SUPERVISED LEARNING FOR IMAGE SEARCH RESULT DIVERSIFICATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BURAK GÖYNÜK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

DECEMBER 2019

Approval of the thesis:

SUPERVISED LEARNING FOR IMAGE SEARCH RESULT DIVERSIFICATION

submitted by **BURAK GÖYNÜK** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Assoc. Prof. Dr. İsmail Sengör Altıngövde
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Assoc. Prof. Dr. İsmail Sengör Altıngövde
Computer Engineering, METU _____

Assist. Prof. Dr. Tayfun Küçükyılmaz
Computer Engineering, TED University _____

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Burak Göynük

Signature :

ABSTRACT

SUPERVISED LEARNING FOR IMAGE SEARCH RESULT DIVERSIFICATION

Göynük, Burak
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. İsmail Sengör Altıngövde

December 2019, 65 pages

Due to ambiguity of user queries and growing size of data living on the internet, methods for diversifying search results have gained more importance lately. While earlier works mostly focus on text search, a similar need also exists for image data, which grows rapidly as people produce and share image data via their smartphones and social media applications such as Instagram, Snapchat, and Facebook. Therefore, in this thesis, we focus on the result diversification problem for image search. To this end, as our first contribution, we adopt R-LTR [1], a supervised learning approach that has been proposed for textual data and modify it to allow tuning the weights of visual and textual features separately, as would be required for better diversification. As a second contribution, we extend R-LTR by applying an alternative paradigm that takes into account an upperbound for the future diversity contribution that can be provided by the result being scored. We implement R-LTR and its variants using PyTorch's neural network framework, which enables us to go beyond the original linear formulation. Finally, we create an ensemble of the most promising approaches for the image diversification problem. Our experiments using a benchmark dataset

with 153 queries and 45K images reveal that the adopted supervised algorithm, R-LTR, significantly outperforms various ad hoc diversification approaches in terms of the sub-topic recall metric. Furthermore, certain variants of R-LTR proposed here are superior to the original method and provide additional (relative) gains of up to 2.2%.

Keywords: information retrieval, search result diversification, image diversification, supervised learning, tensor

ÖZ

GÖRÜNTÜ ARAMA SONUCU ÇEŞİTLENDİRMESİ İÇİN DENETİMLİ ÖĞRENME

Göynük, Burak
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. İsmail Sengör Altıngövde

Aralık 2019 , 65 sayfa

Kullanıcı sorgularının belirsizliği ve internetteki verilerin boyutu nedeniyle, arama sonuçlarını çeşitlendirme yöntemleri son zamanlarda daha önemli hale geldi. Önceki çalışmalar çoğunlukla metin aramaya odaklanırken, görüntü verileri de insanların akıllı telefonlarıyla Instagram, Snapchat ve Facebook gibi sosyal medya uygulamalarıyla görüntü verilerini işleyip paylaşmaları gibi sebeplerden ötürü çok önemli hale gelmiştir. Bu nedenle, bu tezde, görüntü arama için sonuç çeşitlendirme problemine odaklanılmıştır. Bu amaçla, ilk katkımız olarak, metinsel veriler için önerilen denetimli öğrenme yaklaşımı R-LTR'yi [1] benimsedik ve daha iyi çeşitlendirme için gereken görsel ve metinsel özellikler için ağırlıkların ayrı ayrı ayarlanmasına izin verecek şekilde değiştirdik. İkinci bir katkı olarak, sonucun sağlayabileceği gelecekteki çeşitlilik katkısı için bir üst limiti dikkate alan alternatif bir paradigma uygulayarak R-LTR'yi genişletiyoruz. PyTorch'un sinir ağı çerçevesini kullanarak R-LTR ve türevlerini kullanıyoruz ki bu, orijinal lineer formülasyonun ötesine geçmemizi sağlıyor. Son olarak, imaj çeşitlendirme sorununa en umut verici yaklaşımları bir araya getirmek için kolektif öğrenme metodunu uyguluyoruz. 153 sorgu ve 45K görüntü içeren bir

veri seti kullanan deneylerimiz, uygulanan denetimli R-LTR algoritmasının, alt konu hatırlama ölçütü cinsinden çeşitli spesifik çeşitlendirme yaklaşımlarından önemli ölçüde daha iyi performans gösterdiğini ortaya çıkarmıştır. Ayrıca, burada önerilen bazı R-LTR varyantları, orijinal metottan daha üstündür ve %2.2'ye kadar ilave kazançlar sağlayabilmektedir.

Anahtar Kelimeler: bilgi elde etme, arama sonucu çeşitlendirme, görüntü çeşitlendirme, denetimli öğrenme, tensor

To my family with love and respect

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisor Assoc. Prof. Dr. Ismail Sengör Altıngövde for his guidance, wise, and enthusiastic approach during our work. His dedication on scientific contribution always motivates me to go one step further. I am very happy to have a chance to work with him.

I would also like to thank my teammates on Web Search Engine course term project; Deniz Çelik, and Uğur Kaş. Thank you for not only being responsible teammates, but also being a life long friends.

In addition, I want to thank you Arda Bilen, for always being a good friend to me, and also delivering my papers for this thesis when I was abroad.

Moreover, I would like to thank my colleague, Bahattin Tozyılmaz, for supporting me whenever I need inspite of thousands of kilometers between us.

My special thanks go to my family, bringing me up where I am. Especially, I would like thank my father, mother, sister, father-in-law, mother-in-law, sister-in-law, grandmother, and grandfather. I can not explain how I am so grateful for their endless love, and continuous support.

Finally, I would like to thank my wife Tansu Göynük for her life-long love, friendship, encouragement, and support. You make my world brighter.

This work is partially funded by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant no. 117E861.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

MMR	Maximal Marginal Relevance
MSD	Max Sum Dispersion
MMC	Maximum Marginal Contribution
xQuAD	Explict Query Aspect Diverisifcation
DCG	Discounted Cumulative Gain
nDCG	Normalized Discounted Cumulative Gain
IDCG	Ideal Discounted Cumulative Gain
ERR	Expected Reciprocal Rank
IA	Intent Aware
sim	Similarity
rel	Relevancy
div	Diversity
IA-Select	Intent Aware Selection
R-LTR	Relational Learning to Rank
TREC	Text Retrieval Conference
SR	Subtopic Recall
CR	Cluster Recall
SVM	Support Vector Machine
SGD	Stochastic Gradient Descent
UNESCO	World Heritage Site of the United Nations Educational, Scientific and Cultural Organization
GPS	Global Positioning System
API	Application Programming Interface
P-IA	Intent Aware Precision

HOG	Global Histogram of Oriented Gradients
CN	Global Color Naming Histogram
CM	Global Color Moments
CSD	Global Color Structure Descriptor

CHAPTER 1

INTRODUCTION

1.1 Motivation

Throughout the human history, people have always been affected by the conditions and requirements of the ages they have been living in. By the foundation of World Wide Web, the latest era; Information Age was started and it changed people's life significantly. As the first impact, any bit of information was started to be shared among all people around the world, as this era connected whole humanity from all around the world. This makes earth a smaller place and encouraged more and more people to use internet. As can be seen from Figure 1.1, the number of people using internet is increased dramatically and nowadays, it seems more than half of world's population is online. It is also predicted that in the near future, the number of people using internet will continue increasing dramatically and more and more people will be online [2].

The accessibility of the internet was an important factor that led to dramatic increase in the number of people using it. During 1990s, just after world wide web is founded, the internet was not so accessible and was not open for entire world population. That initial network was connecting just few computers and systems, which were not open to public access and essentially used for the academic purposes. The spreading of personal computers was the first step making internet accessible for whole humanity. With the extensive usages of personal computers, people were able to access internet from their homes, which opened a new window from their homes to entire universe. The personal computers were good to connect internet and handle people's ordinary daily tasks. On the other hand, these ancient devices were not so flexible because

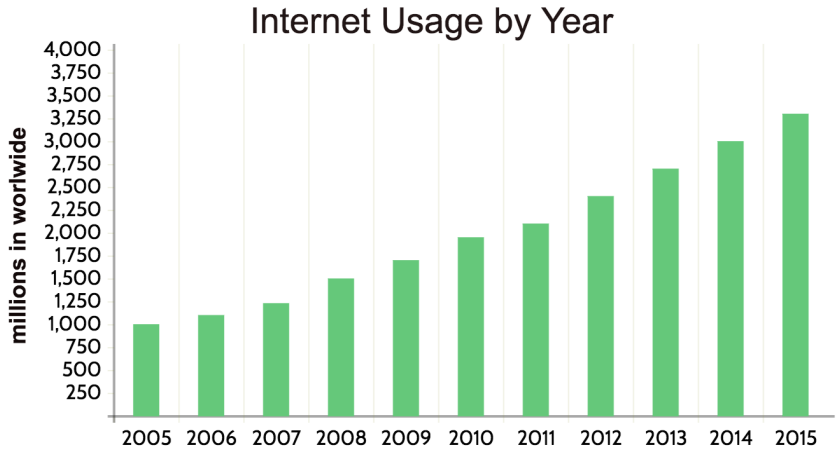

Figure 1.1: Illustration of internet usage increase over years. Data is obtained from Global Internet Report 2016 [2].

of the lack of mobility. After few decades from pioneer personal computers, smartphones took the stage and solved that problem. With them, people became able to connect internet from wherever they want. Consequently, personal computers and smartphones made internet so accessible that half of the earth population started to use internet for different purposes.

The extensive use of internet leads to significant increase on the amount of data in it. Its huge amount of the data and accessibility make people to check internet as the first resource if they need any data. In other words, internet becomes the main source of information and this makes internet the biggest library of the universe. Also, these factors are still shaping and improving that library; as the data living on the internet is growing continuously.

Analogous to the library terminology, the more books in a library, the harder it becomes to find a book. The growing size of the data prevents users from being navigated to the desired and correct data they need. In addition, in information retrieval terminology, the only interface for users to access desired data is keyword-based queries. People should type queries, which is a combination of few terms, to access their data need. Indeed, from user perspective; it becomes very hard to find the most suitable keywords for expressing correct data. In parallel to users, understanding queries is also problematic for retrieval systems. The queries, especially short

ones, do not provide a complete specification for the information need. Many relevant terms can be absent from queries and terms included may be ambiguous [3]. For example, by typing query *Python*, the user intention may be searching for python the snake, or python programming language, or even a shortcut for comedy series named as *Monty Python's Flying Circus*.

As discussed above, because of the dramatic increase of the data volume and problematic, ambiguous and unclear queries, it becomes very hard for a retrieval system to understand exact user needs and intentions. In addition, matching these intentions with the data living on the system is another problem for a retrieval system. In order to navigate users to correct resources in spite of these big problems, researchers focused on improving the search effectiveness. In this sense, a successful search engine should return a result set for a query that the lists most relevant items and at the same time, that includes as much diverse items as possible to cover different aspects (or, intents) of the query. In other words, a successful search engine should return a result set so that items in the set should be the most relevant items to the query and elements in the result set should be diverse with respect to each other. To illustrate, for the query *python*, to be sure that user will be able to find the resource she/he needs, a good search engine should include both snake, programming language and tv show aspects in the result set in order to provide a satisfying search performance.

To address the aforementioned issues, and its lots use cases (such as searching the web [4], social media [5], product reviews [6], structured databases [7], etc.) in recent years; the researchers attacked diversification problem and developed various methods to make the result set produced by the search engine diverse, i.e., covering different aspects of a query. Previous studies can be categorized into two main groups, namely, implicit and explicit diversification methods. The implicit diversification methods rely on document properties as proxies for representing the information needs covered by each document [4]. On the contrary to the implicit methods, the explicit diversification methods try to model the aspects underlying a query explicitly and rank documents to cover each of these aspect. Hence, explicit methods do not necessarily need to deal with details of the features of the documents in the collection [4]. Details of these methods, and well-known examples of them, such as Maximal Marginal Relevance (MMR) [8], Maximum Marginal Contribution (MMC) [9], Max-

Sum Dispersion (MSD) [10] and Explict Query Aspect Diverisifcation (XQUAD) [11] [12], will be discussed in the following chapters.

As the diversification problem becomes very critical for the information retrieval and different methodologies are developed to overcome this problem, various evaluation metrics and frameworks are created to evaluate the performance of proposed methods. The well-known diversity metrics such as α-*nDCG* [4], *ERR-IA* [13, 14], and *subtopic recall* [15] are developed to construct a standard about diversity scores. In this thesis, these metrics are used to evaluate the diversity of generated rankings and measure the effectiveness of proposed solution.

1.2 Problem Definition

As discussed in section below, the smartphones made serious effect on the internet usage thanks to mobility and accessibility they have provided. In addition to their contributions on the proliferation of internet, they also caused dramatic increase on the data living on internet. More specifically, the biggest impact of the smartphone usages lead to significant increase on media and image data on internet. With the help of smartphones, a new trend, namely social media, was born. This new phenomena is so powerful that nearly every online user is using social media. By 2019, 3.3 billions people are using social media from all around the world. Social media usage is very important concept; because, with its wide spreading, users become not only consumer of the data, but also become the producer, as they have started to share what they have, live or think with the others. With the latest trends on social media [5], people are tend to express their feelings with image based data through most popular social media sites such as, Facebook, Instagram, YouTube and Twitter. As a result, people produce more and more image data.

In addition to social media, another hot trend, online media also has serious effect on size of image data on the internet. As online media becomes more popular, main media resources such as series, films and shows are moved from the televisions to the online platforms and stored on the internet, which makes online media to be another important actor making image data pool become bigger.

The advancements on the technology in the last few decades also was another important factor that led to the increase of the online image data. The direct impact of technology can be examined from two different dimensions. Firstly, with the help of improvements on the technology, researchers started to find new observations to understand our universe better. By the using of the latest technology in the research areas such as space science, geography and ocean sciences, now; scientist are able to execute millions of experiments by collecting and analysing image based data. For instance, one of the hundreds of programs of NASA, Hubble Space Telescope captured 1 million observations by 2014. It is still continuing to generate 844 gigabytes image data for each month according to NASA statistics. Given that there are millions of programs and experiments run by scientists all around the world and comparing it with the data just Hubble produces on each month, we can realize that the total size of data produced by scientific experiments would be huge. Secondly, improvements on technology did not only increase number of images on the internet, but also changed size of an individual image. Thanks to improvements on both hardware and software on cameras, today; these are able to capture high quality images with their high dynamic ranges. Hence, both increase on number of images on the web and increase of size of individual image, caused dramatic increase on total image data size on the web.

In addition to natural reflection of data increase to all data types, the reasons listed above caused image data on the web increase more, when it is compared with other data types such as textual and sound based data. As a result, similar to textual based search diversification problem was becoming popular, nowadays, the problem of the image based search result diversification is also gaining more popularity from both academical and professional directions. Similar to *python* query example given in section below, when a user types a query to retrieve an image, for example *Hagia Sophia, Istanbul*, the final search result set of the query should contain images from different perspectives and within different conditions (such as taken in daylight or at night, in summer or winter) to provide a satisfying search experience. This difference between diverse and non-diverse result sets for the same query is visualized in Figure 1.3 and Figure 1.2. To solve that problem, in this thesis, we have concentrated on image based search diversification problem and proposed, implemented and evalu-

ated a new approach that is based on and adopted from textual search diversification techniques.

Figure 1.2: Example non diverse image result set for the query Hagia Sophia.

Figure 1.3: Example diverse image result set for the query Hagia Sophia.

1.3 Contributions

The main focus of this thesis is adopting and implementing cutting-edge, high performance methodologies to achieve image based search result diversification. Main

contributions of this thesis can be summarized as follows;

- We adopt a supervised learning solution, which is described in [1] and named as relational learning to rank (R-LTR). As R-LTR is designed to work on textual data, we re-formulate its ranking function and redesign its tensor structure so that it can work on multiple tensors and multiple features. Consequently, the new diversification framework, referred to as R-LTR$_{\text{IMG}}$ is operational for both textual and visual data. Thanks to newly defined tensors in this framework, the new version of the system is able to tune between and modify the weights of textual and visual features. We also integrate a technique to compute the similarity of a textual query and image result, namely *Selecting the Representative Image*, as described in [16], to our framework. This increases uniqueness of the R-LTR$_{\text{IMG}}$ framework, since *Selecting the Representative Image* approach has been usually applied for clustering purposes [16].

- By a careful analysis of our dataset (described later), we identify the most useful descriptors to serve as textual and visual features. After running several experiments, we determined the most suitable and best performing distance calculation method for each feature.

- We propose different R-LTR variations based on different learning strategies. As the initial strategy, R-LTR is implemented as a simple neural network, without a hidden layer. Then, neural network architecture is enriched by introducing hidden layers and nodes. Also, ensemble learning techniques are used to gain performance by using aggregated scores of different neural network architectures.

- Our final contribution is based on the following observation: R-LTR learns a ranking function based on an iterative selection process, where the diversity of a given document is computed wrt. the previously selected documents, i.e., following the paradigm of the well-known Maximal Marginal Relevance (MMR) diversification [8]. We extend R-LTR with an alternative approach, inspired by the Maximum Marginal Contribution (MMC) idea of [9]. While diversifying a result set, the MMC approach takes into account an upperbound for the *future diversity contribution* that can be provided by the document being scored.

As far as we know, the earlier approaches for supervised diversification (such as [1, 17, 18, 19]) essentially follow the MMR paradigm and hence, ours is the first attempt to *learn* an alternative ranking function.

- Our experiments are conducted using the *Div150Cred* dataset employed in the 2014 Retrieving Diverse Social Images Task (of MediaEval Initiative) in a well-crafted framework. For the baseline strategies, MMR and MSD (described in Chapter 2), we employed a dynamic feature weighting strategy for higher performance. For all the diversification methods, we used various pre-processing techniques, and employed a particular strategy based on representative images, to better capture the query-image relevance. We show that the adopted R-LTR$_{\text{IMG}}$ and its proposed variants outperform MMR and MSD in diversification effectiveness. Furthermore, according to the results reported in the Diversity task of MediaEval, certain R-LTR$_{\text{IMG}}$ variants are also superior to all but one of the methods explored in this campaign.

Our work presented in this thesis is accepted for publication in European Conference on IR Research (ECIR 2020) with the title 'Supervised Learning Methods for Diversification of Image Search Results'.

1.4 Organization of the Thesis

The rest of this thesis is organised as follows:

- Chapter 2 reviews previous works on the search result diversification problem.
- Chapter 3 begins with describing the work adopted in thesis, namely Relational Learning to Rank Approach (R-LTR) [1] for Search Result Diversification on textual data. Then, we describe the structural additions to support image diversification, and more crucially, propose our R-LTR variants.
- Chapter 4 describes used dataset, and its utilities. This chapter also covers baseline methods, their descriptions, and implementations. This chapter is closed by discussing the standard evaluation metrics used in the experiments.

- Chapter 5 presents our extensive experiments. In general, our experiments can be divided into two groups, while one set of the experiments focus on improving framework performance, such as epoch number analysis and feature importance analysis, the second set of experiments focus on evaluating the performance of the new solution by comparing its performance with the baseline models. This chapters ends with the discussions about our findings.
- Chapter 6 summarizes the work done, presents final discussions and provides possible future work directions.

CHAPTER 2

RELATED WORK

In this chapter, the definition of the search result diversification problem will be explained in more detail. Also, existing techniques to tackle that problem, which have been identified during literature review, will be discussed by providing their background information. The general idea of implicit and explicit diversification methodologies and their well-known example techniques can also be found on this chapter. At the end of this chapter, description of global diversity metrics and diversity evaluation techniques will be described in detail.

2.1 Diversification Problem In Detail

When a user type a query to retrieve any data, that query is processed by a search engine and desired set of information is provided to user. The search engines are positioned at the heart of information retrieval activities, which is serving billions of people to their data needs at each day.

A typical search engine operates three main duties as can be seen in Figure 2.1.

- Firstly, a search engine is responsible from crawling, which is about checking and being aware of newly generated content from entire web. The crawling is the process of discovering new contents and expanding information borders of the search engine.
- Secondly, search engine operates indexing tasks. Thanks to indexes, the systems become able to map all information or document to an index, which makes accessing and querying over documents easy.

- Finally, search engine handle ranking operation. With ranking process, the engine can generate a result set which is suitable for a query. The result set is shown to users through a user interface and they can find their data need.

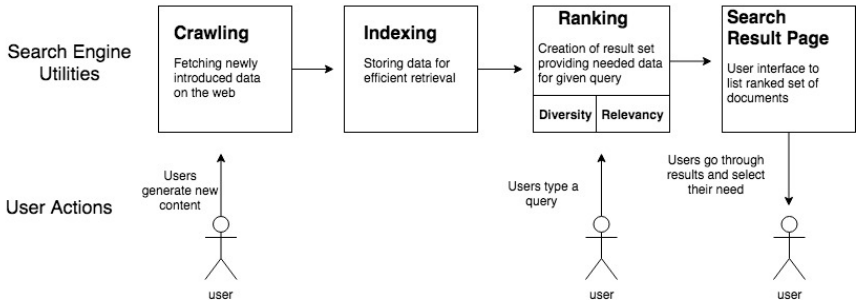

Figure 2.1: Main components of search engines, with main responsibilities of each component.

The search result diversification is one of the important concept of ranking process. As stated by Shengli Wu, Zhongmin Zhang and Chunlin Xu [20], search result diversification is usually achieved within two steps. At first step, for a given query, the search engine executes a ranking algorithm to construct a ranked list of documents by considering only relevancy of the documents with respect to query. Then, second step takes the place and it applies a diversification algorithm to re-rank constructed list to improve diversity and cover every aspects of the query as much as possible. Hence, search result diversification is highly related and coupled with ranking process of a search engine.

The dramatic increase in data size on the web, which happened in previous few decades, affected performances of search engines negatively. Because that caused lots of duplicate documents should be lived and these were listed in the same result list. In addition to growing size of data and duplicates, queries was another important factor had huge negative impact on the search engine performances. As stated in sections below, queries are ambiguous, which can refer to different meanings. Also, queries are usually composed by few terms, which makes quite complicated to understand exact user needs. These two factors forced search engines to perform better, such that each document in the constructed result set should be relevant to the query, at the same time documents in result set should be so diverse that they can cover every aspects of the query. That fact would increase the chance of desired document's

occurrence in result set, which has direct effect on user's satisfaction on search experience. As a result of these discussions, researchers focused on diversifying search results by constructing different methods for search result diversification problem.

As expressed, diversification solutions aim to increase novelty to have a maximum coverage for a query in terms of every aspects it can have. More formally, for a given set of *N* available documents relevant to query *q*, and a constraint *k*, desired length for result set, diversification process aims to select a subset *S* with *k* documents from *N* items, such that diversity between the items in *S* is maximized [21]. From probability point of view, following formula express diversity score, $P(S|q)$, which denotes chance of covering every categories of a query. The main aim is to construct a subset *S* such that the following formula is maximized [22];

$$P(S|q) = \sum_{c} P(c|q)(1 - \prod_{d \in S}(1 - V(d|q, c))) \tag{21}$$

It can be inferred from both definitions that diversification problem actually a maximum coverage problem aiming to have maximum diversity by covering every aspects of query. As this problem can be reproduced by reduction from maximum coverage problem, it can be proved that search result diversification problem is NP-Hard [4].

2.2 Existing Methods

After defining search result diversification problem formally, and proving its type as NP-hard, many researchers have tried to solve it from different paradigms. While solving diversification problems, researchers generally approach problem from two different dimensions; which are diversification strategy and aspect representation type used in that solution.

2.2.1 Diversification Strategies

Diversification strategy or approach defines how a solution aims to cover every aspects or dimensions of the query in produced result set. There are three main ap-

proaches in terms of diversity strategy used in existing solutions.

- Novelty Based Diversification Approach: This approach compares newly processing document with each document in result set and just focusing on difference of new document with respect to elements in result set. This approach is just interested in novelty introduced by each document and aims to decrease redundancy by promoting differences between elements.
- Coverage Based Diversification Approach: This approach focuses on query aspects and tries to measure each candidate document's contribution on covering all of the aspects. While considering a candidate document to be included in result set, it just checks aspect coverage of current document without comparing it with already selected documents in current result set. By this definition, prerequisite of this approach would be identification of the query and finding its all aspects, which reduces this problem to resolving query ambiguity.
- Hybrid Approach: In recent year, few solutions were developed which behaves like combination of both novelty, and coverage based approach. In general, these methods are based on learning, aim to use both coverage and novelty information as features of the entire system.

2.2.2 Solutions by Aspect Representation

In order to solve diversification problem, each document in the corpus is needed to be defined mathematically and represented on vector space model. The aspect representation of the solution determines the way of representing document in the solution space. There are two main aspect representation types used for current solutions, which are implicit and explicit diversification techniques.

2.2.2.1 Implicit Diversification

Implicit diversification techniques were the initial ones in the literature by diversification solutions according to aspect representation. These techniques represent document by using its' properties only, without knowing details of the query. With implicit

diversification, the features are defined at the beginning of the process and each document is represented as the combination of these features. It is important that these features are independent from query aspects. This diversification technique tries to estimate similarities and differences of the documents by comparing these query-independent features of each document. In brief, implicit techniques solve diversification problem from a document based approach, by representing each document as query independent features in solution domain and comparing these to construct a diverse result set covering information needs of users.

One of the pioneer solutions with implicit diversification is Maximal Marginal Relevance (MMR) algorithm, proposed by J. Carbonell and J. Goldstein [8]. The relevancy and novelty are the two different concepts should be provided together for a good search experience. As these two conditions are contrary to each other, it is not possible to cover both of them at the same time. This algorithm aims to construct a result set with the harmony of both relevancy and diversity, by using an objective function which has a parameter to tune weights of relevancy and diversity, with a tradeoff value between them. After representing each document in the corpus by well-defined features and being able to measure similarities between them, this approach formulates following objective function to construct relevance, and diverse set.

$$MMR(d, q, S) = 1 - \lambda * \text{sim}(q, d) + (\lambda) * \max_{d_i \in S}(\text{div}(d, d_i)) \tag{22}$$

As can be inferred from the objective function formula, MMR tries to tune between relevancy and diversity. While, left side of the equation, namely $sim(q, d)$ is responsible from constructing a result set from relevance documents to query, the right side tries to extend resultset with documents introducing novelty by comparing it with the all documents in the current result set.

The lambda value, namely λ, in the formula is called as trade-off value or diversity coefficient and it makes algorithm able to tune between relevance and diversity. With increasing λ value, algorithm produces more diverse results with less relevancy, on the other hand; with lower λ values, algorithm tends to produce more relevant results, which may contain more redundant documents because of the lack of diversity. To

sum up, while comparing two runs with λ is 0.1, and 0.9; it is shown that while first run was better in terms of relevance metrics such as precision, second run was better on diversity metrics such as cluster recall.

There are lots of variants of the MMR, as it inspired many algorithms such as the formula below.

$$MMR(d, q, S) = (1 - \lambda) * \mathrm{sim}(q, d) + \frac{\lambda}{|S|} * \sum_{d_i \in D} \mathrm{div}(d, d_i) \tag{23}$$

This function just contributes general idea of the MMR by measuring diversity. Instead of defining diversity as the maximum distance between the current document and all documents in the corpus, the formula above calculates diversity score as the average of distance between current one to all documents.

Although the objective functions may have small differences among MMR versions, the general execution of the algorithm is the same for all of them. The algorithm starts with its execution with an empty set, namely S, and tries to construct a result set with k documents from a candidate set, R. For each iteration, it tries to select the document with highest mmr score(or any other objective function), puts selected document to result set, and extracted it from possible candidate documents. The formal explanation of this greedy based search algorithm can be found in Algorithm 1.

Algorithm 1 Greedy Search Based MMR Algorithm

1: $S \leftarrow \emptyset$
2: **while** $|S| < k$ **do**
3: $\quad d_i \leftarrow \mathrm{argmax}_{d_i \in R}(\mathrm{mmr(d_i, S)})$
4: $\quad S \leftarrow S \cup d_i$
5: $\quad R \leftarrow R \setminus d_i$
6: **end while**
7: **return** S

Another popular solution for implicit diversification is Max-Sum Dispersion [10]. Similar to MMR, this method conducts greedy search to construct a final result set based on an objective function. Unlike MMR's objective function, this method takes

two documents as input and returns score of the document pair. Hence, this algorithm works on document pairs, instead of processing one document at a time.

$$MSD(d_i, d_j, q) = (1 - \lambda) * (\mathrm{sim}(d_i, q) + \mathrm{sim}(d_j, q)) + 2 * \lambda * \mathrm{div}(d_i, d_j) \quad (24)$$

This formula returns a score of two documents, namely d_i and d_j, taken as input, by having similarity scores of these documents with respect to query and diversity scores between the documents. By multiplying these scores with the tradeoff value, objective function returns an output score of the processing pair.

The following algorithm illustrates the construction of result set by using MSD objective function;

Algorithm 2 Greedy Search Based MSD Algorithm

1: $S \leftarrow \emptyset$
2: **while** $|S| < \mathrm{floor(k/2)}$ **do**
3: $\quad d_i, d_j \leftarrow \mathrm{argmax}_{d_i, d_j \in R}(\mathrm{msd(d_i, d_j)})$
4: $\quad S \leftarrow S \cup [d_i, d_j]$
5: $\quad R \leftarrow R \setminus [d_i, d_j]$
6: **end while**
7: **if** $is\_odd\_number(k)$ **then**
8: $\quad d_k \leftarrow get\_random\_doc(R)$
9: $\quad S \leftarrow S \cup [d_k]$
10: $\quad R \leftarrow R \setminus [d_k]$
11: **end if**
12: **return** S

As can be inferred from definition above, at each iteration, algorithm tries to find pair of two documents, which has the greatest msd score and put these to current result set. For the given resultset length, namely k, the execution is finalised with $k/2$ execution if k is even. Otherwise, if k is odd, algorithm terminates by executing one more statement, by selecting a random document from candidate set, and appending it to result set.

In addition to heuristic based extensions of the MMR, there are also some effective

extensions of it are developed using probabilistic models. One of the most popular solution influenced by MMR and based on probability is *Risk Minimization*. Zhair and Lafferty [23] developed a framework to calculate score of each document, with given query and current result set by calculating loss value based on probabilistic models and tries to reduce loss values at each iteration.

Maximum Marginal Contribution (MMC) [9] is another approach, which is very similar to MMR, but in addition to taking into account the documents already selected in to S, MMC also considers somewhat an upperbound on the future diversity, i.e., computed as the contribution of the most diverse l documents to the current document d. In Equation 25, the first two components are exactly same as MMR, while the third component captures the highest possible diversity that can be obtained based on d, in case that it is chosen into S.

$$MMC(d, q, S) = (1 - \lambda) * \text{rel}(q, d) + \frac{\lambda}{|S|} * (\sum_{d_i \in S} \text{div}(d, d_i) + \sum_{\substack{l=1 \\ d_j \in D-S-d}}^{k-|S|-1} \text{div}(d, d_j)) \tag{25}$$

2.2.2.2 Explicit Diversification

The explicit diversification technique tackles diversification process from a query-oriented point of view, unlike implicit diversification. As general process of the explicit diversification, initially, these techniques try to define every aspects or subtopics of a given query. As Ozdemir and Altingovde mentioned [24], generally this is done by identifying all possible ambiguities and reformulations of the query. After identifying every dimensions of the query, these techniques try to cover every dimensions in final result set by matching these dimensions with the documents. By the definition, the main challenge of this process is to find every aspects of the query to cover every information need.

Intent Aware Select (IA-Select) is one of the earliest explicit diversification algorithms in the literature. Agrawal et al. [22] developed a method able to get relationships of queries and documents by the categories on a taxonomy. Thanks to that classification, this method is able to represent each document and query in category

domain. As related categories of each element in the corpus is known, diverse result set is generated by selecting and promoting elements whose categories not in current result set. As a result of this process, result set does not include redundant documents covering the same topics, as selection of documents causing redundancy are fined.

$$IA - Select(S, q, d) = \sum_{c \in \tau} f(c|q, S_i) * f(d|q, c) \tag{26}$$

The objective function returns score with given query q, document d, and current result set, S. The returned score is calculated by iterating through each category defined in taxonomy and measuring introduced novelty of the candidate document by comparing remaining categories of the query not included in current result set and possible categories can be contributed by the current document. IA-Select method uses the same greedy approach with MMR, as described in Algorithm 1. In general, this algorithm iterates through each element in candidate document set and tries to select the one, which maximizes objective function at each time.

Santos et al. [4, 12, 11, 25] introduced another state-of-art explicit diversification technique, Explict Query Aspect Diverisifcation (xQuAD). xQuAD framework sorts out the biggest problem of explicit diversification, which is identifying all query aspects, by gathering all reformulations of a query from TREC subtopics and search engines, such as query logs of a search operation. Thanks to these information resources, the framework is able to extract information needs of a query as much as possible. After identifying all possible dimensions of both queries and documents, xQuAD algorithm defines an objective function working on probabilistic combination of diversity and relevancy.

$$xQuAD(S, q, d) = (1 - \lambda) * P(d|q) + \lambda * P(d, S|q) \tag{27}$$

As it can be inferred from the definition of the objective function, $P(d|q)$ represents relevancy on probabilistic model, while $P(d, S|q)$ denotes diversity. xQuAD algorithm works with the greedy approach as expressed below to construct a result set with the documents producing maximum score output from objective function.

Both IA-Select and xQuAD try to identify query aspects and promote documents covering as much as dimensions thanks to coverage part of their objective functions. In addition, intersection of uncovered categories of the query in current result and aspects of the current document is also important for selection of the document. Hence, these two algorithms care both coverage of the all query aspects and novelty introduced by each element, which make these algorithms work as hybrid diversification in terms of their diversification strategies.

To sum up, while diversification strategy of a solution describes the way of handling diversification, a general structure of the solution; aspect representation defines the way of representing document on the solution space. Together, these two important concepts formulates the solution. The Table 2.1 summarizes characterics of the explained algorithms according to these two dimensions.

Table 2.1: Algorithms By Diversification Strategy and Aspect Representation.

Algorithm	Diversification Strategy	Aspect Representation
MMR	Novelty Based	Implicit
MSD	Novelty Based	Implicit
MMC	Novelty Based	Implicit
xQuAD	Hybrid	Explicit
IA-Select	Hybrid	Explicit

All algorithms explained below works with a greedy approach, as expressed formally in MMR algorithm. As the main working principle, these algorithms iterate whole candidate documents sequentially and try to construct result set by getting the one with maximum score among all elements. As an alternative approach to greedy one, thanks to improvements on machine and deep learning techniques, some algorithms are developed using learning approaches for candidate document selection process. In general, these types of algorithms try to optimize their objective function parameters by training algorithm on development dataset. Then, algorithms use optimized parameters on objective functions and select documents accordingly. Some of the famous learning based examples such as Supervised Learning, or Relational Learning to Rank methods can be found later sections in detail.

2.3 Diversity Measurement Techniques

As examined in the previous chapter, there are many solutions developed to solve search result diversification problem. The increasing number of solutions caused another problem to be solved, which is measurement of the effectiveness of a solution. In other words, the evaluation of diversity was another problem researchers faced with in the information retrieval terminology. We will discuss few well known diversity evaluation techniques by providing rationales behind them in this section.

One of the pioneer approach for evaluating variety of a result set is subtopic(cluster) recall, which was influenced by explicit diversification techniques. For a given query q, and a result set R, subtopic recall is calculated by dividing number of subtopics covered by result set to number of subtopics can be generated from q.

$$subtopic\text{-}recall(S, q) = \frac{\text{num-covered-subtopics(S)}}{\text{num-covered-subtopics(q)}} \tag{28}$$

where *num-of-subtopics* for a set *S* is the length of union of covered subtopics by all documents in S.

The *subtopic recall* formula above outputs ratio of covered subtopics of the query by given result set. So, the more diverse result sets produce higher scores, while the result sets with redundant documents are tend to produce lower scores. There can be some possible variants of the subtopic recall parametrized by a cutoff value, namely l. To illustrate, while *subtopic-recall@10* represent recall value of a result set with length 10, *subtopic-recall@20* expresses recall value of result set with 20 documents, or first 20 documents in the result set.

In addition to subtopic recall, Discounted Cumulative Gain (DCG) is another metric for evaluating effectiveness of a result set. This metric is designed by considering positions of each document in result set. Under the assumption of the most relevant documents to the query should be on lower indexes and newly selected documents should be determined according to their relevancy by query, that metric defines following function to measure cumulative gain at position i by using logarithmic reduc-

tion factor according to index of the document in the result set [26];

$$DCG@i(S,q,d) = \sum_{k=1}^{i} \frac{2^{sim(q,S_i)} - 1}{\log_2(i+1)} \tag{29}$$

As can be understood from the formula, *DCG@i* outputs a score for given query and result list. It is obvious that this function can return different maximum or minimum values for different queries. In order to evaluate complete search engine performance, evaluation method should be executed with all queries and results of each query should have the same impact on the final output. That requires each DCG metric by query should be normalized to have a standard form. Thanks to nDCG@i formula below, this normalization is done by dividing current DCG score of the result set to ideal DCG score can be produced for the current query.

$$nDCG@i(S,q,d) = \frac{\mathrm{DCG@i(S,q,d)}}{\mathrm{IDCG@i(S,q,d)}} \tag{210}$$

IDCG is calculated as follows.

$$IDCG@i(S,q,d) = \sum_{k=1}^{k=|REL_i|} \frac{2^{sim(q,S_i)}}{\log_2(k+1)} \tag{211}$$

Term REL_i in Equation 211 denotes the list of relevant documents, which is already ordered by relevance of documents in the entire corpus up to position i.

After inspiring diversification strategies, the methodology of intent awareness also had huge impact on diversity evaluation metrics. Agrawal et al. adopted the concept of intent awareness to diversity evaluation metrics and introduced one of the avant-garde approach, namely *nDCG-IA*. By definition of diversity, there can be multiple intents of user to type a query. With introduced intent awareness on DCG formula, following function becomes able to inject query aspects and by multiplying DCG score with probability of an intent in a query, which is given [13]. The probability of an intent in a query is defined as *P(i|q)* in the formula below, and by definition; it

should be between 0 and 1.

$$nDCG - IA@i(S, q, d) = \sum_{c=1} P(c|q) * nDCG_c@i(S, q, d) \tag{212}$$

The *Expected Reciprocal Rank* is another metric, which measures search result performance based on user cascade model. During development of that method, it is assumed that; the method for calculating position where user stops to search for desired information can be calculated within linear time complexity, so, ***ERR*** metric for a result set with n documents is described as follows [27];

$$ERR(S, q) = \sum_{i=1}^{|S|} \frac{1}{i} * P(\text{user finds information need at index i}) \tag{213}$$

ERR metric is improved by involving intent aware methodologies, like DCG. Similar to nDCG-IA, with the identification of all possible intents for a query, ERR-IA metric reduces total query execution to be processed within all possible aspects for a given query. It is computed for each category included a query, by multiplying ERR score of a given query intent pairs with the probability of the given intent within a query. More formal expression of ERR-IA can be found in Equation 214.

$$ERRIA@k(S, q) = \sum_{i \in I_q} P(i|q) * ERR@k(S, q) \tag{214}$$

CHAPTER 3

SUPERVISED LEARNING APPROACH FOR IMAGE SEARCH DIVERSIFICATION

In this chapter, main contribution of our work and proposed approach will be presented in more detail. Firstly, preliminaries will be discussed in order to review the related background. As tensors are highly critical components of our learning based approach, we also briefly review tensors. Then, we describe the supervised learning solution of [1] for the diversification of textual results. Next section will introduce our proposal, which is adopting the latter supervised learning solution for the image data. Details of relevancy, diversity calculations, extracted features, learning strategies and optimization processes are presented in this section. Finally, we also define additional improvements to provide a better search experience such as face detection, and geographical filtering.

3.1 Preliminaries

The prerequisite information to understand proposed solution is described in this section. In particular, we first describe the applications of learning for diversification in general. In addition, as tensors play a key role in the proposed solution, these methodologies are also described.

3.1.1 Learning In Search Result Diversification

As expressed above, a typical search engine has three main components crawling, indexing, and ranking. With the improvements in learning based approaches, deep

and machine learning algorithms are started to be used on various areas. One of the most promising area of learning approaches is the information retrieval as these two concepts are dealing with the same set of problems, which can be reduced to the same root problem.

Learning based algorithms are widely employed in search engines to improve performance. For example, machine learning based clustering solutions make indexing becomes by combining related documents together, and expressing them as a single set of documents [28]. As ranking process depends on the features of each document in candidate set, selecting correct features to represent document is very important for the entire information retrieval process. To extend retrieval process from working only on handcrafted features with small amount of data and taking advantage from big data, there are some deep learning based algorithms are developed, which contribute whole retrieval process and provide promising results [29]. In addition, some machine learning based user modeling algorithms are implemented to track user behaviours and adapt retrieval system according to personalized structure [30].

In addition to achievements above, learning based approach also had important effects on diversification process. Since a query is unstructured and just combination of few terms, most of the search queries are ambiguous and hard to understand exact user need from retrieval system point of view. Machine learning algorithms make contribution on that area to represent all categories of the query, so that retrieval system can provide every possible user needs [31]. Also, as trade-off based algorithms become popular among diversity solutions, varied from MMR, the importance of parameters is increased as search engine performance is directly affected by those parameters. Thanks to learning based algorithms, systems optimize parameters by learning, and train themselves by making practices on development environment. In general, learning approaches achieve this by defining a loss function, measuring the difference between system output and optimum result, and tries to decrease the loss value at each iteration. Details of the supervised learning diversification approach can be found in the next sections.

3.1.2 Tensors and Its Applications

Tensors are algebraic objects, which are defined on the vector space and representing multidimensional arrays of scalars. Tensor is the most generalized way to declare scalar based data. Indeed, every scalar quantity is a specific type of tensor in algebraic language. To illustrate;

- Tensors with 0 rank are named as scalar.
- Tensors with 1 rank are named as vector.
- Tensors with 2 ranks are named as matrix [32].
- Tensors with more than 2 ranks are named as high order tensors.

As can be seen in Figure 3.1, tensors are higher order generalizations of matrices, which makes this data type appropriate for representing multi-aspect data [33].

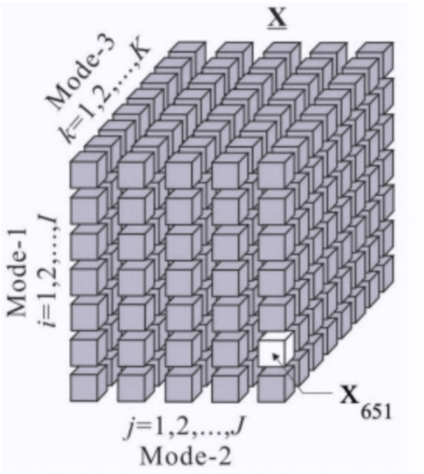

Figure 3.1: Tensor with 6 ranks. This tensor is represented as X[i][j][k], and selected element is represented as X[6][5][1].

Thanks to its power for representing multi-aspect data, tensors are used in lots of different domains such as recommender systems, anomaly detection and social network analysis [33]. In order to extract relationships of the desired aspects from high order tensors, and use data keeping these relations, matrices are needed to be reshaped, and reduced to contain only needed data. Matrix reshaping is the name of general process

to unfold a N-way tensor into a matrix by reordering its elements [34]. For instance, a 3 * 4 * 5 tensor can be reduced to 3 * 20, or 12 * 5 matrices. For this case of 3 dimensional tensors, this process is achieved by incrementing keeping only one dimension is the same, and iterating by one over other dimensions. As a result of this process, one slice of the tensor is obtained, which is a matrix. Following equation formally defines matricization process of this example;

$$X_{(1)} = [X_1, X_2, ...X_k] \in R^{ijk} \tag{31}$$

where X_i defines a slice, or matrix of tensor R.

3.2 A Supervised Learning Approach for Textual Diversification: R-LTR

As output of ranking process has a direct impact on search engine performance, it is one of the most critical components of the search systems. Generally, concepts of relevancy and diversity are handled on search engine together, instead of considering separately, as illustrated in ranking functions or algorithms expressed above such as MMR, or MSD. With the extensive use of these types of algorithms as initial approaches, researchers tried to make preliminary tests to measure search engine performance by tuning few parameters on small amount of data and optimizing search performance. Although there were few improvements, unfortunately; such ad-hoc approaches couldn't improve performance dramatically because of the manual involvement needed to be done. With the proliferation of new era, new techniques are developed to solve ranking problem based on machine learning algorithms and this methodology named as Learning-to-Rank. As a general idea, these type of techniques employ two methods; a ranking function and a loss function. The ranking function is responsible for calculating scores of each document with given conditions, while loss functions aim to calculate possible error of the generated result set. At each iteration, system updates coefficients of ranking function by getting insights from loss values. Newly introduced Learning to Rank methods, such as SVM, RankBoost, RankNet provide promising results because these solutions are able to work on large sets of training data and optimize parameters [35].

Traditional Learning-to-Rank methods focus on representing documents individually, which forces algorithms to work on just query-to-document dimension. In other words, these solutions do not care inter document relationships, which is a key concept to achieve diversity. Inspired by the main idea of Learning-to-Rank methodologies, a new concept, namely Relational-Learning-to-Rank solution is developed and this new concept becomes able to work on document-to-document dimension. Essentially, the latter approach extends main idea of its ancestor by considering relations between documents [35].

Zhu et al. proposed a Relational-Learning-to-Rank framework based on supervised learning approach to solve the text based search diversity problem [1]. As most of the popular diversity solutions, this solution also achieves diversity problem from both relevancy and diversity points of view. Their framework attempts to optimize parameters for the components that capture the relevancy and diversity.

For the relevancy component of the ranking function, definition of similarity between queries and documents is needed. Instead of expressing similarity between a document and a query by a single value, a feature vector is used for this representation, because there can be different valued similarities for different features. As a result of this, system becomes able to be trained and learn more important features for deciding query - document similarity. Hence, for a document, a vector is calculated, where each element in the vector representing a similarity score between that document and query by a specific feature. More formally, let X_{d1} be a vector containing similarity scores between document *d1* and query q. And let system provides k features to measure query document similarity.

$$X_{\mathrm{d1}} = [x_{\mathrm{d10}}, x_{\mathrm{d11}}, .., x_{\mathrm{d1k}}] \tag{32}$$

Formula can be generalized by considering all documents By generalizing the formula 32 for all documents, formula 33 is obtained.

$$X = [X_{\mathrm{d1}}, X_{\mathrm{d2}}, ..., _{\mathrm{d1n}}] \tag{33}$$

Where X_{di} denotes feature vector keeping similarity scores for document *di* and query

and *n* represents number of documents in candidate set for a given query. As any vector X_{di} has length of *k*, relevancy feature vector size, and *X* has *n* vectors, number of documents, it can be easily inferred that *X* is a matrix with size *n* * *k*.

Keeping document to document relationship is more challenging because each document has different features, and to reflect that relationship each feature should be represented properly. That fact makes diversity calculation is more complicated than relevancy component. Like relevancy, basic unit of diversity component is a feature vector of a document. A document d_i, which has l features, is represented on vector model as;

$$d_i = [feature_1, feature_2, ..., feature_l] \tag{34}$$

Document to document relationship is calculated by getting distances for each features in vector. To reflect all these features affect to document to document diversity score with the same weight, a new feature vector is calculated such as;

$$d_{\text{ij}} = [div\text{-}score_1, div\text{-}score_2, ..., div\text{-}score_1] \tag{35}$$

where d_{ij} denotes diversity feature vector representing diversity scores between document i and document j. Each element in vector, namely $div\text{-}score_1$ is calculated by computing distances of each feature in document feature vectors. In other words, $d_{ij}[2]$ represents diversity score between document i and j with respect to second feature.

As there are n documents in candidate set, all relations between each document should be represented in a complex data structure instead of scalar values. This is achieved by defining a tensor with size $n * n * l$, namely R^{ijl} where R^{ij} is a vector representing relationship between *document i* and *document j* as d_{ij} above, and element R^{ijk} denotes k-th feature of diversity between *document i* and *document j* [1].

By using data structures above, the following ranking function is defined;

$$f_s(d_i, R_i) = \omega_r * x_i + \omega_d * h(d_i, R_i) \tag{36}$$

such that the term $\omega_r * x_i$ is responsible for calculating relevancy component of the ranking function, where $\omega_d * h(d_i, R_i)$ aims to promote diversity between current document, among elements in candidate set in S. Specifically, x_i represents relevancy feature vector of d_i with respect to given query. $h(d_i, R_i)$ denotes relational function, and outputs a scalar for given tensor R_i, and feature vector of d_i. Details of the relational function h will be expressed in sections below. Finally, parameters ω_r and ω_d denotes relevancy, and diversity coefficient of the framework. By learning methods mentioned later, framework tries to optimize those parameters to provide best search experience.

As can be inferred from the definition of ranking function, this method is executed by each variants of current candidate set, namely S. Framework starts execution with S as empty set, formally $S = \emptyset$, and tries to select one element at each iteration, which document produces highest ranking function score. Hence, generalized version of ranking function includes many closures of f by current result set, and formal definition of it can be found below [1];

$$F(D, R) = (f_{S\emptyset}, f_{S1}, f_{S2}, ..., f_{Sn}) \tag{37}$$

where D represents set of all candidate documents, and R denotes diversity feature tensor as expressed above. Hence, in the light of Equations 36 and 37, ranking function of R-LTR is defined as follows:

$$\text{R-LTR}(d_i, R_i, S) = \omega_r * x_i + \omega_d * h_S(R_i) \tag{38}$$

3.3 Diversification Framework Image Search: R-LTR$_{\text{IMG}}$

We suggest that representing diversity component with one tensor (as in R-LTR) is inadequate because an image has both textual and visual components. While title

and comments of an image are considered as its textual data, RGB values, and HOG features are its visual descriptors. To represent an image with all features it has, and enable those features to affect diversity calculation, ranking function in Eq. 38 is extended by adding one more tensor which contains visual features of document to document diversity. Hence, new version of the ranking function becomes:

$$\text{R-LTR}_{\text{IMG}}(d_i, R_i, S) = \omega_r * x_i + \omega_{textDiv} * h_S(RT_i) + \omega_{visDiv} * h_S(RV_i) \quad (39)$$

In Eq. 39, the 3-way tensors RT and RV store the pairwise image diversity scores based on the textual and visual diversity, respectively. We refer to this adopted version as $\text{R-LTR}_{\text{IMG}}$.

Revised version of the ranking enables separating textual and visual components, as each may have different importance for diversification. The new ranking function is appropriate for image diversification because the system becomes able to process image based data and its features. Moreover, the separation of diversity coefficients makes framework better at optimizing parameters.

As a further extension, instead of considering an MMR-style approach in $\text{R-LTR}_{\text{IMG}}$, which only takes into account the diversity wrt. the documents that are already in S, we apply the philosophy of aforementioned MMC approach. More specifically, for a given image d_i to be scored, we also compute the upperbound of the diversity that can be brought to S, if d_i is selected. To the best of our knowledge, earlier works on supervised approaches for implicit diversification are based on MMR, and our work is the first attempt to *learn* a framework that considers both the documents in S and those to be inserted into S.

In Eq 310, the last two components address the textual and visual diversity of d_i with respect to l images that are most dissimilar to it in the set of remaining images $U = D - S - d_i$. We refer to this version as $\text{R-LTR}_{\text{IMG-MMC}}$.

$$\begin{aligned}\text{R-LTR}_{\text{IMG-MMC}}(d_i, R_i, S) = \omega_r * x_i + \omega_{textDiv} * h_S(RT_i) + \omega_{visDiv} * h_S(RV_i) + \\ \omega_{textDivNext} * h_U(RT_i, l) + \omega_{visDivNext} * h_U(RV_i, l) \quad (310)\end{aligned}$$

We are aware of a previous work [17] that has also exploited R-LTR for image diversification in a similar setup, i.e., MediaEval evaluation campaign. This work differs from the latter in four ways: First, we implement R-LTR using a neural network framework with back-propagation, which allows us to train more general models. Second, thanks to the flexibility of R-LTR$_{\text{IMG}}$, various learning strategies are implemented and their results are discussed. Third, we extend R-LTR and propose a new variant that learns the MMC ranking function instead of the MMR. Finally, we compare R-LTR variants to two baseline approaches with carefully tuned parameters (to optimize their performance), while the previous work reports only the results of a direct application of R-LTR.

3.3.1 Document Query Relevancy Calculation

There are many well-known methods for calculating query to document distance for textual data, such as *BM-25*. This approach is not working properly for our problem because images have very limited text data, and this textual data may not be so representative as they are user based. In other words, provided textual data of the images are obtained from users while they are sharing these photos in Flickr. Hence, these may not reflect image features correctly and can be noisy. In addition, these type of textual based solutions such as *BM-25*, ignore visual features of image data, which is the most important factor for characterizing an image. To overcome this problem, we have adopted a different solution named $RepresentativeImage$ for query document similarity and extended our framework by integrating that approach to it. As suggested in the $RepresentativeImage$ selection solution, we retrieve most relevant image to the query, which is used as representative document for related query [16]. Whenever a need for calculating similarity between an image and query, representative image is used as replacement of that query, and problem is converged to document to document similarity, which framework is already able to handle as expressed section below.

3.3.1.1 Selecting Representative Image

We have employed a new ranking function to select representative image for a query. Similar to ranking function described in equation 39, this function also uses several features extracted from metadata of photo.

$$query\text{-}sim(d_i, q) = \omega_{geo} * geo\text{-}dist(d_i, q) + \omega_{vis} * vis\text{-}sim(d_i, q) + \omega_{text} * BM\text{-}25(d_i, q) \tag{311}$$

As it can be inferred from equation 311, ranking function uses three features.

- Geographical Distance: Our dataset provides latitude and longitude information of each photo. Since our queries are locations, their geographical coordinates are also known. We have calculated distance of each images to original location, and used this as a feature for measuring relevancy between query and photo.
- Visual Similarity: The most relevant Wikipedia photos of locations is also given in dataset. Since our framework is able to compute document-to-document visual based similarity, we have used visual similarity score between a document and Wikipedia image of given location.
- BM-25 Score: As third feature of similarity method, we have used BM-25 score between query and textual data of the image.

Thanks to Equation 311, our system becomes able to compute similarity score between a document and query. The document, which produces the highest score with given query is selected as representative image for this query.

3.3.2 Document Document Diversity Calculation

This section provides detailed information about diversity calculation component of image based diversification framework. Like expressed on relevancy component, we have changed behaviour of that component also to provide more diverse, and relevant

result sets. Specifications of diversity calculation can be categorized into three main area; relational function, textual, and visual diversity features.

3.3.2.1 Relational Function

Relational function, formally $h(d, R_i)$, is an important factor of diversity calculation. This method is responsible from generating a scalar value for a given document, and document to document diversity tensor. More specifically, this method measures the distance between document and current result set. There can be three different strategies for finding distance between a document and set. With the greatest distance strategy, distance between document and document set defined as maximum distance among all pairs of documents such as q and $q_i \in R$. Similar to greatest distance strategy, smallest distance strategy uses minimum distance obtained from all calculated distance between document pairs, and average distance strategy defines document to set distance as average distance of all documents in the result set to current document [1]. Our experiments show that the framework performed best with average distance strategy, which has formal definition as follows;

$$h(d_i, S) = \frac{1}{|S|} * (\sum_{d_j \in S} \sum_{k \in L} R_{ijk}) \tag{312}$$

where S represents current result set, L denotes list of features used, and R is diversity feature tensor.

3.3.2.2 Textual Diversity Features

Each image contains text based metadata such as description, title and user comments. After identifying all terms in document, we have extracted two features to calculate text based document to document diversity.

- **Cosine Similarity with tf-idf Weighting:** *tf-idf* indicating importance of a term for a document by calculating statistical analysis on occurrences of terms on both document and corpus. These *tf-idf* scores are calculated for each terms in a document and document is represented as a vector of tf-idf values, on

vector space [36]. Then, cosine similarity between two vectors are computed to measure similarity between two documents by using formula below;

$$cos(d_i, d_j) = \frac{d_i \cdot d_j}{|d_i| * |d_j|} \tag{313}$$

where d_i and d_j denote tf-idf based vector representations of the documents.

- **Jaccard Coefficient:** This metric is another index to infer similarities between two sets. As main idea, this metric computes ratio of common terms on both documents. This is achieved by diving number of common terms on both two documents to total number of terms in two documents.

$$jaccard(d_i, d_j) = \frac{terms_{d_i \cap d_j}}{terms_{d_i \cup d_j}} \tag{314}$$

3.3.2.3 Visual Diversity Features

Visual descriptors of an image describes how image is seen from users. Contents of an image, edges, objects and colors can be determined thanks to state of art visual descriptors. Our framework uses some of the best performing visual descriptors among the ones provided by dataset [37]. All visual features used by our framework is listed as follows:

- **Global Histogram of Oriented Gradients (HOG):** This feature is represented by a vector with 81 values. Generally this visual descriptor is used for object, edge, blob and corner detection [38].

- **Global Color Structure Descriptor (CSD):** This visual descriptor designed for representing color structure of an image. Specifically, it denotes an image as color distribution(or histogram) and the local spatial structure of the color citeIte-vil. This descriptor defines image on MPEG-7 Color structure processed on HMMD color space. That visual feature is described by a vector with 64 elements.

- **Global Color Naming Histogram (CN):** This visual feature represents image with a vector of size 11, where each element in the vector denotes image repre-

sentation on one of universal colors domain which are black, blue, brown, grey, green, orange, pink, purple, red, white and yellow.

- **Global Color Moments on HSV Color Space (CM):** Color moments are important visual descriptors as it helps retrieval system to evaluate color similarities between images. This feature denotes first three central moments of color distribution of a document, which are standard deviation, skewness, and mean. As each these distributions are represented by three elements, this feature is represented by a vector with 9 values.

All visual feature descriptors used are defined as vectors with different lengths. Hence, to define relationship between documents with respect to those visual descriptors, a distance calculating function is needed. Our literature researches indicate that there different distance metrics can be used for this purpose, which are cosine, euclidean and chisquare distances. As can be seen in experiments section, after trying both combinations for both descriptors, we've selected the best performing distance calculation methods for each visual feature, which is summarised in Table 3.1.

Table 3.1: This table summarizes distance function and visual descriptor pairs used in our framework.

Visual Descriptors	Distance Functions Used
HOG	Cosine Distance
CSD	Cosine Distance
CN	Euclidean Distance
CM	Euclidean Distance

Thanks to these distance calculation methods, framework calculates distance of documents with respect to each visual descriptors with a scalar between 0 and 1. As 1 scalar by each descriptor, visual diversity feature vector is represented by a vector with 4 elements, where both of the elements between 0 and 1.

In the light of expressed feature vectors for both textual, and visual components,

diversity feature tensors in ranking function can be formulated in detail such as;

$$R_{textual} = R[n][n][2] \tag{315}$$

and

$$R_{visual} = R[n][n][4] \tag{316}$$

To illustrate, $R_{textual}[i][j][1]$ represents diversity score of document i and j with respect to tf-idf weighted cosine, and $R_{visual}[i][j][4]$ denotes CM based diversity score of document i and j.

3.3.3 Learning

The performance of learning process has direct impact on effectiveness of proposed framework as ranking function depends on three parameters namely ω_r, ω_{dtextual} and ω_{dvisual}. This section will explain each step of learning process of the framework in detail.

3.3.3.1 Ideal Ranking List Generation

Ground truth generation is the initial step of learning process of the framework. This step is implemented as a stand alone utility in framework, which gets all metadata as input and return a ranked list with elements which are as diverse as possible it can be. The Algorithm 3 illustrates execution of generating ideal ranking list.

Algorithm 3 Algorithm for ground truth creation

1: $S \leftarrow \emptyset$
2: **while** $|S| < k$ **do**
3: $\quad d_i \leftarrow \text{argmax}_{d_i \in R}(\text{diversity\_score}(\text{S} \cup \text{d}_\text{i}))$
4: $\quad S \leftarrow S \cup [d_i]$
5: $\quad R \leftarrow R \setminus [d_i]$
6: **end while**
7: **return** S

As can be inferred from Algorithm 3, the method starts execution with an empty set, $S_\emptyset$. Sequentially, from all candidate documents $d_i \in R$, it selects the one producing maximum diversity score, according to one of the universal diversity metrics expressed section above, such as *ERR-IA*, *nDCG*, etc. Then, algorithm puts that document to current result set and removes it from candidate document set. In that way, at each iteration, the document introducing the most novelty is selected and result list is expanded.

3.3.3.2 Loss Function

Loss function is critical for learning process, as it defines distance between produced result and optimum one. In our special case, loss function defines distance between generated ranked list and ground truth. As it defines the way learning process will converge, it has direct impact on search effectiveness. We have used negative log-likelihood loss as loss function because it is suitable for learning on neural networks, which has formula below [1];

$$L(f(X, R), y) = -log(P(y|X)) \tag{317}$$

where y denotes ground truth, and $L(f(X, R), y)$ measures distance between ranked list generated by ranking function and ground truth.

3.3.3.3 Optimization Process

Main aim of learning process is to optimize parameters which is achieved at optimization process. By using definition of loss functions and ideal ranking list; optimization is a sequential process which computes loss values of each iteration, and tries to manipulate parameters to converge produced result list to ideal ranking as much as possible. General algorithm of this process is expressed in Algorithm 4.

As can be inferred from Algorithm 4, as the first step, algorithm propagates forward. With that, result set is obtained with current parameters. Then, algorithm calculates loss value between produced result and ground truth by considering indexes of each document. Loss value affects gradient computations of the parameters. Finally, pa-

Algorithm 4 General execution of optimization process

initialize $\omega_r, \omega_{\text{dtextual}}, \omega_{\text{dvisual}}$ with random values
for each epoch **do**
    $ranking\_list \leftarrow forward\_propagation()$
    $loss\_value \leftarrow loss\_function(ranking\_list, ground\_truth)$
    $\Delta\omega_r \leftarrow sgd(loss\_value, \omega_r)$
    $\Delta\omega_{\text{dtextual}} \leftarrow sgd(loss\_value, \omega_{\text{dtextual}})$
    $\Delta\omega_{\text{dvisual}} \leftarrow sgd(loss\_value, \omega_{\text{dvisual}})$ ▷
    Compute gradients of weight vectors using SGD
    $\omega_r \leftarrow \omega_r - learning\_rate * \Delta\omega_r$
    $\omega_{\text{dtextual}} \leftarrow \omega_{\text{dtextual}} - learning\_rate * \Delta\omega_{\text{dtextual}}$
    $\omega_{\text{dvisual}} \leftarrow \omega_{\text{dvisual}} - learning\_rate * \Delta\omega_{\text{dvisual}}$ ▷
    Update weight vectors and models according to learning rate, and gradients
end for
return $\omega_r, \omega_{\text{dtextual}}, \omega_{\text{dvisual}}$

rameters are updated by using gradients, and learning rates to produce closer output to ground truth in next iteration.

3.3.4 Learning Strategies and Neural Network Architectures

That optimization process is implemented by using PyTorch's neural network framework with using back and forward propagation. Ranking scores and ground truth are provided as inputs to the network and network is trained to optimize parameters by reducing the loss value at each step. One of the most popular gradient function, Stochastic Gradient Descent (SGD) is used to find out the direction where parameters should be converged. At each step, back-propagation cycle is triggered to take the values and adjust the parameters by using SGD, in order to reduce total loss value. At the end of this process, optimization is completed by tuning parameters, and converging result list to ground truth on training data as much as possible. While implementing R-LTR, we employed different variants of R-LTR$_{\text{IMG}}$ based on different learning strategies and neural network architectures. Outputs of these variants will be examined in later sections in order to visualize impact of learning process to diversification

performance.

3.3.4.1 Simple Neural Network

As the simplest form of a neural network, we have defined R-LTR$_{\text{IMG}}$'s without a hidden layer. From the definition, the system is composed by only input and output layers and it behaves like a single layer perceptron. As discussed in the previous section, SGD is used for gradient computation and negative log-likelihood method is used as loss function. Additional bias parameter is also used and sigmoid is used as activation function. The execution of the single layer perceptron can be found in Figure 3.2 and Figure 3.3 visualizes architecture of a neural network without a hidden layer. In this work, the framework trained in that way is called as R-LTR$_{\text{IMG}}$.

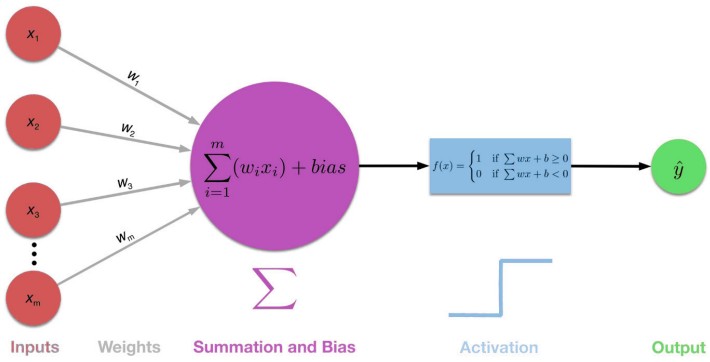

Figure 3.2: Execution of the network with a single layer perceptron.

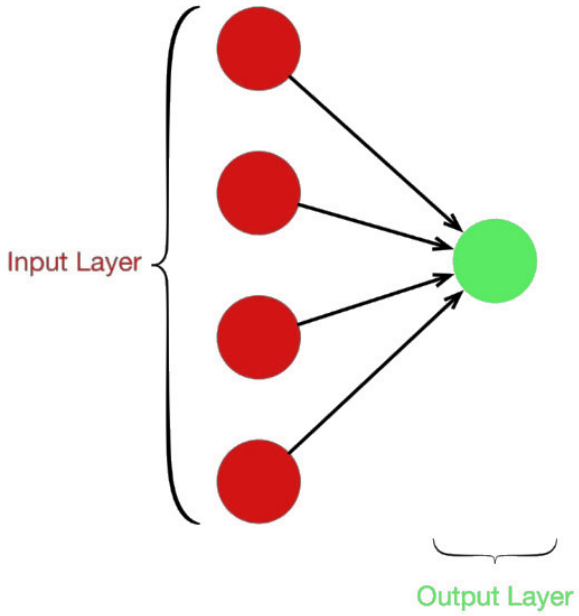

Figure 3.3: Architecture of the network without a hidden layer.

3.3.4.2 Neural Network with Hidden Layer

In order to empower non-linearity to the system and improve performance, we have introduced one hidden layer to the neural network. Like previous architecture, SGD and sigmoids are used for gradient and activation function. Thanks to introduced hidden layer and additional nodes on it, the system becomes able to predict more convenient coefficients. After conducting extensive experiments, it is observed that the system performs best at with 1 hidden layer and 3 nodes on it. We call this version of R-LTR$_{\text{IMG}}$ as R-LTR$_{\text{IMG-MUL}}$ to indicate its coefficient is produced by a multi-layered neural network.

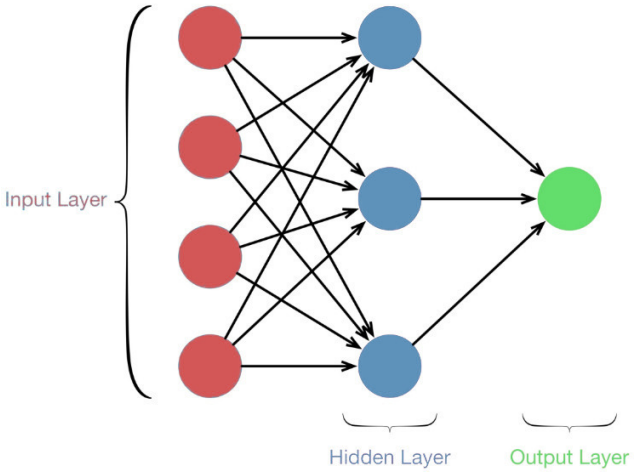

Figure 3.4: Architecture of the network with a hidden layer and 3 nodes on it.

3.3.4.3 Ensemble Learning

We have used bagging based ensemble learning technique discussed in [39]. To this end, we construct random subsets of queries and trained different neural networks by using different query sets, which are initialized with different initial hyper parameters. While computing scores of each documents, results produced by each network are used and average of these scores as used final score of the document. In other words, different networks make contribution to final score thanks to aggregation used. This version of R-LTR$_{\text{IMG}}$ is named as R-LTR$_{\text{IMG-ENS}}$ in the remaining of the thesis.

3.4 Other Improvements

In addition to improvements on working principles of the framework, we have also adopted pre-filtering methods to framework so that it can produce better results. At the beginning of ranking process, irrelevant images are eliminated thanks to these filters, so, pre-filtering step contributes final relevancy, and diversity scores of the system.

3.4.1 Face Detection

While analysing corpus, it is realized that there are people on the foreground of images and these images should be eliminated, because used dataset has queries which are special locations and aim to find location images. To eliminate these irrelevant images, face-detection algorithm is applied on pre-filtering phase of the framework. By integrating Open-CV's built-in face detection algorithms, system becomes able to eliminate images with one or more persons, which increased accuracy of the whole system.

Figure 3.5: Illustration of face detection filtering applied on the framework with real example from dataset. While photo in left side is filtered, image in right side is remained although both have similar visual and textual components, from the same location.

3.4.2 Geographic Filtering

As data set contains queries as location names, such as Big Ben or Acropolis Athens, query represents a geographical place in our system. Latitude and longitude information of each location is extracted from topic files and it is compared with locations of each document. The image is eliminated if its distance from reference point is greater than 10 km.

CHAPTER 4

EXPERIMENTAL SETUP

This chapter provides detailed information about experimental setup. Used dataset will be explained in detail by expressing data source, statistics and query structure. Raw data for generating ground truth is provided by dataset itself, but we have implemented custom algorithms to generate the ideal result lists. This process is also discussed in this chapter. In addition, as mentioned earlier, we have implemented some well known diversity methods on image data and used them as baseline methods. This chapter also includes description of baseline methods. Finally, used evaluation metrics for measuring diversity effectiveness is also represented in that chapter.

4.1 Dataset

In this work, dataset named as *Div150Cred: A Social Image Retrieval Result Diversification with User Tagging Credibility Dataset*, and founded by Ionescu et al. is used [37]. Especially, this dataset was designed to measure performance of search engines from diversity point of view. Also, it was published for participants of MediaEval 2014 Retrieving Diverse Social Images Task and validated at the MediaEval Benchmarking Initiative for Multimedia Evaluation [37], which is one of the most popular benchmark for measuring diversity on multi media based retrieval systems.

Dataset brings 45,375 images of around 300 locations in 35 different countries together, which are produced from 3000 users. Although these locations are vary from widely known places such as Golden Gate Bridge at San Francisco, to some local places not known from most of the people such as Jokhang Temple at Tibet. The most of the locations, which are used in dataset, are listed in the World Heritage Site

of the United Nations Educational, Scientific and Cultural Organization (UNESCO) [37]. Dataset includes several information for each location, such as; location keyword, GPS coordinates, url to its Wikipedia web page, some descriptive photos from Wikipedia and a ranked set of photos retrieved from Flickr with document indexes. In addition, for every image in the dataset, metadata information from the Flickr, several visual descriptors and textual models were provided. Dataset is divided into two parts. Where the first part, namely devset, is designed for training purpose and contains photos of 30 locations, the second part, named as testset, used for evaluating and validating performance of newly proposed retrieval methods, which includes images from 123 locations. Summary of the statistical data comparison between development and test dataset can be seen in Table 4.1. Note that, while testing, we diversify the top-100 images retrieved for a query, as earlier works imply that going deeper in the ranking increases the likelihood of irrelevant results (e.g., [40])

Table 4.1: Comparision of development and test datasets.

Statistic	Devset	Testset
Num. of location	30	123
Num. of images	8923	36452
Num. of images per location	297	296

The details of query structure and ground truth generation of the dataset is explained in sections below.

4.1.1 Queries

During construction of dataset, while retrieving data from Flickr API, location names are used as queries [37]. The same approach was followed while creating dataset, so location names correspond queries in the dataset, which are generally combination of few terms, such as *Louvre Museum* and *Topkapi Palace*. The queries already listed on the dataset and were ready for running experiments.

4.1.2 Ground Truth Generation

Dataset also provides both relevancy and diversity ground truth information for each query.

- *For relevancy information;* a binary relevancy score is available for each document with respect to a query, where 0 indicates document is irrelevant by query, 1 means that document is relevant to query.

- *For the diversity based ground truth;* dataset contains list of all subtopics related with a query and information about subtopics of each document. One document may be related with multiple subtopics.

In order to construct ground truth, which is ideal ranking list from both relevancy and diversity points of view, we have implemented more specific, dataset dependent version of Algorithm 3.

To provide a best performing result set, our ground truth generation algorithm works on just relevant documents and ignores irrelevant documents. With this way, the main iteration is executed on only relevant documents to the query and result set includes only relevant items. To promote diversity on the result set, algorithm selects the element, which maximizes diversity score from all possible candidate documents. So, that approach guarantees that newly selected document will introduce novelty with respect to all other possible documents. As a result, thanks to that approach, generated ground truth satisfies both relevancy and diversity concerns.

4.2 Baseline Methods

This section provides list of baseline methods used to compare our framework performance.

4.2.1 MMR

As stated above, MMR is one of the widely used ranking algorithm for diversification. To have a fair competition, we have implemented MMR from scratch with the same visual and textual descriptors used in R-LTR and its variants. As it can be seen in MMR definition, similarity and diversity scores should be scalar, instead of tensors as represented in framework. So, this implementation requires calculating distances between documents as a single value, with the contributions of all visual and textual features. We have overcome that problem by adopting solution named dynamic feature weighting model to our implementation as presented in [16] with the formula below.

$$div(d_i, d_j) = \frac{1}{|K|} * \sum_{k \in K} (\frac{1}{\theta_k^2} * \mathrm{div_k}(d_i, d_j)) \tag{41}$$

K in the formula represents list of features should be considered, where k denotes an individual feature in the list, and θ_k represents variance of all image similarities with respect to the *i-th* feature and helping normalization of each feature contribution so that they can have the normalized weights [16]. The expression $div_k(d_i, d_j)$ is calculated based on distance measuring methods such as euclidean or cosine distances as this function basically computes distances between two vectors. Finally, $div(d_i, d_j)$ denotes diversity scores between documents d_i and d_j within the given descriptors on the dynamic feature weighting model.

The selected λ value is highly critical in MMR's performance. In order to evaluate MMR and all other methods in a fair situation, we have conducted the same optimization process with using MMR and the system tried to find the most suitable λ value, which makes performance of MMR the best.

4.2.2 MSD

We have also implemented MSD algorithm for using it as baseline methods. Similar to MMR, we have adopted dynamic feature weighting model to compute diversity score with a scalar and decided λ value thanks to our optimization process, which

caused improvements on performance of algorithm. The former two approaches, MMR and MSD, have been widely employed in the literature (e.g., [41, 4]), and hence, serve as the baselines in our setup.

4.2.3 Flickr Search Engine

Dataset already provides ranked result list by Flickr search engine for each query. We have re-constructed Flickr result list and evaluated this result set with the same metrics used to evaluate results of the proposed framework. As Flickr uses state of art retrieval techniques, that helped us to compare newly proposed framework with a technique from latest retrieval technology [37].

4.2.4 Methods from MediaEval Competitions

For many years, MediaEval Benchmark organizes Diverse Images Task in which new frameworks are introduced and their performances are compared within a competition, under defined conditions and runs. We have selected the solutions working on our dataset and compared their performance with respect to different variants of R-LTR$_{\text{IMG}}$.

4.3 Evaluation Metrics

Performance of newly proposed framework is measured by global diversity metrics expressed in Chapter 2. In particular, to evaluate effectiveness of the diversity of R-LTR$_{\text{IMG}}$, $\alpha - nDCG$ and $strec@20$(subtopic recall) metrics are used as suggested by dataset paper. During the evaluation process, global utilities served by TREC, named as NDeval is used to generate these scores.

CHAPTER 5

EXPERIMENTAL RESULTS

The details of the conducted experiments are explained in this chapter. We have executed lots of experiments for two reasons: improving effectiveness of the proposed framework and measuring its performance with respect to baseline methods by using standard metrics as described in previous chapter.

5.1 Epoch Number Analysis

As described in optimization Algorithm 4, epoch number defines the iteration count executed during the optimization process. In neural network based algorithms, epoch number is important to have a well performing model. If epoch number is too small, the model may not be trained well, because it is more possible to reach only local maximum or minimum, instead of the global one, which causes a model be underfitting. If the epoch number is too big, there may be another problem named overfitting and this causes a model to stuck in variants of the data, as it deals with edge cases on a specific data set.

In order to define correct epoch number, we have conducted several experiments to visualize both loss values and diversity scores with obtained parameters produced by these epoch numbers.

As it is clear from Figure 5.2, loss values are converged to 0.05 after epoch 800 for each runs. Hence, this fact indicates that, after epoch 800, there is not any significant change on loss value, as both of the runs produce similar results.

After not getting clear indicator for epoch number from loss value experiments, an-

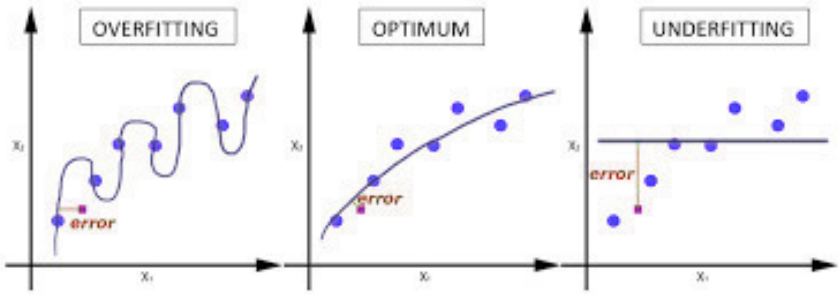

Figure 5.1: Illustration of normal, over, and under fittings. As can be seen, total error is higher on both over, and under fitting graphics.

other experiment is conducted by keeping control group the same(all other hyper parameters such as learning rate, initial values of tensor, etc), and measuring diversity scores of parameters produced by 300, 900, 1500, and 6000 epoch executions.

Table 5.1: Comparision of diversity scores by $strec@20$ metric according to parameters produced by different epoch numbers.

Epoch Number	Score
300	0.443
900	0.455
1500	0.449
6000	0.451

As the execution with 900 epochs produced highest score, it is used for remaining experiments to evaluate framework.

5.2 Feature Importance Analysis

As explained on Chapter 3, framework has six features in total for diversity calculation. 2 features are used as textual descriptors, namely tf-idf weighted cosine similarity and Jaccard coefficient. On the other hand, system uses 4 features for visual descriptors, whose names are HOG, CM, CN, and CSD. After optimization process is completed, the weight tensors are calculated. These tensors are used as coefficients

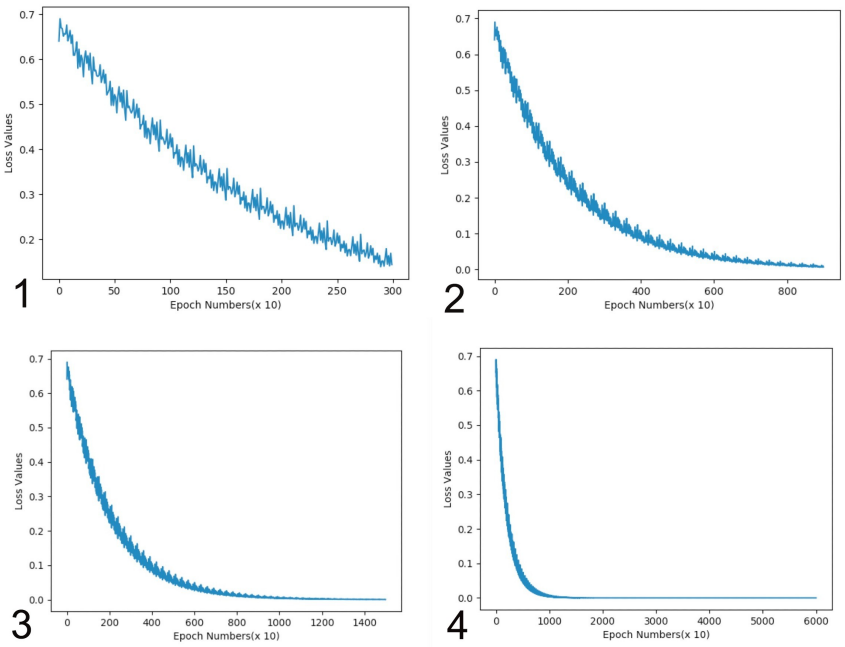

Figure 5.2: This figure illustrates loss value versus epochs graphics. Graphs from 1 to 4 represents executions with epoch numbers 300, 900, 1500, and 6000 respectively.

of the features used. The summary of the calculated weights, for all features used in framework, can be found in Table 5.2 and Table 5.3.

Table 5.2: Weights of Textual Features.

Feature	Weight
Tf-idf weighted cosine similarity	10.891
Jaccard Coefficent	9.154

As higher values of weight indicate more importance of a feature, from the Table 5.2 and Table 5.3 it can be inferred that while tf-idf weighted cosine is more important than Jaccard coefficient from textual features point of view. Similarly, for the visual descriptors, CSD feature has the biggest impact among all other visual descriptors.

Table 5.3: Weights of Visual Features

Feature	Weight
HOG	2.062
CN	4.607
CSD	6.457
CM	1.689

5.3 Pre-Filtering Effect

Effectiveness of used pre-filtering methods is also measured by several experiments. Experiments are executed to cover every combination of two pre-filtering methods applied. Respectively, by keeping control group is the same, 4 experiments are conducted with the configurations of only face detection enabled, only geographical filtering enabled, both enabled and both disabled. Following table summarizes experiment results with produced *strec@20* scores by each configuration.

Table 5.4: Impact of pre-filtering operations to diversity score

Configuration	strec@20 score
Both disabled	0.441
Only face detection enabled	0.451
Only geo filtering enabled	0.449
Both enabled	0.455

As it is proved that geographical filtering and face detection; together, improve diversity of the result set, while comparing performances of proposed framework with baseline models, configuration, with both of them are enabled, is applied.

5.4 Result Set Length Effect on Diversity Score

In order to provide a satisfied search experience, proposed algorithm should work successfully on different search result page size. In other words, diversity performance

of the system should be independent from ranking list size. To test proposed framework effectiveness on different result set lengths, we have executed few experiments to measure ranking list diversity score respectively, while the result list has 5, 10, 15 and 20 documents. The results of the experiments is summarized in Figure 5.3.

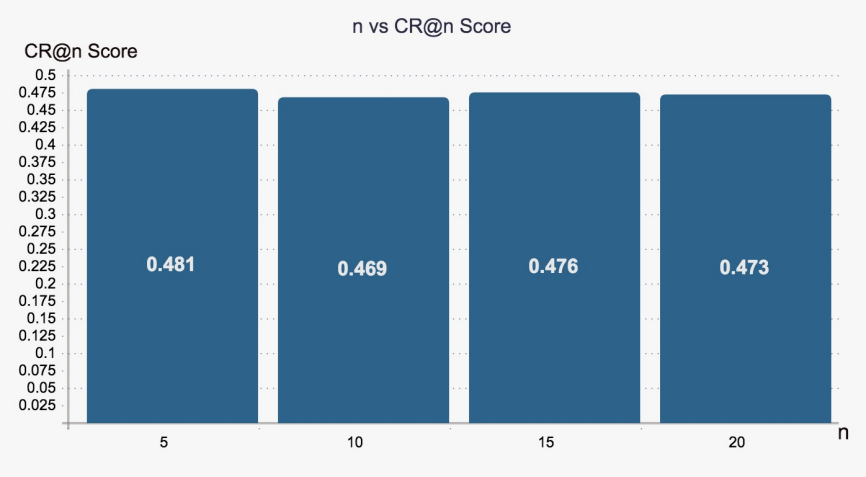

Figure 5.3: Illustration of CR@n score changes with respect to change on length of result set, n.

Although there are some fluctuations are observed between $CR@n$ scores for different values of n, as these changes are around 1%, it is an acceptable level. Hence, it can be inferred that proposed framework is sustainable and applicable for different result set sizes. For the rest of the experiments, result set with length 20 is used, because all official ranking metrics suggested in dataset paper based on result set with 20 documents, such as CR@20 and ERR-IA@20.

5.5 Diversity Performance Analysis

Diversity effectiveness analysis of proposed framework is done by comparing performance of new framework with baseline methods. As mentioned earlier, MMR, MSD and Flickr's state of art retrieval system are used as baseline models to evaluate performance of proposed framework. Both baseline models and new framework are evaluated by using universal metrics α-nDCG@20 and CR@20(subtopic recall). These metrics are selected because these are offical metrics expressed in dataset paper to validate solutions working on the related dataset [37].

Table 5.5: Diversification performance of baseline and proposed approaches. The symbol (*) denotes stat. significance wrt. MMR using paired t-test (at 0.05 confidence level).

Diversity Solution	α-nDCG@20	ST-recall@20
Flickr	0.573	0.342
MSD	0.617	0.369
MMR	0.654	0.413
R-LTR$_{\text{IMG-MMC}}$	0.667	0.425
R-LTR$_{\text{IMG}}$	0.691*	0.455*
R-LTR$_{\text{IMG-MUL}}$	0.695*	0.460*
R-LTR$_{\text{IMG-ENS}}$	**0.697***	**0.465***

We see that, as expected, non-diversified Flickr ranking has the lowest performance, and among the two traditional baselines, MMR is better. The proposed R-LTR$_{\text{IMG-MMC}}$ approach outperforms MMR, with the relative gains of 2% and 3% in terms of α-nDCG and ST-recall metrics, respectively. However, it is still inferior to R-LTR$_{\text{IMG}}$ and its variants, R-LTR$_{\text{IMG-MUL}}$ and R-LTR$_{\text{IMG-ENS}}$. Specifically, R-LTR$_{\text{IMG}}$ provides a relative improvement of 6.3% (3.7%) over the best baseline, MMR, in terms of the ST-recall (α-nDCG) metrics, respectively. R-LTR$_{\text{IMG-MUL}}$ performs better than R-LTR$_{\text{IMG}}$ with providing 11.6% better performance than MMR. R-LTR$_{\text{IMG-ENS}}$ is the overall winner with a further (relative) gain of 12.6% (2.1%) over MMR (R-LTR$_{\text{IMG}}$) in terms of ST-recall, respectively.

Hence, the best performing variant of our framework, namely R-LTR$_{\text{IMG-ENS}}$ outperforms Flickr's state-of-art search engine with the relative gain of 36% in terms of ST-recall metric. Also, it is better than MMR, which is the best baseline method by 12.5%.

5.6 Impact of Learning Strategy on Diversification Performance

R-LTR$_{\text{IMG}}$, R-LTR$_{\text{IMG-MUL}}$ and R-LTR$_{\text{IMG-ENS}}$ approaches are using the same idea and ranking function. The only difference between them is the learning methodology used

while computing coefficients. So, the differences of these frameworks' performance indicates effectiveness of the learning methodology.

As expressed in the previous sections, R-LTR$_{\text{IMG}}$ uses coefficients computed by a simple neural network, R-LTR$_{\text{IMG-MUL}}$ uses neural network with one hidden layer to compute coefficients while R-LTR$_{\text{IMG-ENS}}$ uses *ensemble learning* techniques to generate optimum coefficients. Moving from R-LTR$_{\text{IMG}}$ to R-LTR$_{\text{IMG-ENS}}$, level of non-linearity is increased thanks to used hidden layers and multiple networks. It is observed from table 5.5, R-LTR$_{\text{IMG-ENS}}$ beats both R-LTR$_{\text{IMG-MUL}}$ and R-LTR$_{\text{IMG}}$; while R-LTR$_{\text{IMG-MUL}}$ is performing better than R-LTR$_{\text{IMG}}$. By combining these results with the learning strategies used, it can be concluded that increasing non-linearity produces better results to optimize coefficients, since ensemble learning is performing the best and simple neural network is performing at worst among all R-LTR$_{\text{IMG}}$ variants.

5.7 Comparison to Diversity 2014 Task Results at MediaEval

Among 14 participants of the Diversity task, 10 of them have submitted a run employing both textual and visual features, as we do here. Their median score for ST-recall is 0.4191 and indeed, 9 out of these 10 runs report a score less than 0.45, i.e., inferior to both R-LTR$_{\text{IMG}}$, R-LTR$_{\text{IMG-MUL}}$ and R-LTR$_{\text{IMG-ENS}}$, which yields 0.465.

There is only one run outperforming the R-LTR variants achieves a score of 0.473 [42], but they exploit additional visual features that are not provided in the dataset and hence, not available to us. Indeed, most submitted runs derive new features from the data and/or employ different pre-processing techniques. Therefore, our comparison here is preliminary and will be supported with additional experiments in our future work.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Growing size of the data on the web and especially, the dramatic increase of image data make result diversification for image search a hot research topic. In this thesis, a new approach, which is based on a relational learning to rank method [1], is proposed to solve image diversification problem.

In order to achieve image based search diversification, we adopted a solution that is designed for textual data and based on relational learning to rank model [1]. To extend the latter approach, a new tensor is introduced to take into account the visual descriptors of image data. Also, the ranking method is re-formulated to reflect the affect of visual features as well as textual descriptors. Thanks to this re-formulation, our framework becomes able to tune weights of visual and textual components, so that most useful features can be identified. After defining tensors, the model parameters are optimized by implementing the framework as a neural network.

As both query and documents are represented with the same data types in textual domain, query-document comparison is more problematic on the image domain when compared to textual data diversification. To overcome this problem, another solution named *Representative Image* [16] is adopted to the framework to compute the query and image similarity. Finally, some dataset specific pre-filtering methods are applied to improve the search effectiveness of the framework.

Our exhaustive experiments show that the proposed framework provides significant increase on diversity scores in terms of various metrics. Although there is a small

decrease in the relevancy score, it is negligible as the overall score of the ranked lists produced by our approach are still higher than all baseline methods. Therefore, it can be said that; the proposed framework produces superior results than the traditional and state of the art approaches.

6.2 Future Work

Possible future work directions are listed as follows:

- In addition to Div150 Cred dataset, additional datasets may be used to validate effectiveness of the proposed framework.
- The current framework is designed to operate on only textual and visual features. As a further extension, different set of features such as social features (comments, likes, etc) can be used.

REFERENCES

[1] Y. Zhu, Y. Lan, J. Guo, X. Cheng, and S. Niu, "Learning for search result diversification," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 293–302, ACM, 2014.

[2] M. Kende, "Global internet report 2016," *Internet Society: Reston, VA, USA*, 2016.

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

[4] R. L. T. Santos, C. Macdonald, and I. Ounis, "Search Result Diversification," *Foundations and Trends® in Information Retrieval*, vol. 9, no. 1, pp. 1–90, 2015.

[5] K. D. Onal, I. S. Altingovde, and P. Karagoz, "Utilizing word embeddings for result diversification in tweet search," in *Proc. of AIRS*, pp. 366–378, 2015.

[6] R. Krestel and N. Dokoohaki, "Diversifying customer review rankings," *Neural Networks*, vol. 66, pp. 36–45, 2015.

[7] E. Demidova, P. Fankhauser, X. Zhou, and W. Nejdl, "*DivQ*: diversification for keyword search over structured databases," in *Proc. of SIGIR*, pp. 331–338, 2010.

[8] J. G. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries.," in *SIGIR*, vol. 98, pp. 335–336, 1998.

[9] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. T. Jr., and V. J. Tsotras, "On query result diversification," in *Proc. of ICDE*, pp. 1163–1174, 2011.

[10] S. Gollapudi and A. Sharma, "An axiomatic approach for result diversification," in *Proc. of WWW*, pp. 381–390, 2009.

[11] R. L. Santos, J. Peng, C. Macdonald, and I. Ounis, "Explicit search result diversification through sub-queries," in *European conference on information retrieval*, pp. 87–99, Springer, 2010.

[12] R. L. T. Santos, *Explicit web search result diversification*. PhD thesis, University of Glasgow, 2013.

[13] T. Sakai, N. Craswell, R. Song, S. Robertson, Z. Dou, and C.-Y. Lin, "Simple evaluation metrics for diversified search results.," in *EVIA@ NTCIR*, pp. 42–50, 2010.

[14] R. L. Santos, C. Macdonald, and I. Ounis, "On the suitability of diversity metrics for learning-to-rank for diversity," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 1185–1186, ACM, 2011.

[15] H.-T. Yu, A. Jatowt, R. Blanco, H. Joho, and J. M. Jose, "An in-depth study on diversity evaluation: The importance of intrinsic diversity," *Information Processing & Management*, vol. 53, no. 4, pp. 799–813, 2017.

[16] R. H. van Leuken, L. Garcia, X. Olivares, and R. van Zwol, "Visual diversification of image search results," 2009.

[17] S. Dudy and S. Bedrick, "OHSU @ mediaeval 2015: Adapting textual techniques to multimedia search," in *Working Notes of the MediaEval 2015 Workshop*, 2015.

[18] J. Xu, L. Xia, Y. Lan, J. Guo, and X. Cheng, "Directly optimize diversity evaluation measures: A new approach to search result diversification," *ACM TIST*, vol. 8, no. 3, pp. 41:1–41:26, 2017.

[19] L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng, "Modeling document novelty with neural tensor network for search result diversification," in *Proc. of SIGIR*, pp. 395–404, 2016.

[20] S. Wu, Z. Zhang, and C. Xu, "Evaluating the effectiveness of web search engines on results diversification.," *Information Research: An International Electronic Journal*, vol. 24, no. 1, p. n1, 2019.

[21] M. Drosou and E. Pitoura, "Search result diversification," *SIGMOD record*, vol. 39, no. 1, pp. 41–47, 2010.

[22] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *Proceedings of the second ACM international conference on web search and data mining*, pp. 5–14, ACM, 2009.

[23] C. Zhai and J. Lafferty, "A risk minimization framework for information retrieval," *Information Processing & Management*, vol. 42, no. 1, pp. 31–55, 2006.

[24] A. M. Ozdemiray and I. S. Altingovde, "Score and rank aggregation methods for explicit search result diversification," tech. rep., Technical Report METU-CENG-2013-01, Middle East Technical University, Ankara . . . , 2013.

[25] R. L. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *Proceedings of the 19th international conference on World wide web*, pp. 881–890, ACM, 2010.

[26] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.

[27] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan, "Expected reciprocal rank for graded relevance," in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 621–630, ACM, 2009.

[28] J. D. Martin, "Clustering full text documents," in *Proceedings of the IJCAI Workshop on Data Engineering for Inductive Learning at IJCAI*, vol. 95, Citeseer, 1995.

[29] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: a review," *EURASIP journal on wireless communications and networking*, vol. 2017, no. 1, p. 211, 2017.

[30] S. K. Bhatia, J. S. Deogun, and V. V. Raghavan, "Conceptual query formulation and retrieval," *Journal of Intelligent Information Systems*, vol. 5, pp. 183–209, 1995.

[31] S. J. Cunningham, J. Littin, and I. H. Witten, "Applications of machine learning in information retrieval," 1997.

[32] J. C. Kolecki, "An Introduction to Tensors for Students of Physics and Engineering," *Unixenguaedu*, vol. 7, no. September, p. 29, 2002.

[33] F. Yang, F. Shang, Y. Huang, J. Cheng, J. Li, Y. Zhao, and R. Zhao, "Lftf: A framework for efficient tensor analytics at scale," *Proceedings of the VLDB Endowment*, vol. 10, no. 7, pp. 745–756, 2017.

[34] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[35] T. Qin, T.-Y. Liu, X.-D. Zhang, D.-S. Wang, W.-Y. Xiong, and H. Li, "Learning to rank relational objects and its application to web search," in *Proceedings of the 17th international conference on World Wide Web*, pp. 407–416, ACM, 2008.

[36] S. Qaiser and R. Ali, "Text mining: use of tf-idf to examine the relevance of words to documents," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, 2018.

[37] B. Ionescu, A. Popescu, M. Lupu, A. L. Gînscă, B. Boteanu, and H. Müller, "Div150cred: A social image retrieval result diversification with user tagging credibility dataset," in *Proceedings of the 6th ACM Multimedia Systems Conference*, pp. 207–212, ACM, 2015.

[38] M. Matsugu, K. Mori, M. Ishii, and Y. Mitarai, "Information processing apparatus, information processing method, pattern recognition apparatus, and pattern recognition method," Mar. 9 2010. US Patent 7,676,441.

[39] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, pp. 1619–1630, 2006.

[40] B. Boteanu, I. Mironica, and B. Ionescu, "Pseudo-relevance feedback diversification of social image retrieval results," *Multimedia Tools Appl.*, vol. 76, no. 9, pp. 11889–11916, 2017.

[41] E. S. Xioufis, S. Papadopoulos, A. Gînsca, A. Popescu, Y. Kompatsiaris, and I. P. Vlahavas, "Improving diversity in image search via supervised relevance scoring," in *Proc. of International Conference on Multimedia Retrieval*, pp. 323–330, 2015.

[42] E. S. Xioufis, S. Papadopoulos, Y. Kompatsiaris, and I. P. Vlahavas, "Socialsensor: Finding diverse images at mediaeval 2014," in *Working Notes of the MediaEval 2014 Workshop*, 2014.