance of the retrieval using TREC evaluation software.

# Chapter 6

# Result of Retrieval on Wikipedia Dataset

In the previous chapter, the various query formulation technique has been discussed in detail. In this chapter the performance of the retrieval has been showed in a tabular form. For defining similarity between query and document, Lucene provides various scoring technique such as BM-25, LM -Jelinek Mercer, Dirichlet, TF-IDF [9, 18, 12, 1]. These scoring methods have been used to find the top relevant document for a query Wikipedia page. The Table 6.1 below compasses various combination of different query formulation methods with the various scoring techniques.

Table 6.1: Retrieval Results On Wikipedia 2016 Dataset

| Model | Query Formulation | Rel-Ret | MAP | P@5 | P@10 |
|---|---|---|---|---|---|
| BM25 (1.5, 0.75) | Single Query Method | 2848 | 0.2347 | 0.3124 | 0.2217 |
| | Multiple Query Method | 3147 | 0.2378 | 0.2631 | 0.2301 |
| | Query Per Sentence | 3304 | 0.2017 | 0.2492 | 0.2178 |
| | POS Tag + SQM | 3612 | **0.3517** | 0.3921 | 0.3441 |
| TF-IDF Similarity | Single Query Method | 2615 | 0.2151 | 0.2871 | 0.2212 |
| | Multiple Query Method | 3178 | 0.2621 | 0.2411 | 0.2529 |
| | Query Per Sentence | 3142 | 0.2141 | 0.2210 | 0.2001 |
| | POS Tag + SQM | **3716** | 0.3359 | 0.3621 | 0.3212 |
| LM-Jelinek-Mercer (0.4) | Single Query Method | 2825 | 0.2605 | 0.2878 | 0.2698 |
| | Multiple Query Method | 3117 | 0.2667 | 0.2978 | 0.2767 |
| | Query Per Sentence | 3234 | 0.2897 | 0.2912 | 0.2798 |
| | POS Tag + SQM | 3695 | 0.3214 | **0.4197** | **0.3861** |

## 6.1 Visualization of the Result

In the figure 6.1, the mean average precision (MAP) of different query formulation method has been depicted by a bar plot. It is evident from the below figure that POS with single query method outperforms all other query formulation technique.

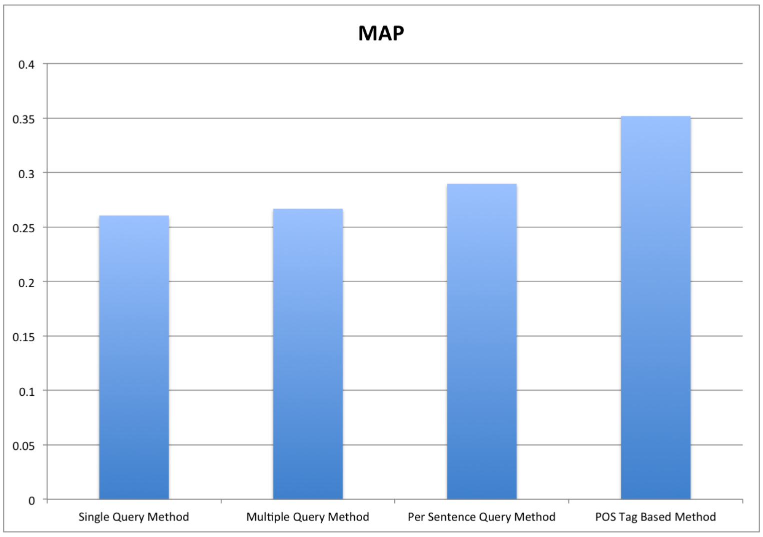


Figure 6.1: Comparison of MAP on Different Query Formulation

In this context, it is very essential to know the how many relevant document has been retrieved by the system. For 300 queries there are 4023 number of relevant images document. The figure 6.2 compares the number of relevant documents that has been retrieved by different methods. It can be seen that the POS tagged based query formulation technique retrieves 3695 number of relevant documents, i.e 88% of relevant document has been retrieved.

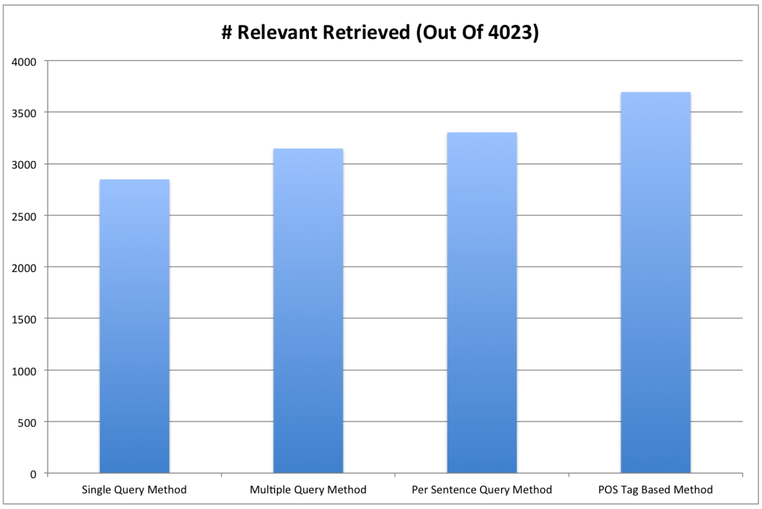


Figure 6.2: Comparison of Number of Relevant Image Retrieved

Besides MAP and number of relevant document retrieved, there are also some other popular metrics that calibrate the performance of the retrieval, such as bpref, R-Prec and Reciprocal rank. The figure 6.3 shows the comparison of these metrics through a clustered bar plot. In the figure 6.3, it is also evident that in all the metrics, the POS tag based method outperforms all other query formation technique.

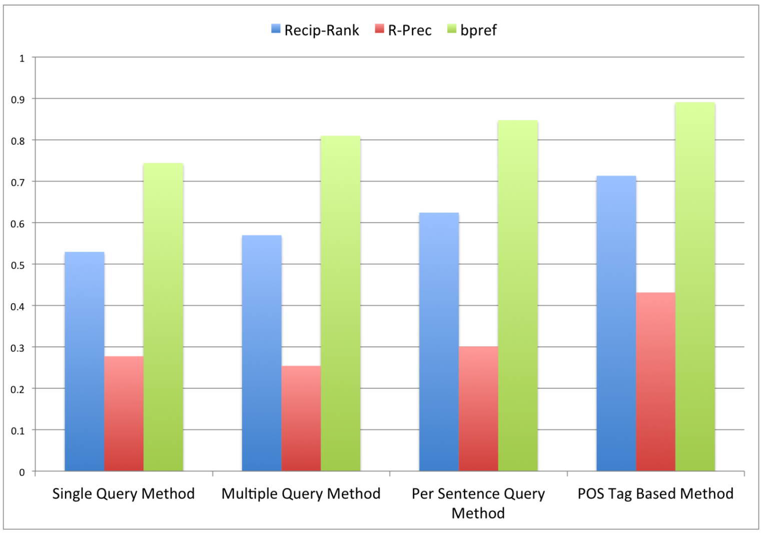


Figure 6.3: Comparison Reciprocal Rank, R-Prec and bpref in various Query Formulation Technique

# Chapter 7

# Conclusion

The problem of text illustration with images is one of the actively researched fields today. In spite of quite a big volume of work already published, a glaring void has remained in the form of absence of publicly available standard evaluation framework and data set. This is a major obstacle in objectively comparing various proposals, more so because of the nature of the field of information retrieval. Though the shared task proposed in the proceedings of "Forum For Information Retrieval Evaluation", provides a moderately good data set from ImageCLEF'2010 [5] collection, a set of query and relevance judgement, but the number of query in the query set was not so big, with merely 27 short stories. So the claim of out the performance of any approach depending on this data set is vulnerable. So, in the later phase in the work, the whole Wikipedia data set [20] with huge number of captioned image has been used. More importantly to test the performance of our approach, a set of 300 mostly diversified Wikipedia page from different category has been used as a query set.

We have used Wikipedia data set[20] in XML format.The data set has been processed to remove the noise and extract each image of a page with its text caption. With this much of information a well-formatted data set with each image and caption has been constructed. Also a set of 300 queries with only text portion of a Wikipedia page has been taken to form a query set and a relevance judgement file has been made in TREC format to evaluate the retrieval performance.

We have experimented some baseline technique for query formulation with the data set [5] and query set provided by the shared task of FIRE '16.

We have used Apache Lucene 4.9 [9] for indexing purpose and java programming languages as the main tool of implementation.For plotting purpose, we have used Microsoft Excel 2016.

We have seen by far that in the shared task experiment of FIRE'16 [2] the performance of the query expansion [6, 13] by Relevance Feedback (RF) method [11, 4] was best. But the

RF method works when we have a relevance judgement file. So we have demonstrated that query formulation using POS tagged based technique performed better in the Wikipedia'16 [20] data set. We have also presented pictorial comparison of various technique that has been taken up to counter the text illustration problem by a number of plot.

Above all, we have made a XML format image data set generated from a recent dump of Wikipedia'16 [20] which consist of more than 21 lacs of images with its caption in english. Besides we contributed a set of query and relevance judgement file in standard TREC format. These data set can be used to numerous number of experiment that can be performed in future to address the problem of text illustration with images. We contributed this data set to the field of research in information retrieval.

However much work needs to be done in more careful study of behaviours of various approaches before arriving at concrete judgments about merits and demerits of the various techniques. We have also not undertaken detailed study about the query expansion problem which should include trial with the state of the art Word2Vec or similar approaches. These remain as future work which we hope will be undertaken later.

# Bibliography

[1] Akiko Aizawa. An information-theoretic perspective of tf-idf measures. *Inf. Process. Manage.*, 39(1):45–65, January 2003.

[2] Debasis Ganguly, I Calixto, and Gareth Jones. Overview of the automated story illustration task. In *Proceedings of the FIRE Workshop*, FIRE '15, DAIICT, India, 2015. FIRE.

[3] Donna Harman. Information retrieval. chapter Relevance Feedback and Other Query Modification Techniques, pages 241–263. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[4] Donna Harman. Relevance feedback revisited. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 1–10, New York, NY, USA, 1992. ACM.

[5] ImageCLEF. Imageclef 2010 wikipedia dataset, 2010. Available from http://imageclef.org/2010/wiki.

[6] Parvaz Mahdabi and Fabio Crestani. Patent query formulation by synthesizing multiple sources of relevance evidence. *ACM Trans. Inf. Syst.*, 32(4):16:1–16:30, October 2014.

[7] Christopher D. Manning. Part-of-speech tagging from 97linguistics? In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part I*, CICLing'11, pages 171–189, Berlin, Heidelberg, 2011. Springer-Verlag.

[8] Christopher D. Manning, P Raghavan, and H Schütze. *Introduction to Information Retrieval.* Cambridge University Press. 2008, USA, 2008.

[9] Michael McCandless, Erik Hatcher, and O Gospodnetic. *Lucene in Action.* Manning Publications, USA, 2012.

[10] Sanjay S. P., N Ezhilarasan, A Kumar M, and Soman K P. Automated story illustration using word embedding. In *Proceedings of the FIRE Workshop*, FIRE '15, pages 69–72, DAIICT, India, 2015. FIRE.

[11] Raymond K. Pon, Alfonso F. Cárdenas, and David J. Buttler. Online selection of parameters in the rocchio algorithm for identifying interesting news articles. In *Proceedings of the 10th ACM Workshop on Web Information and Data Management*, WIDM '08, pages 141–148, New York, NY, USA, 2008. ACM.

[12] Thomas Roelleke and Jun Wang. Tf-idf uncovered: A study of theories and probabilities. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 435–442, New York, NY, USA, 2008. ACM.

[13] Alan F. Smeaton and Francis Crimmins. Relevance feedback and query expansion for searching the web: A model for searching a digital library. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '97, pages 99–112, London, UK, UK, 1997. Springer-Verlag.

[14] Amanda Spink. Term relevance feedback and query expansion: Relation to design. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 81–90, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[15] Tomek Strzalkowski, Jose Perez-Carballo, and Mihnea Marinescu. Natural language information retrieval in digital libraries. In *Proceedings of the First ACM International Conference on Digital Libraries*, DL '96, pages 117–125, New York, NY, USA, 1996. ACM.

[16] A Talsania, S Modha, H Joshi, and A Ganatra. Automated story illustrator. In *Proceedings of the FIRE Workshop*, FIRE '15, pages 73–75, DAIICT, India, 2015. FIRE.

[17] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, EMNLP '00, pages 63–70, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[18] Andrew Trotman, Antti Puurula, and Blake Burgess. Improvements to bm25 and language models examined. In *Proceedings of the 2014 Australasian Document Computing Symposium*, ADCS '14, pages 58:58–58:65, New York, NY, USA, 2014. ACM.

[19] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. A study of methods for negative relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 219–226, New York, NY, USA, 2008. ACM.

[20] Wikipedia. Wikipedia 2016 dataset, 2016. Available from https://dumps.wikimedia.org/.

[21] P Zainab, A Ritika, T Aman, T Sumit, and W Maitreya. Image illustration of text using natural language processing. In *Proceedings of the International Journal of Computer Applications*, IJCA '15, pages 21–26, CA, USA, 2015. IJCA.

[22] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, April 2004.